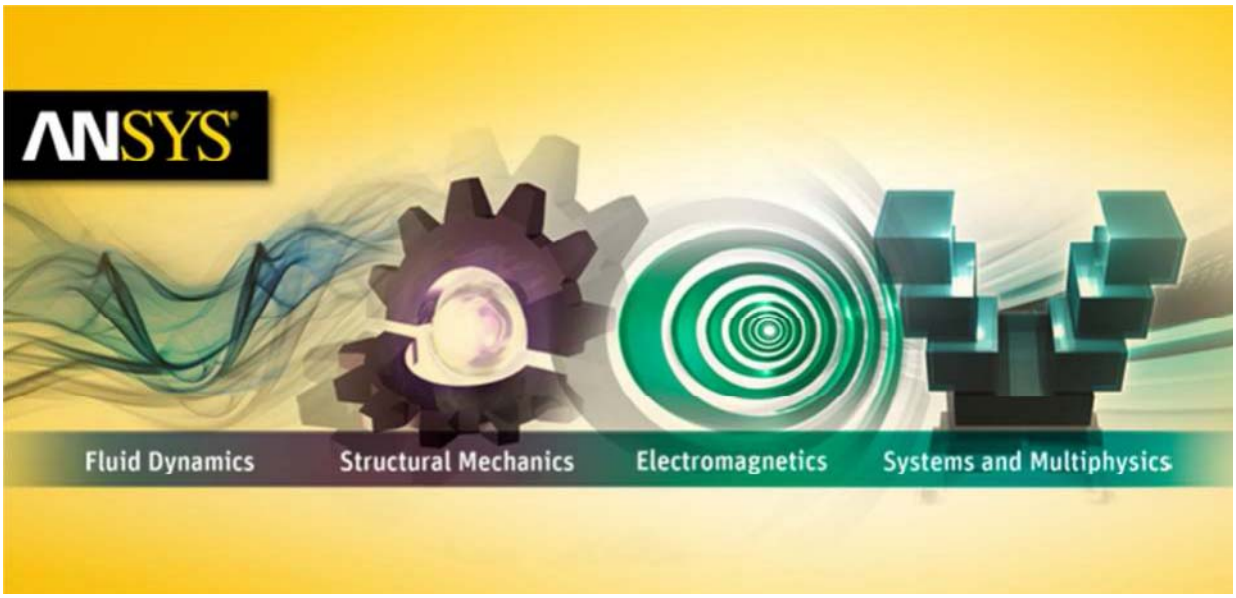


HFSS™ ONLINE HELP

ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

November 2013
ANSYS Electromagnetics Suite 15.0

ANSYS, Inc. is certified to ISO 9001:2008.

Legal Notices

Legal Notice

Copyright and Trademark Information

Disclaimer Notice

U.S. Government Rights

Third-Party Software

This product contains the following licensed software which requires reproduction of the following notices.

Apache commons CLI, Apache Commons Compress, Apache Commons DBCP, and Apache Commons Pool (http://commons.apache.org) - Copyright 2001-2008 The Apache Software Foundation Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

See Apache License for a complete listing of the Apache License.

Apache Formatting Objects Processor Apache License, Version 2.0
http://xmlgraphics.apache.org/fop/license.html

Apache™ FOP License
License

All new Apache™ FOP releases will be licensed under the Apache™ License, version 2.0. Releases until version 0.20.5 are released under the Apache™ License, version 1.1.

To inspect the license terms please consult the LICENSE and NOTICE files in the root directory of the distribution. The full license text can also be accessed via the web from the Apache Licenses page.

Apache FOP comes with a number of libraries all of which are currently coming from within the Apache Software Foundation. These libraries are all licensed under one of the Apache License versions. There is one exception which is "xml-apis.jar" which contains code licensed by the W3C. Please consult the license files included with the libraries in the "lib" directory for details.

BLACS http://netlib.org/scalapack/LICENSE

Boost C++ Libraries http://www.boost.org is licensed under the `Boost Software License V1`http://www.boost.org/users/license.html
CATIA is a registered trademark of Dassault Systems s.a. in France and other countries.
CGAL is licensed under GNU LESSER GENERAL PUBLIC LICENSE (LGPL) Version 3.0. See GNU LESSER GENERAL PUBLIC LICENSE (LGPL) for a complete listing of the GNU LESSER GENERAL PUBLIC LICENSE.
cpp (GCC): Copyright (C) 2003 Free Software Foundation, Inc.
cjpeg: Independent JPEG Group's CJPEG, version 6b 27-Mar-1998 Copyright (C) 1998, Thomas G. Lane
3DConnexion: Development tools and related technology provided under license from 3Dconnexion. © 1992 - 2007 3Dconnexion. All rights reserved. THE 3DCONNEXION SDK AND ALL MATERIALS PROVIDED HEREUNDER BY 3DCONNEXION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. 3DCONNEXION AND ITS AFFILIATED COMPANIES MAKE NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NONINFRINGEMENT REGARDING THE 3DCONNEXION SDK AND ALL MATERIALS OR ITS USE AND OPERATION ALONE OR IN COMBINATION WITH ANY PRODUCT. IN NO EVENT SHALL 3DCONNEXION OR ANY OF ITS AFFILIATED COMPANIES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES (INCLUDING BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA OR PROFITS; OR BUSINESS INTERRUPTION) ARISING IN ANY WAY OUT OF USE, REPRODUCTION, MODIFICATON AND/OR DISTRIBUTION, OR THE INABILITY TO USE, ANY PORTION OF THE 3DCONNEXION SDK OR ITS DERIVATIVES OR MODIFICATIONS, INCLUDING THE DEVELOPER FORCE FEEDBACK GAME, AND ANY OF THE MATERIALS PROVIDED HEREUNDER BY 3DCONNEXION, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE), EVEN IF 3DCONNEXION OR ANY OF ITS AFFILIATED COMPANIES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Because some states/ jurisdictions do not allow the exclusion or limitation of consequential or incidental damages, the above limitation may not apply to you.
ANSYS products make use of the Cray MPI Library technology from Cray, Inc.

DocBook XSL StyleSheets http://wiki.docbook.org/DocBookLicense

Copyright

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ``Software''), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Except as contained in this notice, the names of individuals credited with contribution to this software shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the individuals in question.

Any stylesheet derived from this Software that is publically distributed will be identified with a different name and the version strings in any derived Software will be changed so that no possibility of confusion between the derived package and this Software will exist.

Warranty

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL NORMAN WALSH OR ANY OTHER CONTRIBUTOR BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contacting the Author

The DocBook XSL stylesheets are maintained by Norman Walsh,<ndw@nwalsh.com>, and members of the DocBook Project, <docbook-developers@sf.net>

DXF is a trademark of Autodesk, Inc

The GNU Standard C++ Library http://gcc.gnu.org is licensed under The GNU General Public License (GPL) Version 2 June 1991 Copyright © 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGR4AM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

See GNU GENERAL PUBLIC LICENSE (GPL) for a complete listing of the GNU GENERAL PUBLIC LICENSE.

gzip http://www.gzip.org is licensed under The GNU General Public License (GPL) Version 2 June 1991 Copyright © 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc. at open.source@ansys.com.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGR4AM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

See GNU GENERAL PUBLIC LICENSE (GPL) for a complete listing of the GNU GENERAL PUBLIC LICENSE.

ANSYS products make use of the IBM Platform MPI technology (formerly named HP-MPI) from IBM Corporation.

IGES is a trademark of IGES Data Analysis, Inc

IIOP.Net http://iiop-net.sourceforge.net is licensed under GNU LESSER GENERAL PUBLIC LICENSE (LGPL) Version 2.1, February 1999 Copyright © 1991, 1999, Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc. at open.source@ansys.com.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. NO WARRANTY BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

See GNU LESSER GENERAL PUBLIC LICENSE (LGPL) for a complete listing of the GNU LESSER GENERAL PUBLIC LICENSE.

Portions Copyright © 2002-2009 Charlie Poole or Copyright © 2002-2004 James W. Newkirk, Michael C. Two, Alexei A. Vorontsov or Copyright © 2000-2002 Philip A. Craig

Intel is a registered trademark of Intel Corporation

ANSYS products make use of the Intel (TM) MPI Library technology from Intel Corporation.

Inventor and Mechanical Desktop are registered trademarks of Autodesk, Inc.

java service wrapper http://wrapper.tanukisoftware.org/doc/english/licenseServer.html **is licensed under Community Software License Agreement** Version 1.0 Copyright © 1999, 2006 Tanuki Software, Inc.

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc. at open.source@ansys.com.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Java Service Wrapper and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub-license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

See Community Software License Agreement for a complete listing of the Community Software License Agreement.

JCalendar http://flib.sourceforge.net/JCalendar/doc/index.html

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc. at open.source@ansys.com.

jdk http://java.sun.com/javase/ **is licensed under Sun's Binary Code License**.

JGroups - Copyright (c) 2002-2010 Bela Ban/Red Hat - This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA. See ANSYS, Inc. Contact Information for contact information. See GNU LESSER GENERAL PUBLIC LICENSE (LGPL) for a complete listing of the GNU LESSER GENERAL PUBLIC LICENSE.
This software is based in part on the work of the Independent JPEG Group.
CONTAINS IBM(R) 64-bit Runtime Environment for AIX(TM), Java(TM) 2 Technology Edition, Version 1.4 Modules (c) Copyright IBM Corporation 1999, 2002 All Rights Reserved
Jython http://www.jython.org is licensed under the PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2 Copyright (c) 2007 Python Software Foundation; All Rights Reserved See http://www.jython.org/license.html for a complete listing of the license.

LBNL double-double precision, quad-double precision and arbitrary precision (also termed "multiprecision" or "multiple precision") software:

libtiff

libungif

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and

associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Mathcad is a registered trademark of Mathsoft Engineering & Education, Inc.

Mesa Library license requirements: Copyright (C) 1999-2005 Brian Paul All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Mesa Component Licenses also apply. Reference http://mesa3d.sourceforge.net/.

METIS 4.0.1 Copyright 1997, Regents of the University of Minnesota. METIS was written by George Karypis (karypis@cs.umn.edu)

Our policy regarding the distribution of METIS with third-party applications is as follows: Non-commercial applications METIS can be freely distributed provided that proper references are included and the original documentation and copyright notice is included.

Commercial applications METIS can be freely distributed provided that proper references are included, the original documentation and copyright notice is included and METIS is a relatively small portion of the overall application. In either case, permission to distribute/include METIS with your application must be obtained by sending email to metis@cs.umn.edu.

Microsoft, Windows, Windows 2000 and Windows XP are registered trademarks of Microsoft Corporation.

ANSYS products use the Mono open-source technology to execute on non-.NET enabled platforms. ANSYS has obtained for its customers a covenant from Microsoft not to sue any such customer who uses Mono to run ANSYS software on these platforms. The specific covenant can be accessed at http://www.microsoft.com/interop/collab/ansys/customer-covenant.aspx.

Numpy: http://numpy.scipy.org/

ScaLAPACK http://netlib.org/scalapack/LICENSE
Copyright (c) 1992-2011 The University of Tennessee and The University
of Tennessee Research Foundation. All rights
reserved.
Copyright (c) 2000-2011 The University of California Berkeley. All
rights reserved.
Copyright (c) 2006-2011 The University of Colorado Denver. All rights
reserved.
$COPYRIGHT$
Additional copyrights may follow
$HEADER$

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer listed
in this license in the documentation and/or other materials
provided with the distribution.
- Neither the name of the copyright holders nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

The copyright holders provide no reassurances that the source code
provided does not infringe any patent, copyright, or any other
intellectual property rights of third parties. The copyright holders
disclaim any liability to any recipient for claims brought against
recipient by any third party for infringement of that parties
intellectual property rights.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU Tar http://www.gnu.org/software/tar is licensed under The GNU General Public License (GPL) Version 3, 29 June 2007 Copyright © 2007 Free Software Foundation, Inc. http://fsf.org/

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGR4AM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

See GNU GENERAL PUBLIC LICENSE (GPL) for a complete listing of the GNU GENERAL PUBLIC LICENSE.

TIFF: Copyright (c) 1988-1997 Sam Leffler, Copyright (c) 1991-1997 Silicon Graphics, Inc. - Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

TIFF: pnmtotiff.c - converts a portable anymap to a Tagged Image File Derived by Jef Poskanzer from ras2tif.c, which is: Copyright (c) 1990 by Sun Microsystems, Inc. Author: Patrick J. Naughton, naughton@wind.sun.com Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The Visualization ToolKit (VTK): http://www.vtk.org

VTK is an open-source toolkit licensed under the `BSD license` http://opensource.org/licenses/bsd-license.php.

Visual MainWin® Copyright 2002 Mainsoft Corporation. All rights reserved. Mainsoft, Mainwin, the software porting company, e-porting, the e-porting company, and Visual MainWin are trademarks or registered trademarks of Mainsoft Corporation in the United States and/or foreign countries.
whirlgif - Copyright (c) 1997, 1998, 1999 by Hans Dinsen-Hansen Copyright (c) 1995, 1996 by Kevin Kadow Copyright (c) 1990, 1991, 1992 by Mark Podlipec. All rights reserved. This software may be freely copied, modified and redistributed without fee provided that this copyright notice is preserved intact on all copies and modified copies. There is no warranty or other guarantee of fitness of this software. It is provided solely "as is". The author(s) disclaim(s) all responsibility and liability with respect to this software's usage or its effect upon hardware or computer systems.
wxPytjon: http://www.wxpython.org/ for the Grapical User Interface is licensed under the `wxWindows Licence` http://wxwidgets.org/newlicen.htm.

XML.c - implementation file for basic XML parser written in ANSI C++ for portability. It works by using recursion and a node tree for breaking down the elements of an XML document. @version V2.25 @author Frank Vanden Berghen BSD license: Copyright (c) 2002, Frank Vanden Berghen All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of the Frank Vanden Berghen nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#ZipLib: Copyright (C) 2001 Mike Krueger Copyright (C) 2004 John Reilly
ZLIB 1.1.3 Copyright (C) 1995-1998 Jean-loup Gailly and Mark Adler

- **ajax4jsf** http://labs.jboss.com/jbossajax4jsf/
- **bsh** http://www.beanshell.org/
- **cismet** https://github.com/cismet/cismet-gui-commons
- **infinispan** http://www.jboss.org/infinispan
- **itext** http://itextpdf.com
- **jboss** http://www.jboss.com/
- **JCommon** http://www.jfree.org/jcommon/
- **jexcelapi** http://jexcelapi.sourceforge.net/
- **jgroups** http://www.jgroups.org
- **jfreechart** http://www.jfree.org/jfreechart/
- **jgraph** http://www.jgraph.com/
- **LOGBack** http://logback.qos.ch/
- **Mono runtime** http://www.mono-project.com/
- **pthreads-win32** http://sourceware.org/pthreads-win32
- **richfaces** http://www.jboss.org/richfaces
- **7-Zip** http://www.7-zip.org/

are licensed under GNU LESSER GENERAL PUBLIC LICENSE (LGPL) Version 2.1, February 1999 Copyright © 1991, 1999, Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc. at open.source@ansys.com.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. NO WARRANTY BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

See GNU LESSER GENERAL PUBLIC LICENSE (LGPL) for a complete listing of the GNU LESSER GENERAL PUBLIC LICENSE.

- **ant** http://ant.apache.org/
- **axis** http://ws.apache.org/axis/
- **boilerpipe** http://code.google.com/p/boilerpipe
- **derby** http://db.apache.org/derby/
- **Geronimo 1.0.1 stax** https://svn.apache.org/repos/asf/geronimo/specs/tags/geronimo-stax-api_1.0_spec-1.0.1/
- **jakarta commons** http://jakarta.apache.org/commons/

  (beanutils, codec, collections, compress, dbcp, net, pool, digester, discovery, el, fileupload, io, httpclient, lang, validator, logging, net, cli)
- **jackrabbit**, jcr http://jackrabbit.apache.org/
- **jakarta-oro** http://jakarta.apache.org/oro/
- **jakarta jstl** http://jakarta.apache.org/taglibs/index.html
- **jdom** http://www.jdom.org/index.html
- **log4j** http://logging.apache.org/log4j/
- **lucene** http://lucene.apache.org/java/docs/
- **Metadata extractor** http://www.drewnoakes.com/drewnoakes.com/code/exif/
- **myfaces** http://www.myfaces.org/
- **nekohtml** http://sourceforge.net/projects/nekohtml
- **PDFBox** http://www.pdfbox.org/
- **poi** http://jakarta.apache.org/poi/
- **River** http://river.apache.org/
- **Rome** http://java.net/projects/rome/
- **TagSoup** http://home.ccil.org/~cowan/XML/tagsoup/
- **Tika** http://tika.apache.org/
- **tomcat** http://tomcat.apache.org/index.html
- **xerces** http://xerces.apache.org/
- **XMLBeans** http://xmlbeans.apache.org/index.html
- **xmlrpc** http://ws.apache.org/xmlrpc/index.html

are licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
See Apache License for a complete listing of the Apache License.

- **dom4j** http://sourceforge.net/projects/dom4j

is licensed under BSD License Copyright © 2002-2007, www.pdfbox.org All rights reserved. You may obtain a copy of the License at: http://www.pdfbox.org/license.html#License. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of pdfbox; nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- **graphviz** http://www.graphviz.org/
- **wsdl4j** http://wsdl4j.sourceforge.net/

are licensed under Common Public License Version 1.0 (CPL)

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc. at open.source@ansys.com.

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

See Common Public License Version 1.0 (CPL) for a complete listing of the Common Public License Version 1.0 (CPL).

1. ANSYS, Inc. Contact Information

ANSYS, Inc.

Southpointe
275 Technology Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Contact ANSYS, Inc. at open.source@ansys.com for Open source code.

2. ANSYS Trademarks List

- Ansoft™
- Ansoft Designer®
- Ansoft Links™
- ANSYS®
- ANSYS® AI*Environment™
- ANSYS® Airpak®
- ANSYS® Asas™
- ANSYS® Aqwa™
- ANSYS® Autodyn®
- ANSYS® AutoReaGas™
- ANSYS® BladeModeler™
- ANSYS® CFX®
- ANSYS® DesignModeler™
- ANSYS® DesignSpace®
- ANSYS® DesignXplorer™
- ANSYS® DesignXplorerVT™
- ANSYS® EKM™
- ANSYS® Engineering Knowledge Manager™
- ANSYS® Emag™
- ANSYS® Fatigue™
- ANSYS® FIDAP™
- ANSYS® Flotran®
- ANSYS® Fluent®
- ANSYS® Iceboard®
- ANSYS® ICEM CFD™
- ANSYS® Icechip®
- ANSYS® Icemax®
- ANSYS® Icepro™
- ANSYS® Icepak®
- ANSYS® Icewave™

- ANSYS® LS-DYNA®
- Maxwell®
- ANSYS® Mechanical™
- ANSYS® Mesh Morpher™
- ANSYS® Multiphysics™
- ANSYS® ParaMesh™
- ANSYS® Polyflow®
- ANSYS® Professional™
- Simplorer®
- Solver on Demand®
- ANSYS® Structural™
- ANSYS® TGrid™
- ANSYS® TurboGrid™
- ANSYS® Workbench™
- ANSYS® Vista™
- ANSYS® VT Accelerator™
- CADOE™
- CoolSim™
- Enductive Solutions™
- ePhysics®
- Full-Wave SPICE™
- FloWizard®
- FLUENT/FLOWLAB®
- GAMBIT®
- G/Turbo®
- HFSS™
- Mixsim®
- Nexxim®
- Optimetrics™
- Parlcs®
- PExprt™

- Q3D Extractor™
- QuickEye™
- RMxprt™
- Serende®
- Slwave™
- Simulation Driven™
- Simulation Driven Product Development™
- Super-Compact®
- VerifEye™
- TPA™

3. GNU LESSER GENERAL PUBLIC LICENSE (LGPL)

Included herewith:

Version 3

Version 2.1

VERSION 3

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and

performs whatever part of its purpose remains meaningful, or

- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
  - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
  - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU

GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

VERSION 2.1

GNU LESSER GENERAL PUBLIC LICENSE (LGPL) Version 2.1, February 1999 Copyright © 1991, 1999, Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc. at open.source@ansys.com.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. NO WARRANTY BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFOR-

MANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with

modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License,

whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is

unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to

decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

4. Apache License

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor

for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

5. Common Public License Version 1.0 (CPL)

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial

Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

6. Community Software License Agreement

Community Software License Agreement

Version 1.0 Copyright © 1999, 2006 Tanuki Software, Inc.

ANSYS, Inc. will provide you with a complete machine-readable copy of the source code, valid for three years. The source code can be obtained by contacting ANSYS, Inc. at open.source@ansys.com.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Java Service Wrapper and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub-license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Tanuki Software, Inc. Community Software License Agreement Version 1.0

IMPORTANT-READ CAREFULLY: This license agreement is a legal agreement between you and Tanuki Software, Inc.("TSI"), which includes computer software, associated media, printed materials, and may include online or electronic documentation (Software). PLEASE READ THIS

AGREEMENT CAREFULLY BEFORE YOU INSTALL, COPY, DOWNLOAD OR USE THE SOFTWARE ACCOMPANYING THIS PACKAGE.

Section 1 - Grant of License

Community editions of the Software are made available on the GNU General Public License, Version 2 ("GPLv2"), included in Section 3 of this license document. All sections of the Community Software License Agreement must be complied with in addition to those of the GPLv2.

Section 2 - Your Obligations

A copy of this license must be distributed in full with the Product in a location that is obvious to Your customers. The Software Program may not be modified, nor may the Product in any way obfuscate or obstruct the copyright notice and license information displayed in the console and log files by the Software Program on startup.

Section 3 - GPLv2 License Agreement

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps:

(1) copyright the software, and

(2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an

announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distri-

bution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribu-

tion is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Section 4 - 3rd Party Components

(1) The Software Program includes software and documentation components developed in part by Silver Egg Technology, Inc. ("SET") prior to 2001. All SET components were released under the following license.

7. Common Development and Distribution License (CDDL)

Original Software or previous Modification; or C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means

- the common form of computer software code in which modifications are made and
- associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

- under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
- under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
- The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
- Notwithstanding Section 2.1(b) above, no patent license is granted:
  - for code that You delete from the Original Software, or
  - for infringements caused by:
    - the modification of the Original Software, or
    - the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- under intellectual property rights (other than patent or trademark) Licensable by Contribu-

tor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

- under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of:
  - Modifications made by that Contributor (or portions thereof); and
  - the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
- The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
- Notwithstanding Section 2.2(b) above, no patent license is granted:
  - for any code that Contributor has deleted from the Contributor Version;
  - for infringements caused by:
    - third party modifications of Contributor Version, or
    - the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or
    - under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You:

- rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and
- otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNC-

TION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

---

8. GNU GENERAL PUBLIC LICENSE (GPL)

Included herewith:

Version 3

Version 2

VERSION 3

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component,

or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a. The work must carry prominent notices stating that you modified it, and giving a relevant date.

b. The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.

c. You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d. If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the

software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c. Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d. Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e. Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you

nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a. Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b. Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c. Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d. Limiting the use for publicity purposes of names of licensors or authors of the material; or

e. Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f. Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered

work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

VERSION 2

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and

change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps:

(1) copyright the software, and

(2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way

you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Section 4 - 3rd Party Components

(1) The Software Program includes software and documentation components developed in part by Silver Egg Technology, Inc. ("SET") prior to 2001. All SET components were released under the following license.

Copyright (c) 2001 Silver Egg Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub-license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Table of Contents

2. Getting Help

3. Working with HFSS Projects

7. Drawing a Model

8. Assigning Boundaries

9. Assign Excitation

12. Modifying the Model View

13. Defining Mesh Operations

14. Specifying Solution Settings

15. Running Simulations

16. Optimetrics

17. Post Processing and Generating Reports

18. Network Data Explorer

19. Technical Notes

20. Scripting

21. ANSYS Workbench Integration Overview

22. Desktop Scripting with IronPython

1

Getting Started with HFSS

HFSS™ is an interactive software package for calculating the electromagnetic behavior of a structure. The software includes post-processing commands for analyzing this behavior in detail.

Using HFSS, you can compute:

- Basic electromagnetic field quantities and, for open boundary problems, radiated near and far fields.
- Characteristic port impedances and propagation constants.
- Generalized S-parameters and S-parameters renormalized to specific port impedances.
- The eigenmodes, or resonances, of a structure.

You are expected to draw the structure, specify material characteristics for each object, and identify ports and special surface characteristics. HFSS then generates the necessary field solutions and associated port characteristics and S-parameters.

Note If you are using the Eigenmode Solution solver, you do not need to specify sources for the problem. HFSS calculates the resonances for the model based on the geometry, materials, and boundaries.

As you set up the problem, HFSS allows you to specify whether to solve the problem at one specific frequency or at several frequencies within a range.

System Requirements

HFSS supports certain versions of Windows and Linux. For supported platforms and system requirements, go to:

http://www.ansys.com/Support/Platform+Support.

Setting Up a Printer on Linux

To print from ANSYS software on Linux, you must first configure a printer. To do this, launch the MainWin control panel.

1. Run **mwcontrol &** in the installation subdirectory.
   The **MainWin Control Panel** appears.
2. Double-click on the **Printers** icon to start the **MainWin Printers** panel.
3. Then double-click on the **Add New Printer** icon.
   This starts the **Add Printer Wizard**.
4. Select the **Let the wizard search for printers** radio button and click **Next**.
5. In the **Identify your Unix Printer** dialog do one of the following:
   - If your printer is listed, select it.
   - If your printer is not listed, you will need to cancel and get someone with root permission to setup a printer queue on your machine (and then you will need to come back and run this wizard later).

Note On Solaris you setup a new print queue by running "lpadmin" (as root). On Red Hat Linux, you can run 'System Settings/Printing' to launch printconf-gui (as root).

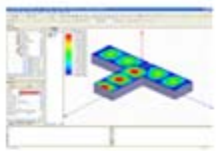
6. Click **Next**.
   The **Print Command** dialog appears.
7. Change the Print Command only if instructed to do so by your user administrator.
8. Click **Next**.
   The **Choose PPD File** dialog appears.
9. Select your printer manufacturer and model from the list or use the **Choose File** button to browse to a PPD file provided by your printer manufacturer. Click **Next**.

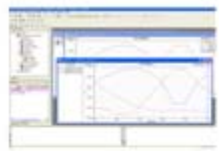
   The **Printer Name** dialog appears.
10. Enter a **Name** to identify the printer. Click **Next**.
11. Choose whether this printer should be the default and click **Next**.
12. Choose whether you would like to print a test page and click **Next**.
13. In the **Finish Adding New Printer** dialog, verify the printer setup information. If the information is incorrect, use the **Back** button to return to the appropriate dialog and correct the entry. If the information is correct, click **Finish** to complete the setup of your printer.

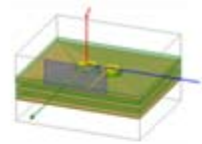
With a print queue setup, and the printer added, you should then see the printer when running the software.

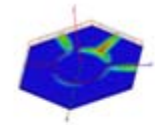
Welcome to HFSS Online Help

Click for help on these topics.

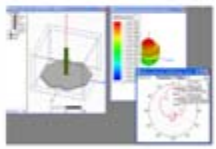

Getting Started HFSS

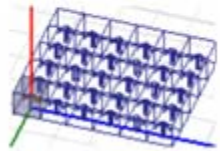

Getting Started Optimetrics


Signal Integrity

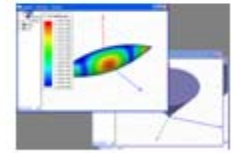

RF Microwave

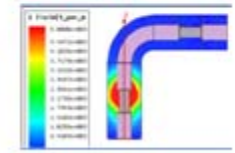

Integrated Circuits


Antennas


Arrays


Finite Element Boundary Integral


Introduction to HFSS-IE


Introduction to HFSS Transient

Click for pdf format **Introduction to HFSS:** Fundamental Principles, Concepts and Use.

. Introduction to HFSS

Click for help on HFSS **Process Flow** topics:

Click for HFSS General Quick Links including HFSS Desktop and Working with Projects.

HFSS General Quick Links

Use the following links for quick information on the following topics.

HFSS User Interface	HFSS-IE
Setting Up an HFSS or HFSS-IE Design	Solution Types
Modeling	Materials
Boundaries and Ports	Meshing
Analysis	Optimetrics
Results	Scripting
Example Projects	ANSYS Web site

For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.

- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Online Help **Index tab** selected, search the help index.
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

HFSS User Interface Quick Links

Use the following links for quick information on the following topics.

The HFSS Desktop
Customize Toolbar Options
Setting Options in HFSS
Working with Short Cut Menus
Keyboard Shortcuts for HFSS General Purposes
Running HFSS From a Command Line
HFSS-IE
Moving and Resizing Desktop Windows
Showing and Hiding Windows
Working with Variables
ANSYS Workbench Integration Overview

For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.
- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Online Help **Index tab** selected, search the help index.
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

Modeling Quick Links

Use the following links for quick information on the following topics.

Set the model's units of measurement.
Setting Modeler Drawing Options
Assign transparency to an object.
Selecting Items in the Modeler Window
Subtract objects.
Drawing Objects
Measuring Objects
Choosing the Cursor Movement Mode
Drawing Bondwires
Importing Files

Keyboard shortcuts for the 3D Modeler Window.	Modifying Objects
Modifying the Model View	Creating a User Defined Primitive
Working with Variables	Assigning Materials

For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.
- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Online Help **Index tab** selected, search the help index.
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

Materials Quick Links

Use the following links for quick information on the following topics.

Assigning Materials	Solve Inside or On a Surface
Searching for Materials	Adding New Materials
Assigning Material Property Types	Defining Variable Material Properties
Defining Frequency Dependent Material Properties	Defining Material Properties as Expressions
Defining Functional Material Properties	Viewing and Modifying Material Attributes
Assigning DC Thickness	Specifying Thermal Modifiers

For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.
- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Online Help **Index tab** selected, search the help index.
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

Ports Quick Links

Use the following links for quick information on the following topics.

Assigning Excitations
Assigning Wave Ports for Modal Solutions
Assigning Wave Ports for Terminal Solutions
Floquet Port
Incident Wave
Current Source
Defining an Integration Line
Linking to External Sources
Lumped Port
Auto Assign Terminals
Zoom to Selected Excitation
Voltage Source
Magnetic Bias
Defining a Differential Pair

For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.
- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Online Help **Index tab** selected, search the help index.
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

Meshing Quick Links

Use the following links for quick information on the following topics.

Defining Mesh Operations
Detecting and Addressing Model Problems to Improve Meshing
Specify the initial mesh settings
Plot the finite element mesh
Handling Complicated Models

For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.
- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Online Help **Index tab** selected, search the help index.
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

Analysis Quick Links

Use the following links for quick information on the following topics.

Specifying the HPC and Analysis Options	Remote Analysis
Configuring Distributed Analysis	Specifying Solution Settings
Selecting an Optimal Configuration for Distributed Analysis	Setting Adaptive Analysis Parameters
Specifying Output Variable Convergence	Setting the Order of Basis Functions
Adding a Frequency Sweep	Options for Interpolating Sweeps
Running Simulations	High Performance Computing Integration
ANSYS Workbench Integration Overview	Setup and Run an Array Simulation

For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.
- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Online Help **Index tab** selected, search the help index.
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

Optimetrics Quick Links

Use the following links for quick information on the following topics.

Optimetrics	Parametric Overview
Setting up a Parametric Analysis	Setting up an Optimization Analysis
Setting up a Sensitivity Analysis	Tuning a Variable
Setting up a Statistical Analysis	Setting a Range function
Setup Calculations for Optimetrics.	Adding a cost function
Running an Optimetrics Analysis	Getting Started Guide: Optimizing Waveguide T-Junction

For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.
- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.

- With the Online Help **Index tab** selected, search the help index.
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

Results Quick Links

Use the following links for quick information on the following topics.

View solution data	Creating Reports
Plot field overlay	Scale an excitation's magnitude and modify its phase.
Create 2D or 3D reports of S-parameters	Working with Traces
Plot the finite element mesh	Using the Fields Calculator
Create animations	Radiated Fields Post Processing
Setting up a Near Field Sphere	Setting up a Far Field Infinite Sphere

For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.
- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Online Help **Index tab** selected, search the help index.
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

Scripting Quick Links

Use the following links for quick information on the following topics.

Recording a Script	Running a script
Stopping Script Recording	Pausing and Resuming a Script
Stopping a Script	Scripting Guide

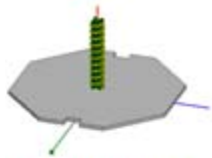
For detailed information on these and many other topics:

- Use F1 on any open dialog to open the Online Help for that dialog.
- Click the "?" icon on the toolbar, and then click on any menu command, icon, or window for help on that selection.
- With the Online Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Online Help **Index tab** selected, search the help index.

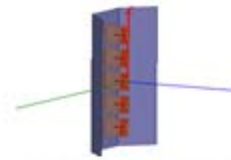
- With the Online Help **Search tab** selected, search the full help text.
- With Online Help **Favorites tab** selected, create a custom list of favorite topics.

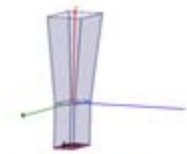
Antennas in HFSS

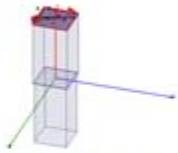
Click graphics for descriptions of the following HFSS and HFSS-IE antenna examples:


Helical Antenna

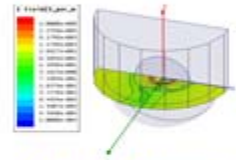

Mast - IE

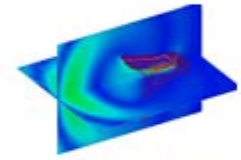

Planar Flared Dipole Array

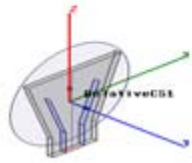

Pyramidal Horn

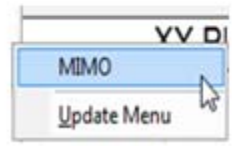

Unit Cell Floquet


Coax Fed Patch (Tuning Example)

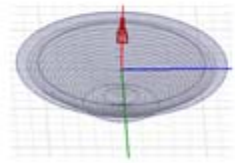

Dielectric Resonator Antenna


Ridged Horn Transient


Phase Center via Optimetrics



MIMO Calculations


Dish FEBI IEPO

Click for Arrays.

Click for HFSS Application Modeling Guide: Antennas.

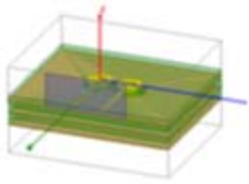
Click for the download site for the HFSS Antenna Design kit.

The HFSS Antenna Design Kit is a stand-alone GUI-based utility which automates the geometry creation, solution setup, and post-processing reports for over 25 antenna elements. This tool allows antenna designers to efficiently analyze common antenna types using HFSS and also assists new users in learning to use HFSS for antenna design. The design kit can be integrated into the HFSS user interface or launched from the standard Windows menu. All antenna models created by the design kit are ready to simulate in HFSS.

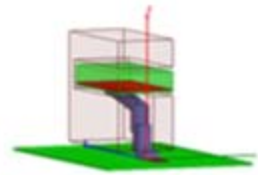
Also see 3D Component Library: Antennas

Signal Integrity in HFSS

Click graphics for descriptions of the following HFSS signal integrity examples:



Via Wizard

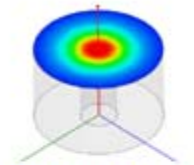


Connector

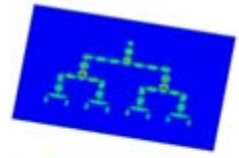
Ball Grid Array Transient

RF/Microwave in HFSS

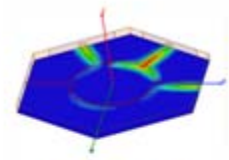
Click graphics for descriptions of the following HFSS RF/Microwave examples:



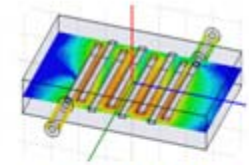
Coaxial Resonator



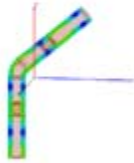
Corporate Feed



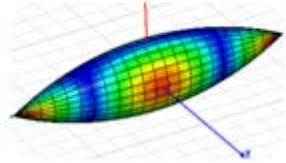
Ring Hybrid



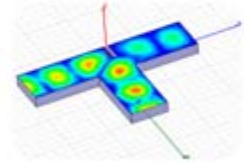
Bandpass Filter



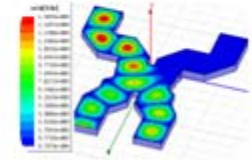
Parametric Sweep of Coax Bend



Ogive – IE



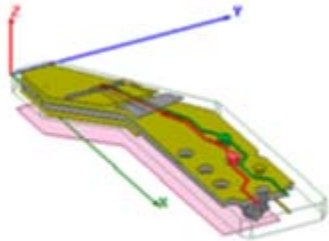
Tee and Optimum Tee



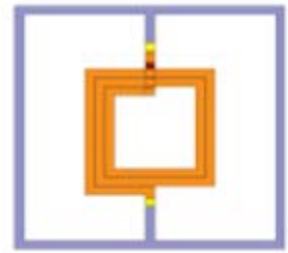
Combiner

Integrated Circuits in HFSS

Click the graphics for examples of modeling integrated circuits in HFSS:



Package

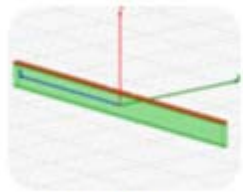


Spiral Inductor

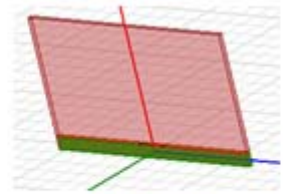
Also see: Application Specific Modeling Guide: Spiral Inductors on Silicon Substrate

Transmission Line Examples for HFSS

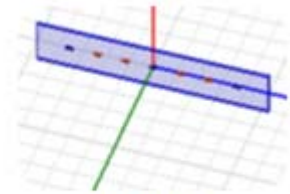
Click on the following examples for descriptions of transmission line examples provided with HFSS.



Coplanar Waveguide



Coplanar Waveguide w gnd

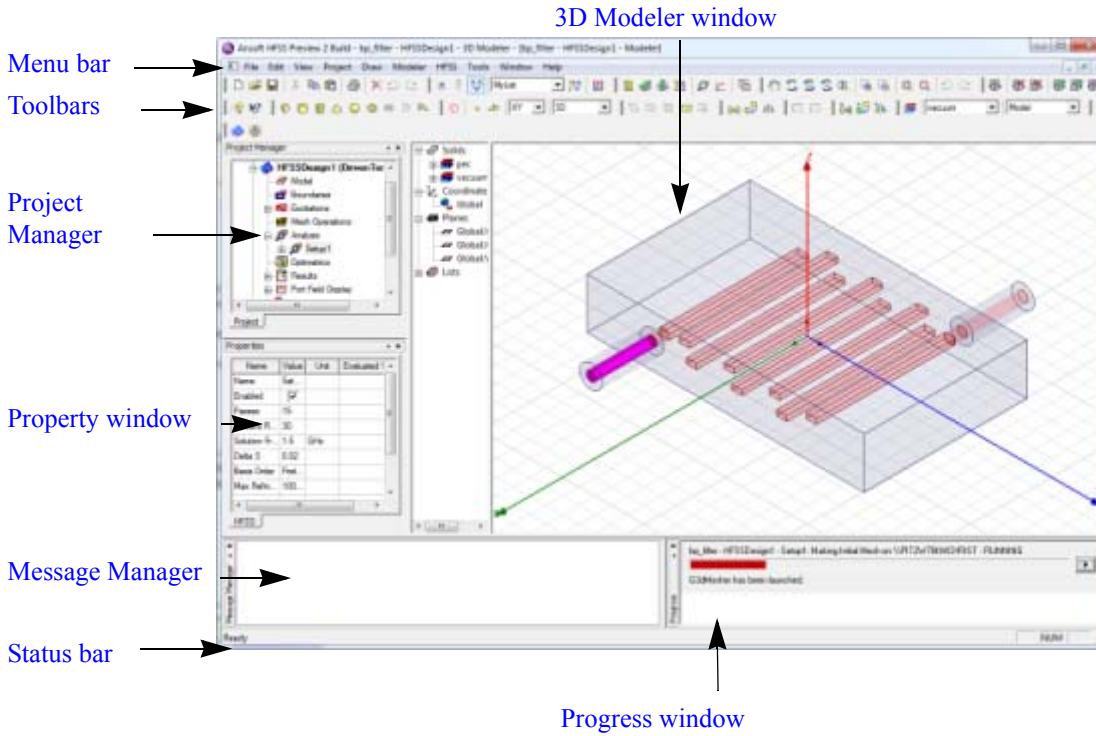


Stripline

The HFSS Desktop

The HFSS desktop consists of several windows, a menu bar, toolbars, and a status bar. You can customize the appearance of the desktop customizing or moving the toolbars, by choosing which windows to display, and by resizing and moving windows.

Click a link below to view more information about that desktop component.



Related Topics

Getting Help

Keyboard Shortcuts for HFSS General Purposes

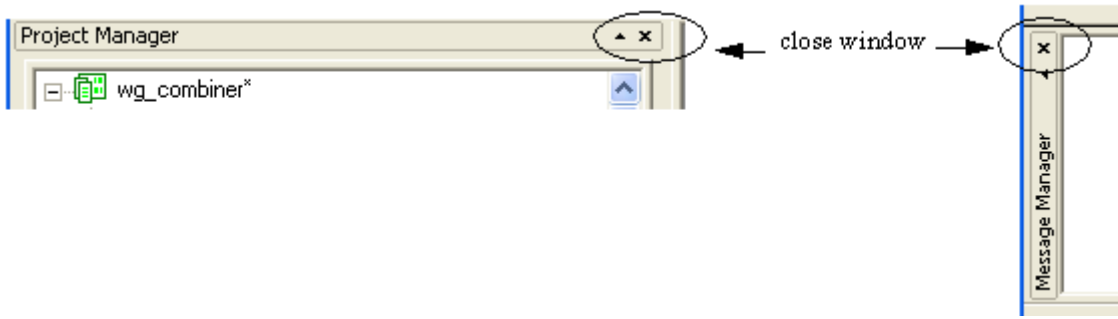
Keyboard shortcuts for the 3D Modeler Window.

Showing and Hiding Windows

The **View** menu contains commands that let you show and hide the windows that comprise the desktop. You can show or hide the Status Bar, the Message Manager, the Project Manager, the docked Properties window, and the Progress window.

The shortcut menu in the toolbar area also lets you show and hide each desktop window.

You can also close the windows by clicking the "x" in the window title bar.



Related Topics

HFSS Desktop

Moving and Resizing Desktop Windows

Moving and Resizing Desktop Windows

You can customize the appearance of the desktop by moving and resizing the Status Bar, the Message Manager, the Project Manager, the docked Properties window, and the Progress window.

To move one of these windows:

1. Click and hold on the title bar.
2. Drag the cursor towards the region where you want to place the window.

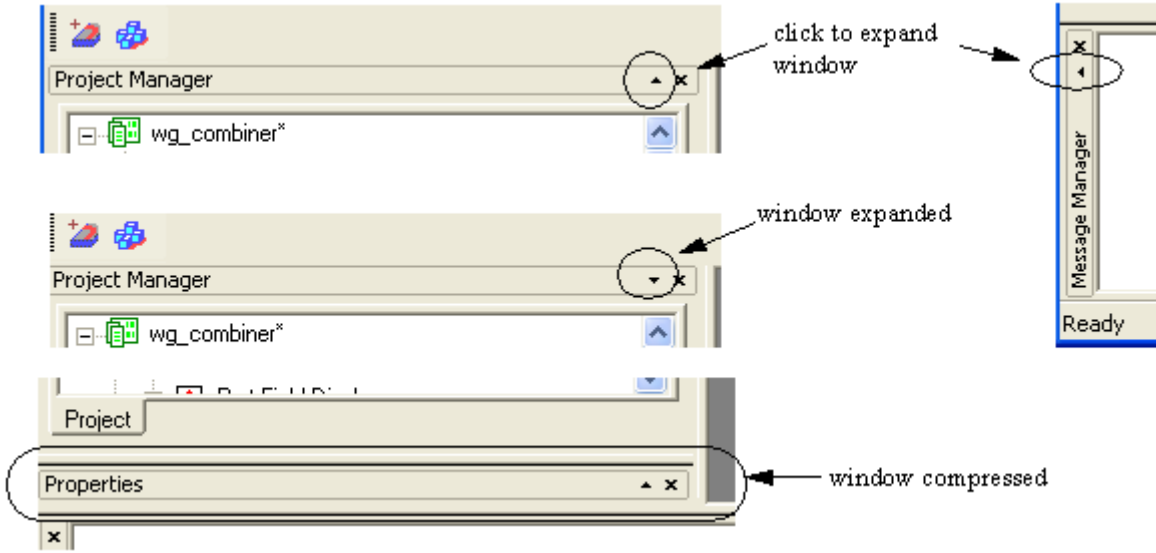
   A rectangle shape follows the cursor. As you drag the rectangle to different parts of the desktop, the changes in dimension show when you have reached a location where you can place the window. This can be at the top, left, bottom, and side of the modeler window. You can place a window next to another, as well as above or below another.

   If you drag the window to the center of the 3D Modeler window, you can place it there as a floating window.
3. Release the mouse button to place the window.

You can also resize the windows in two ways.

- To size a desktop window, place the cursor over an edge of the window. Over the inner-edges, for sizing a window within the desktop, the cursor changes to a double bar with arrows pointing each direction. Over the outer-edges, for sizing the desktop, the cursor changes to a line with arrows pointing each direction. Press and drag to size the window.
- To expand a window to fill the horizontal or vertical space it shares with another window, click the triangle in the window title bar. When you expand a window, the triangle appears as inverted and any other windows in the same horizontal or vertical space are compressed to only the title bar. If a window does not share a horizontal or vertical space with another, the tri-

angle does not appear.



Related Topics

Desktop

Showing and Hiding Windows

Working with the Menu Bar

The menu bar enables you to perform all HFSS tasks, such as managing project files, customizing the desktop, drawing objects, and setting and modifying all project parameters.

To open a help topic about an HFSS menu command, press **Shift**+**F1**, and then click the command or toolbar icon.

HFSS contains the following menus, which appear at the top of the desktop:

File menu	Use the **File** menu commands to manage HFSS project files and printing options.
Edit menu	Use the **Edit** menu commands to modify the objects in the active model and undo and redo actions.
View menu	Use the **View** menu commands to display or hide desktop components and model objects, modify **3D Modeler** window visual settings, and modify the model view.
Project menu	Use the **Project** menu commands to add an HFSS or HFSS-IE design to the active project, view, define datasets, and define project variables.
Draw menu	Use the **Draw** menu commands to draw one-, two-, or three-dimensional objects, and sweep one- and two-dimensional objects.

Modeler menu	Use the **Modeler** menu commands to import, export, and copy 2D Modeler files and 3D Modeler files; assign materials to objects; manage the **3D Modeler** window's grid settings; define a list of objects or faces of objects; control surface settings; perform boolean operations on objects; and set the units for the active design.
HFSS menu	Use the **HFSS** menu to setup and manage all the parameters for the active project. Most of these project properties also appear in the project tree.
Tools menu	Use the **Tools** menu to modify the active project's material library, arrange the material libraries, run and record scripts, update project definitions from libraries, customize the desktop's toolbars, and modify many of the software's default settings.
Window menu	Use the **Window** menu commands to rearrange the **3D Modeler** windows and toolbar icons.
Help menu	Use the **Help** menu commands to access the online help system and view the current HFSS version information.

Related Topics

Getting Help

Keyboard Shortcuts for General Purposes

Keyboard shortcuts for the 3D Modeler Window.

Working with the Toolbars

The toolbar buttons and shortcut pull-down lists act as shortcuts for executing various commands.

To execute a command, click a toolbar button or click a selection on the shortcut pull-down list.

To open a help topic about a toolbar button's functionality, press **Shift**+**F1**, and then click the toolbar button or a command in the shortcut pull-down list.

To display a brief description of the toolbar button, move the pointer over the button or shortcut pull-down list.

Hint To modify the toolbars on the desktop, do one of the following:

- Click **Tools**>**Customize**.
- Right-click the history tree, and then click **Customize** on the shortcut menu.

To reset to toolbars to the default positions and settings:

- Click **Tools**>**Customize**. On the **Customize** dialog box, click **Reset All**.

Related Topics

Customize Toolbar Options

Customize Toolbar Commands

Customize Toolbar Options

To customize the Toolbar displays by using the toolbar list:

1. Select **Tools>Customize**.

   This displays the **Customize** dialog with the **Toolbars** tab selected. The field lists the available toolbars, with those currently selected being checked. To the right of the field are three buttons:

   New... -- launches the New Toolbar dialog that lets you specify a new toolbar name.

   Reset -- This resets the toolbar display to apply your current selections.

   Reset All -- this resets the toolbar display to match the original defaults.
2. Check the buttons to add additional toolbars to the desktop.

   New toolbar icons are added to new rows as you click them. You can drag these to convenient locations.
3. Uncheck any buttons to remove toolbar icons.
4. Use the **OK** button to close the dialog, or the **Cancel** button to close without making changes.

Related Topics

Working with Toolbars

Customize Toolbar Commands

Customize Toolbar Commands

To customize the Toolbar by dragging icons:

1. Select **Tools>Customize**.

   This displays the **Customize** dialog with the **Commands** tab selected. The Categories field lists the available toolbars. The icons for the currently selected toolbar are shown to the right of the field.

2. Select from the Categories list to display the icons you want to add to the toolbar.
3. Drag the icons from the Customize dialog to a location on the desktop toolbar.
4. Use the **OK** button to close the dialog, or the Cancel button to close without making changes.

Related Topics

Customize Toolbar Options

Working with Toolbars

Customizing the Tools Menu: External Tools

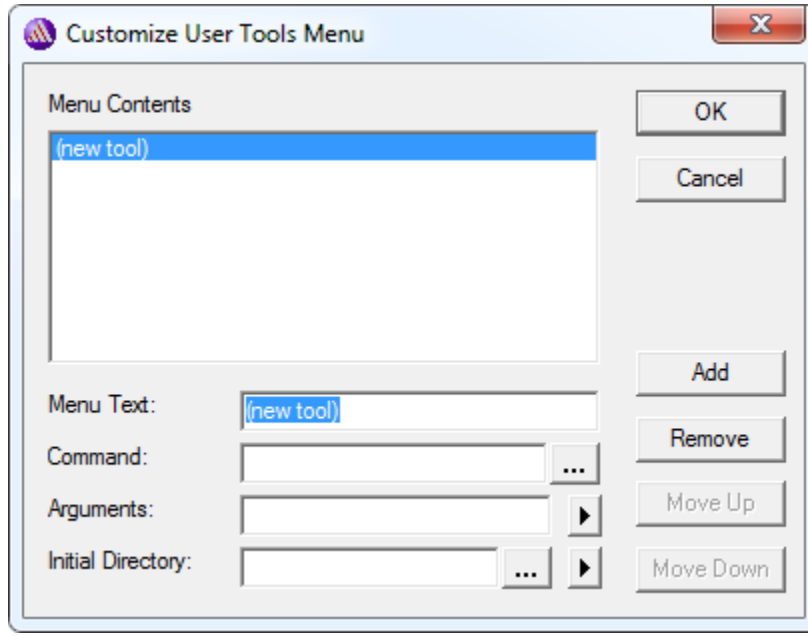
Customizing the Tools Menu: External Tools

To add an executable to the Tools menu:

1. Click **Tools>External Tools**
2. This displays the **Customize User Tools Menu** dialog.

   If a **User Tools** menu item has been defined, its contents are displayed. Command buttons let you Add new commands and Delete selected commands, and Move Up and Move Down commands. This example shows the Reflective.exe program added to the tools menu, and called **Reflective** on the Tools menu. You can specify the command line arguments to the program

and the directory in which it will be run.



3. To add a custom Tools menu entry, click the **Add** button in the **Customize User Tools Menu** dialog.

   This enables the following fields:

   Menu Text field -- this displays [new tool] as text you will replace with the text you want to appear in the User Tools menu.

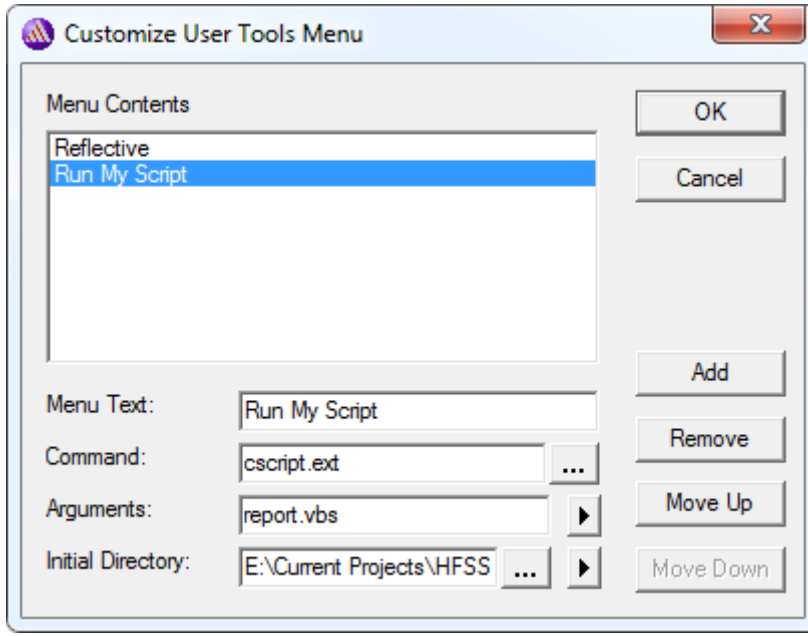
   Command field -- this will display the external executable. An ellipsis button [...] lets you navigate to the file location.

   Arguments field -- this field accepts command arguments from the > button menu selections for File Path, File Directory, File Name, File Extension, Project Directory, or Temp Directory.

   Initial Directory -- this field specifies the initial directory for the command to operate. The ellipsis button {...] displays a dialog that lets you navigate folders in your desktop, or across the network.

4. Click OK to add the External Tools menu to HFSS or Cancel to close the dialog without changes.

You can also add scripts to the **Tools** menu. Assuming you have a script to generate custom reports called report.vbs, use the cscript.exe program to execute your script.



This example shows the cscript.exe program added to the **Tools** menu as **Run My Script**. The command line argument to the csscript.exe program is report.vbs. You can also name the directory in which it will be run.

Related Topics

Scripting

Working with the Shortcut Menus

A variety of shortcut menus — menus that appear when you right-click a selection — are available in the toolbars area of the desktop, in the **3D Modeler** window, and in the **Project Manager** window.

Shortcut menu in the toolbars area	Use the shortcut menu in the toolbars area of the desktop to show or hide windows or toolbars, and customize the toolbars.
Shortcut menu in the 3D Modeler window	Use the shortcut menu in the **3D Modeler** window to select, magnify, and move options (zoom, rotate, etc.), change the view, perform boolean operations, assign materials, boundaries, excitations, or mesh operations to objects, and work with field overlays.
Shortcut menus in the Project Manager window	Use the shortcut menus in the **Project Manager** window to manage HFSS project and design files and design properties; assign and edit boundaries, excitations, and mesh operations; add, analyze, and manage solution setups; add Optimetrics analyses; create post-processing reports; insert far- and near-field radiation setups; edit project definitions; and, run Maxwell SPICE.
Shortcut menus in the History Tree	Use the shortcut menus in the History tree to expand or collapse groupings. If you select particular objects in the history tree, the shortcut menu lists the commands that you can apply to the selected object or objects.

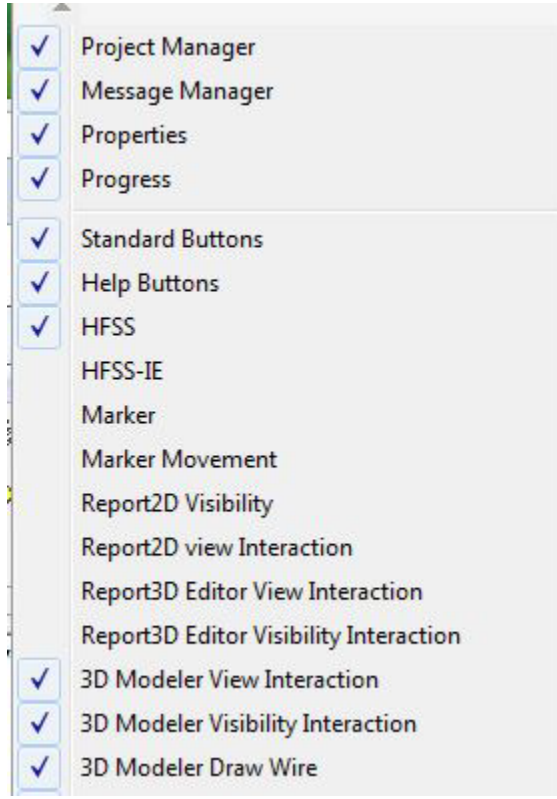
Note All of the commands on the shortcut menus are also available on the menu bar.

Shortcut Menu in the Toolbars Area

Use the shortcut menu in the toolbars area of the desktop to show or hide windows or toolbars, and customize the toolbars.

To access the shortcut menu in the toolbars area:

- Right-click in the toolbars area at the top of the desktop.



A check box will appear next to a command if the item is visible. For example, if a check box appears next to the **Project Manager** command, then the **Project Manager** window is currently visible on the desktop.

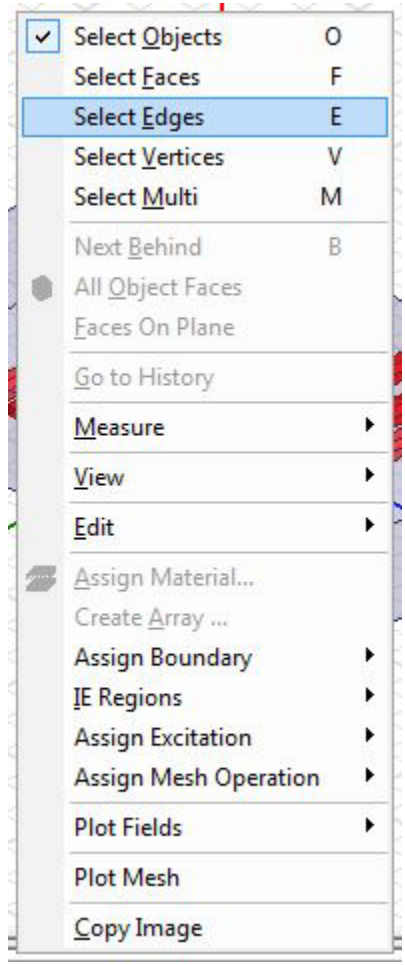
Click **Tools>Customize** to open the **Customize** dialog box, which enables you to modify the toolbar settings on the desktop.

Shortcut Menu in the 3D Modeler Window

Use the shortcut menu in the **3D Modeler** window to select, magnify, and move options (zoom, rotate, etc.), change the view, perform boolean operations, assign materials, boundaries, excitations, or mesh operations to objects, and work with field overlays.

To access the shortcut menu in the **3D Modeler** window:

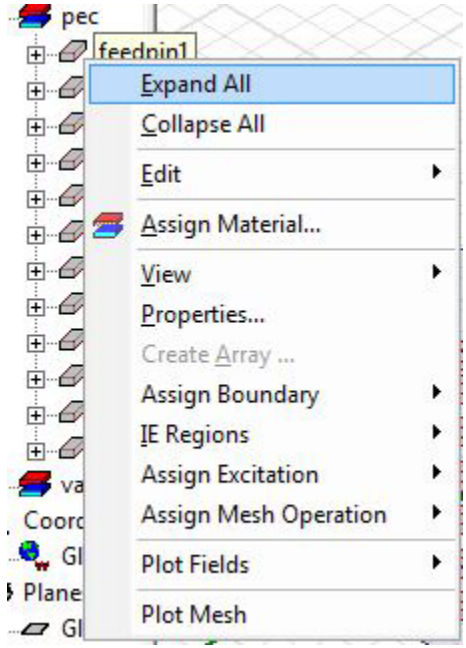
- Right-click in the **3D Modeler** window (grid area).



Shortcut Menus in the Project Manager Window

Each node, or item, in the project tree has a shortcut menu. For example, from the shortcut menu for the Boundaries icon, you can assign boundaries to selected objects; review information for all the boundary assignments for the active design; remove all boundary assignments; show or hide a

boundary's geometry, name, or vectors; change the priority of a previously assigned boundary; and use the **PML Setup** wizard to create a perfectly matched layer (PML) boundary.



Keyboard Shortcuts for HFSS General Purposes

The following keyboard shortcuts apply to HFSS in general

- F1: Help
- F1 + Shift: Context help
- F4 + CTRL: Close program
- CTRL + C: Copy
- CTRL + N: New project
- CTRL + O: Open...
- CTRL + P: Print...
- CTRL + V: Paste
- CTRL + X: Cut
- CTRL + Y: Redo
- CTRL + Z: Undo
- CTRL + 0: Cascade windows
- CTRL + 1: Tile windows horizontally
- CTRL + 2: Tile windows vertically

To customize the shortcut assignments, use Tools>Keyboard Shortcuts.

Related Topics

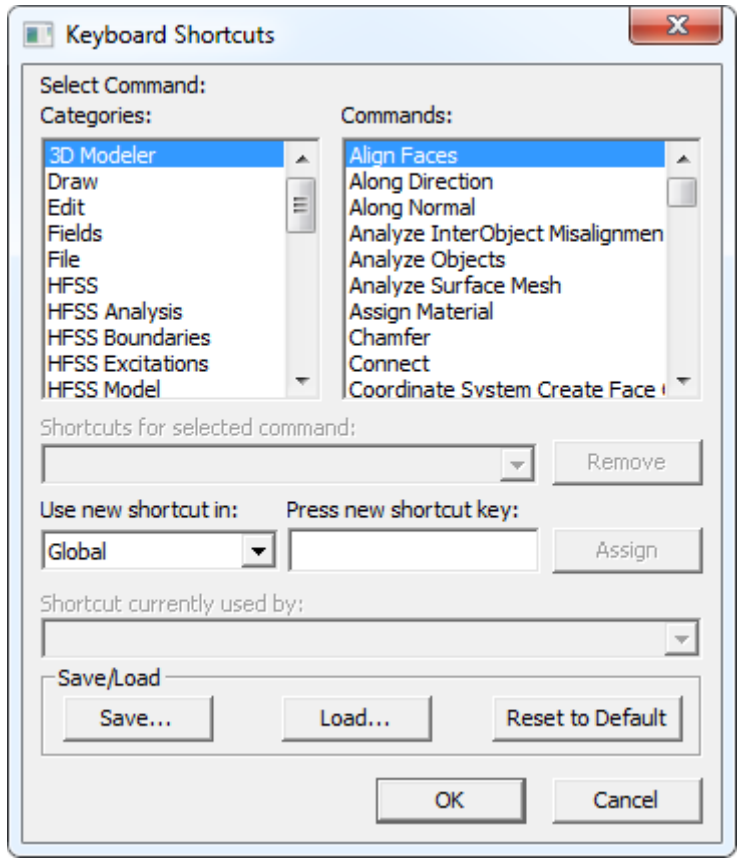
Keyboard Shortcuts for the 3D Modeler Window

Custom Keyboard Shortcuts

Tutorial: HFSS Keyboard Accelerators

Custom Keyboard Shortcuts

Click **Tools>Keyboard Shortcuts** to display a dialog that lets you view existing assignments, create new shortcuts, save and load assignment files.



Selecting a Command category lists the available commands for that category. If the command has an assigned shortcut, it is displayed on the **Shortcuts for the selected command** field. You can use the Remove button to disable the shortcut for the selected command. If the selected command does not have an assigned shortcut, the Shortcuts for selected command field and the Remove button are greyed out.

To create a new shortcut key:

1. Select the Category and Command.
2. If you want disable a current assignment for the selected command, click **Remove**.
3. To assign a keyboard shortcut, place the cursor in the **Press new shortcut key** field.

   The field displays the keystrokes you make. When you have made keystrokes, the dialog enables the Assign button. If you combine keystrokes these are displayed with a "+" between them. For example, Ctrl + p or Alt +o.
4. The **Use new shortcut in** field displays Global by default, which means that the shortcut will apply to all applicable contexts, including HFSS and HFSS-IE. If a limited context exists, the menu will offer a selection.
5. When you have made the assignments you want to make, You can save the assignments to a named file.

   Clicking the **Save...** button displays a browser window that lets you navigate the file structure and assign a name, using an aks suffix for ANSYS Keyboard Shortcut file.

   Buttons on the browser window let you designate the file location as Use Path, Personallib, Syslib, UserLib, or Project folder.

   If you have an existing aks file, you can use the **Load...** button to display a browser window to locate the desired file.
6. You can **OK** the current settings, or **Reset to Default**.

Related Topics

Keyboard Shortcuts for HFSS General Purposes

Keyboard Shortcuts for the 3D Modeler Window

Tutorial: HFSS Keyboard Accelerators

Working with the Status Bar

The status bar is located at the bottom of the application window. It displays information about the command currently being performed.

To display or hide the status bar:

- Click **View**> **Status Bar**.

A check box appears next to this command if the status bar is visible.

Depending on the command being performed, the status bar can display the following:

- **X**, **Y**, and **Z** coordinate boxes.
- A pull-down list to enter a point's absolute, relative, cartesian, cylindrical, or spherical coordinates.
- The model's units of measurement.

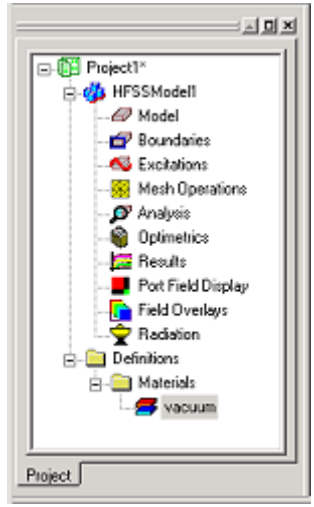
Exiting the Desktop

To exit the desktop, use **File>Exit**.

If simulations are running, a message informs you that if you continue, the simulations will be aborted. If you say OK, HFSS does a clean abort before going on with the Exit.

Working with the Project Manager

The **Project Manager** window displays the open project's structure, which is referred to as the project tree.



The **Project Manager** window displays details about all open HFSS projects. Each project ultimately includes a geometric model, its boundary conditions and material assignments, and field solution and post-processing information.

To show or hide the **Project Manager** window, do one of the following:

- Click **View> Project Manager**.

  A check box appears next to this command if the **Project Manager** window is visible.
- Right-click in the toolbars area on the desktop, and then click **Project Manager** on the short-cut menu.

  A check box appears next to this command if the **Project Manager** window is visible.

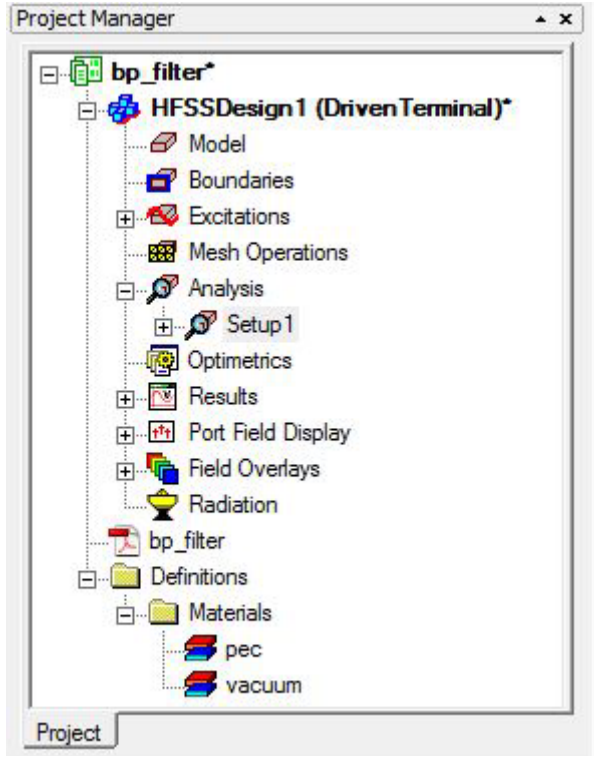
Related Topics

Working with the Project Tree

Shortcut Menus in the Project Manager Window

Working with the Project Tree

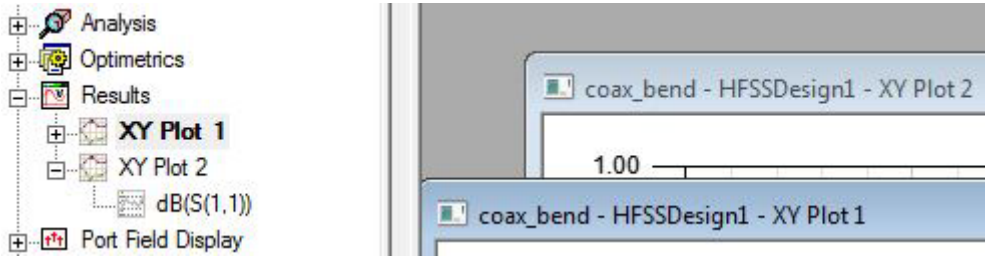
The project tree is located in the **Project Manager** window and contains details about all open HFSS projects, as shown below:



The top node listed in the project tree is the project name. It is named Project*n* by default, where *n* is the order in which the project was added to the current session of HFSS. Expand the project icon to view all the project's HFSS design information and material definitions.

When the Project tree is taller than the window, if you mouse has a scroll wheel, you can click in the Project manager and use the wheel to scroll up and down. If the Project tree is larger than the window size, there is also a scroll bar can control with the mouse cursor.

By default, the Project tree icon for the active window is highlighted, as shown in the figure below. (See General Options: Miscellaneous for the options.) With the option disabled, XY Plot 1 in the Project tree would not be highlighted.



Related Topics

Viewing HFSS Design Details

Setting the Project Tree to Expand Automatically

Viewing Material Definitions Under the Project Tree

Setting the Project Tree to Expand Automatically

You can set the project tree to automatically expand when an item is added to a project.

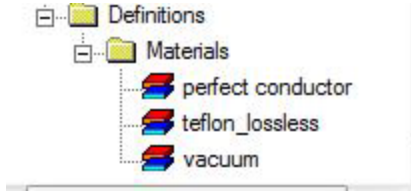
1. Click **Tools** > **Options**>**General Options**.

   The **Options** dialog box appears.

2. Under the **Project Options** tab, select **Expand Project Tree on Insert**.

Viewing Material Definitions

The definitions node is listed at the bottom of the project tree and displays all of the material definitions that are assigned to the objects in the active model.



Related Topics

Adding New Materials

Viewing HFSS Design Details

Once you insert an HFSS design into a project, it is listed as the second node in the project tree. It is named HFSSModel*n* by default, where *n* is the order in which the design was added to the project.

Expand the design icon in the project tree to view all of the specific data about the model, including its boundary conditions and material assignments, and field solution and post-processing information.

The HFSSModel*n* node contains the following project details:

Boundaries	Displays the boundary conditions assigned to an HFSS design, which specify the field behavior at the edges of the problem region and object interfaces.
Excitations	Displays the excitations assigned to an HFSS design, which are used to specify the sources of electromagnetic fields and charges, currents, or voltages on objects or surfaces in the design.
Mesh Operations	Displays the mesh operations specified for objects or object faces. Mesh operations are optional mesh refinement settings that are specified before a mesh is generated.
Analysis	Displays the solution setups for an HFSS design. A solution setup specifies how HFSS will compute the solution.
Optimetrics	Displays any Optimetrics setups added to an HFSS design.
Results	Displays any post-processing reports generated.
Port Field Display	Displays all port fields in the active model.
Field Overlays	Displays field overlay plots, which are representations of basic or derived field quantities on surfaces or objects. Plot folders are listed under **Field Overlays.** These folders store the project's plots and can be customized. See Setting Field Plot Defaults for information on how to customize the plot folders.
Radiation	Displays far- and near-field setups added to an HFSS design.

Note To edit a project's design details:

- In the project tree, double-click the design setup icon that you want to edit.

A dialog box appears with that setup's parameters, which you can then edit.

Viewing the Design List

You can use the **HFSS>Design List** command or the Design List icon to view a dialog with tables of the design properties. The Design list is a dialog that with tabs to let you view the following:

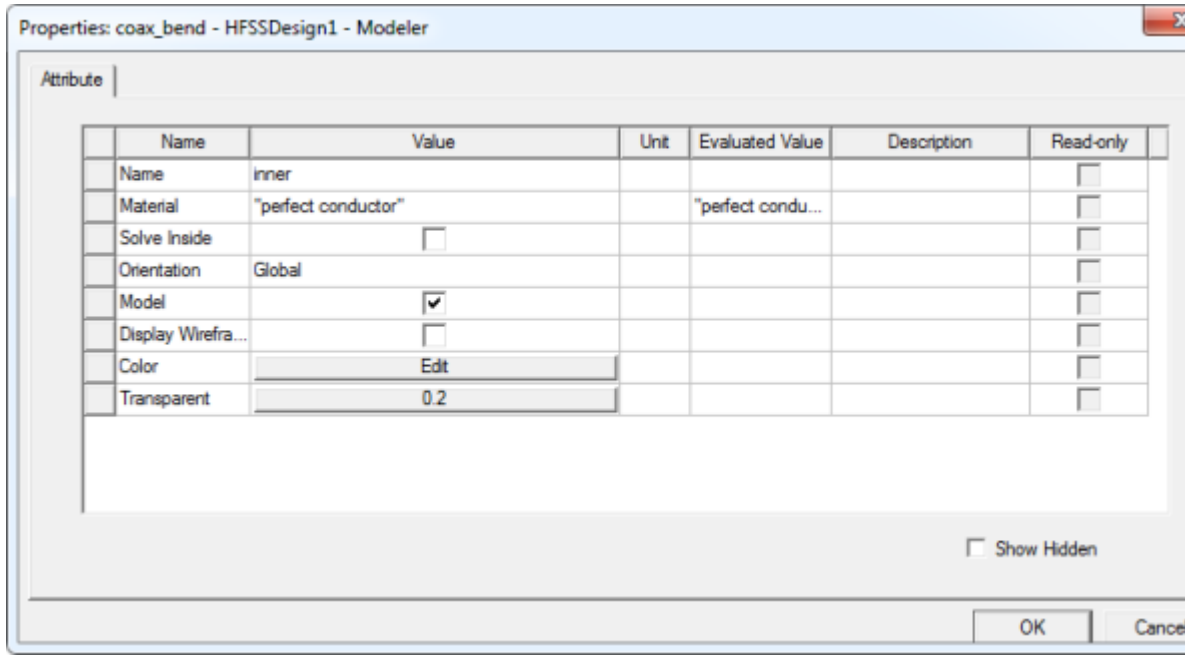
Model	Displays the objects that comprise the model and their properties.
Boundaries	Displays the boundary conditions assigned to an HFSS design, which specify the field behavior at the edges of the problem region and object interfaces.
Excitations	Displays the excitations assigned to an HFSS design, which are used to specify the sources of electromagnetic fields and charges, currents, or voltages on objects or surfaces in the design.
Mesh Operations	Displays the mesh operations specified for objects or object faces. Mesh operations are optional mesh refinement settings that are specified before a mesh is generated.
Analysis Setup	Displays the solution setups for an HFSS design. A solution setup specifies how HFSS will compute the solution.

Working with the Properties Window

The **Properties** window displays the attributes, or properties, of an item selected in the project tree, the history tree, or the **3D Modeler** window. The **Properties** window enables you to edit an item's properties. The properties, and the ability to edit them in the **Properties** window, will vary, depending on the type of item selected. The tabs available in the **Properties** window will also vary, depending on the selection.

By can choose to show or hide a docked **Properties** window as part of the desktop. You can move and resize the docked Properties window within the desktop to suit your work style. When you have a docked Properties window, it displays the properties of any item you select in the Project tree, the History Tree, or the 3D Modeler window. You select **View>Properties Window** to remove the docked properties window.

Regardless of the whether you display a docked **Properties** window, you can still open an undocked **Properties** window for any item in the project tree, the history Tree, or the 3D Modeler window by double-clicking.



Related Topics

Opening the Properties Window

Showing and Hiding the Properties Window

Setting the Properties Window to Open Automatically

Modifying Object Attributes using the Properties Window

Modifying Object Command Properties Using the Properties Window

Opening the Properties Window

1. Select the object whose properties you want to view.
2. Click **Edit>Properties**.
   The **Properties** window for that object appears.
3. When you are finishing making changes, click **OK**.

Rather than opening a separate window, you can also view an object's properties if you have the **Properties** window displayed within the desktop.

Related Topics

Showing and Hiding the Properties Window

Setting the Properties Window to Open Automatically

Modifying Object Attributes using the Properties Window

Modifying Object Command Properties Using the Properties Window

Showing and Hiding the Properties Window

To show or hide the **Properties** window on the desktop, do one of the following:

- Click **View**> **Property Window**.

  A check box appears next to this command if the **Properties** window is visible.
- Right-click in the toolbars area at the top of the desktop, and then click **Properties** on the shortcut menu.

  A check box appears next to this command if the **Properties** window is visible.

Related Topics

Setting the Properties Window to Open Automatically

Setting the Properties Window to Open Automatically

To set the **Properties** window to open after an object is drawn, enabling you to modify the object's properties, do the following:

1. Click **Tools** >**Options**>**Modeler Options**.

   The **Modeler Options** window appears.
2. Click the **Drawing** tab.
3. Select **Edit property of new primitives**.

   Hereafter, after you draw an object in point mode, the **Properties** window will open. However, if you draw an object in Dialog mode, this setting is ignored.

Related Topics

Showing and Hiding the Properties Window

Modifying Object Attributes Using the Properties Window

1. Select the object for which you want to edit its attributes by clicking it in the view window or clicking its name in the history tree.
2. Under the **Attribute** tab in the **Properties** window, edit the object attribute.

   Depending on the attribute type, you can edit it by doing one of the following:
   - Select the check box to apply the attribute; clear the check box to disable the attribute.
   - Click in the field and edit the numeric values or text, and then press **ENTER**. You can modify names, but names must include only letters, numbers and underscores. Illegal names are not accepted and generate a message in the Message window.
   - Click the button and then edit the current settings in the window or dialog box that appears.
   - Click the attribute, and then select a new setting from the menu that appears.

Related Topics

Modifying Objects

Modifying Object Command Properties Using the Properties Window

Modifying Object Command Properties Using the Properties Window

The **Command** tab in the **Properties** window displays information about an action selected in the history tree that was performed either to create an object, such as the **Draw>Box**) command, or to modify an object, such as the **Edit>Duplicate>Mirror** command.

Not all command properties can be modified. In general, the command properties that you can typically modify are the numeric values, such as position values (base position, normal position, start position, etc.), size values (height, radius, etc.), and various other coordinate values. You can also modify many of the unit settings for a command property. You can modify names, but names must include only letters, numbers and underscores. Illegal names are not accepted and generate a message in the Message window.

1. In the history tree, select the command for which you want to edit its properties.

Hint Press and hold **CTRL** to select multiple commands. If you select multiple commands, only the common, or shared, properties will be displayed under the **Command** tab.

2. Under the **Command** tab in the **Properties** window, edit the command's properties.

   Depending on the property type, you can edit it by doing one of the following:

   - Select the check box to apply the property; clear the check box to disable the property.
   - Click in the field and edit the numeric values or text, and then press **ENTER**.
   - Click the button and then edit the current settings in the window or dialog box that appears.
   - Click the attribute, and then select a new setting from the menu that appears.

Related Topics

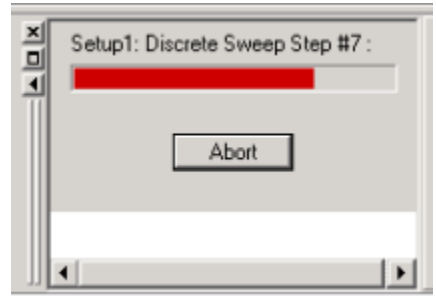
Modifying Objects

Modifying Object Attributes using the Properties Window

Working with the Progress Window

The **Progress** window monitors a simulation while it is running.

In the image below, the **Progress** window is displaying the progress of a discrete frequency sweep, which is nearing completion:



To display or hide the **Progress** window, do one of the following:

- Click **View**> **Progress Window**.

  A check box appears next to this command if the **Progress** window is visible.
- Right-click the history tree, and then click **Progress** on the shortcut menu.

  A check box appears next to this command if the **Progress** window is visible.

Related Topics

Stopping or Aborting Simulation Progress

Viewing Distributed Analysis Subtasks

Stopping or Aborting Simulation Progress

To abort progress, right-click in the Progress window, and select **Abort**.

To stop the simulation cleanly between time steps, right-click in the Progress window, and select **Clean Stop**.

Related Topics

Aborting an Analysis

Viewing Distributed Analysis Subtasks

While a distributed analysis is running, you can access parent and child progress bars. By default, only the main progress bar is displayed, while the child progress bars (or subtasks) remain hidden. You can toggle between showing and hiding the child progress bars.

To show the child progress bars:

- Right-click the progress window, and select **Show Subtask Progress Bars**.

To hide the child progress bars:

- Right-click the progress window, and select **Hide Subtask Progress Bars**.

Working with the Message Manager

The **Message Manager** displays messages associated with a project's development, such as error messages about a design's setup or informational messages about the progress of an analysis. Error messages contains timestamps with a precision of seconds. Each solve of a given frequency, adaptive pass or Optimetrics variation is independent and generates its own messages. This means in some situations, the same error message can be generated multiple times.

To display or hide the **Message Manager**:

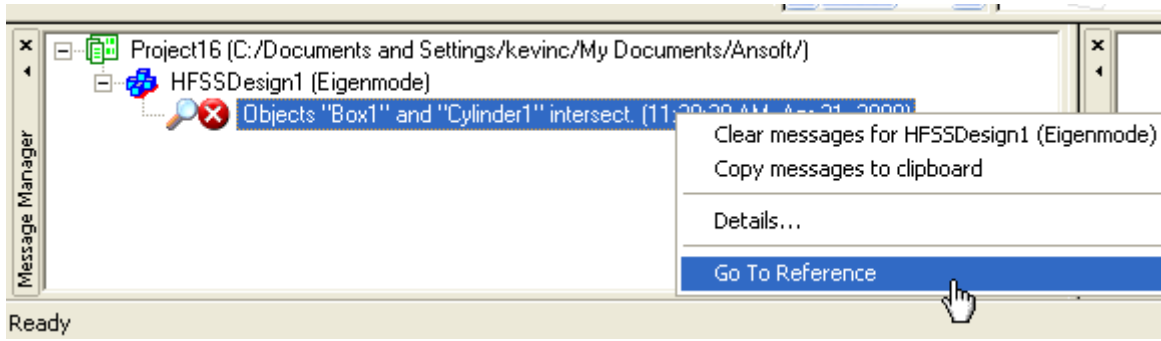
- Click **View**> **Message Manager**.

A check box appears next to this command if the Message Manager is visible.

If you right click in the **Message Manager** window, you see a popup menu that lets you:

- Clear the messages for the current model.
- Copy the messages to the clipboard. This can be helpful for sending the messages to application engineers.
- Details. This brings up a information dialog with the project and design for specific message.
- After you run a validation check, you can right-click on an intersection error message in the

Message window, and select **Go to reference** from the shortcut menu. This selects the intersecting objects.



Related Topics

Showing new messages

Showing errors and warnings

Setting the Message Manager to Open Automatically

Setting the Message Manager to Open Automatically

You can set the Message Manager to open automatically to show new messages and errors and warnings.

Related Topics

Showing new messages

Showing errors and warnings

Showing new messages

You can set the **Message Manager** to automatically be brought up when a new message appears.

1. Click **Tools>Options>General Options**.

   The **General Options** dialog box appears.
2. Click the **Project Options** tab.
3. Under **Additional Options**, select **Show Message Window on new messages**.
4. Click **OK**.

Showing errors and warnings

You can set the **Message Manager** to automatically expand when an item is added to a project.

1. Click **Tools>Options>General Options**.

   The **General Options** dialog box appears.
2. Click the **Project Options** tab.

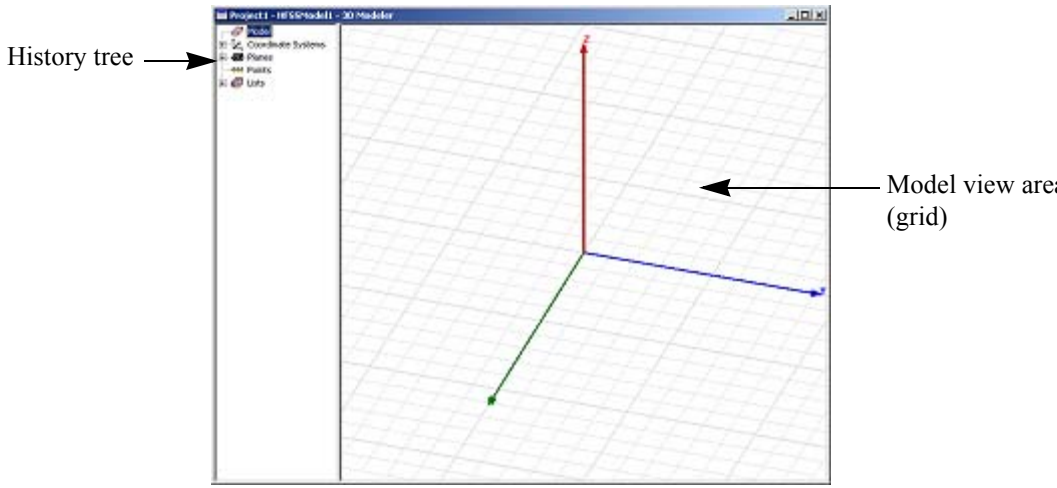
3. Under **Additional Options**, select **Expand Message Window Tree on errors and warnings**.

Click **OK**.

Working with the 3D Modeler Window

The **3D Modeler** window is the area where you create the model geometry. It appears to the right of the **Project Manager** window *after* you insert an HFSS design to a project.

The **3D Modeler** window consists of the model view area, or grid, and the history tree, as shown below:



To open a new 3D Modeler window, do one of the following:

- Insert a new HFSS design into the current project.
- Double-click an HFSS design in the project tree.

The model you draw is saved with the current project when you click **File>Save**.

Objects are drawn in the **3D Modeler** window. You can create 3D objects by using HFSS's Draw menu commands or you can draw 1D and 2D objects, and then manipulate them to create 3D objects. For more information, see Drawing a Model.

You can modify the view of objects in the **3D Modeler** window without changing their actual dimensions or positions. Besides menu and icon commands, you can also use Alt-Click, mouse buttons, and keyboard shortcuts.

For more information, see Modifying the Model View.

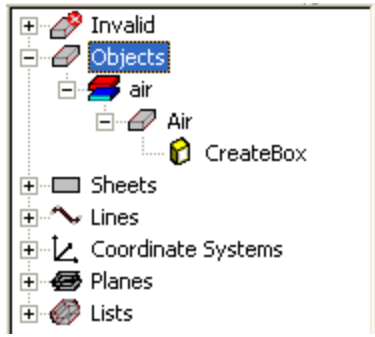
Related Topics

Modifying the Model View

Keyboard shortcuts for the 3D Modeler Window.

Working with the History Tree

The history tree in the **3D Modeler** window lists all the active model's structure and grid details.



You can perform the following tasks with the history tree.

- Expand or Collapse Groupings in the History Tree
- Group Items by Material in the History Tree
- Select Objects in the History Tree
- Use Shortcut Menus in the History Tree
- View Properties for History Tree Objects
- View and Edit Commands on History Tree Objects
- Control the View of Objects in the History Tree
- Select All Objects in a History Tree Group
- Upgrade Version

The history tree contains the following model details:

Invalid	Lists all invalid objects
Objects	Displays all the model's objects and a history of the commands carried out on each object. By default HFSS groups objects by material. you can change this by selecting the **Objects** icon in the history tree and right-click to display the shortcut menu with the **Group Objects By Material** checkbox.
Sheets	Displays all the sheets in the model 3D design area. By default, HFSS groups sheet objects by boundary assignment. You can change this by selecting the Sheet icon in the history tree and right-click to display the shortcut menu with the **Group Sheets by Assignment** checkbox.
Lines	Displays all line objects included in the active model. See Drawing a Straight Line for information on how to draw a line object.
Points	Displays all point objects included in the active model. See Drawing a Point for information on how to draw a point object.

Coordinate Systems	Displays all the coordinate systems for the active model. See Setting Coordinate Systems for more information on this model detail.
Planes	Displays the planes for all the coordinate systems. When you create a coordinate system, default planes are created on its xy, yz, and xz planes.
Lists	Displays the object or face lists for the active model. By default, a list called "AllObjects" appears. Creating an object list is a convenient way to identify a group of objects for a field plot or calculation. Creating a face list is a convenient way to identify a specific set of surfaces for a field plot or calculation. See Creating an Object List

Note While objects created in HFSS can always be classed in the history tree as either a solid, sheet, or wire some imported objects may have mixture of these. HFSS places such objects in an Unclassified folder in the history tree.

Related Topics

Purge History

Generate History

Selecting Several Objects

Creating an Object List

Expand or Collapse Groupings in the History Tree

You can expand or collapse object groupings in the history tree by left clicking the + or - respectively. In addition, right-clicking on any group icon opens a pull-down to expand all groupings or collapse all groupings.

Related Topics

Grouping Items by Material in the History Tree

Selecting Objects in the History Tree

History tree Grouping by Material

In addition, right-clicking on Objects lets you specify whether or not the Objects are sorted by material (the default is to sort by material.) When the objects are sorted by material, 2D and 3D objects are listed separately in the history tree.

Related Topics

Select Objects in the History Tree

Select Objects in the History Tree

Selecting objects in the History tree also selects them in the View window. This can be useful for complex objects, when it may be easier to find the objects of interest by name or material, if the object of interest is inside or behind others.

You can use CTLR-click to make multiple selections. You can select a range of objects by a click on the first, and then SHIFT-Click to select all in the range. You can also use click and drag the mouse to make rubber band selections. Only visible objects are selected. That is, if the hierarchy is closed under the selection, any operand parts are ignored and do not interfere with cut and paste operations.

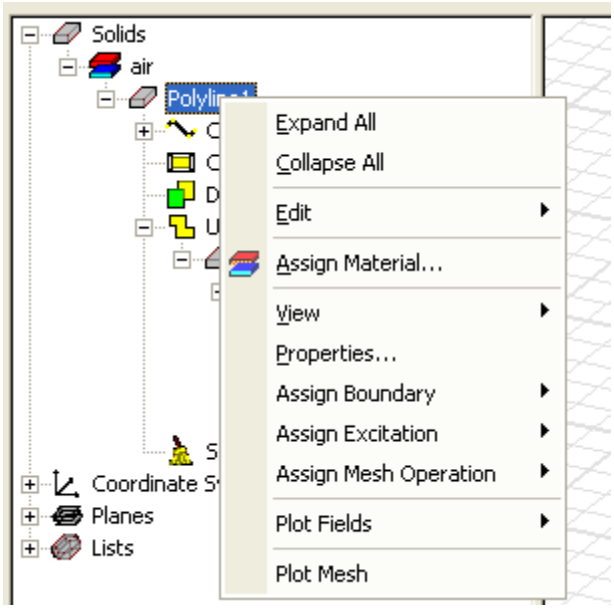
You can also right click on any History tree object or grouping and right click to display a shortcut menu with a Select All command. This lets you select at one time all solids, sheets, lines, non-model objects, or all unclassified objects.

Related Topics

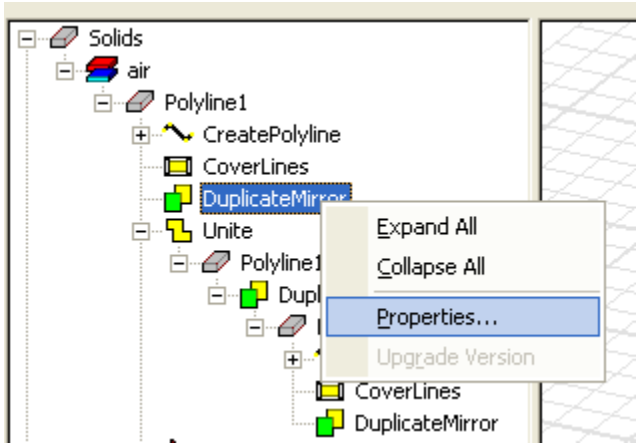
Group Items by Material in the History Tree

Shortcut Menus in the History Tree

If you select particular objects in the history tree, right-click displays a shortcut menu lists the commands that you can apply to the selected command, object, or objects. The shortcut menus for model objects are most extensive.



In other cases, you can only view properties, or expand or collapse hierarchy.



Related Topics

View Properties for History Tree Objects

View and Edit Commands on History Tree Objects

View Item Properties in the History Tree

To view the properties of an item in the history tree:

- Click the item's name in the history tree.

  The item's properties appears in the docked **Properties** window.
- Double-click on an item in the history tree to display a **Properties** window.
- Click the item's name in the history tree, and double right click to display a shortcut menu. Then select **Properties** to display the **Properties** window.

Related Topics

Purge History

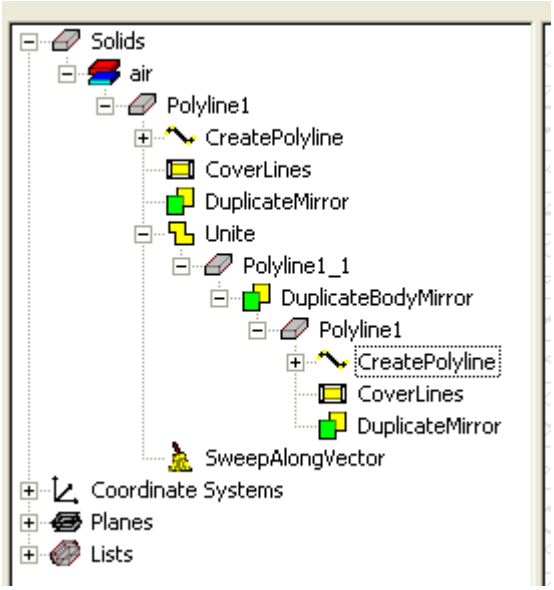
Generate History

Selecting Several Objects

Selecting All from a History Tree Grouping

View and Edit Commands on History Tree Objects

The history tree also lists the history of all commands carried out a model's objects, for example, "CreateBox" or "Subtract." This history is displayed in the order in which it occurred. Here is a history tree from the waveguide combiner example.



Selecting a command in the history highlights the object in the Modeler window and shows that object's properties (if available) in the docked Properties dialog. You can look at the fields in the Properties dialog to see any editable fields for that command, such as coordinate system, line type, coordinates, or units.

For some commands, such as **Edit>Arrange>Move**, or boolean operations, selecting them in the history tree enables the X (delete) icon on the toolbar and the **Edit>Delete** menu. In these cases, you can delete those commands from the history tree as a way of undoing those operations. As an alternative to deletion, you can check **Suppress** command in the Properties window for that command. This undoes the effect of a command on an object without removing it from the History tree.

Related Topics

Select Objects in the History Tree

Purge History command

Generate History command

Modifying Objects

Controlling the View of Objects in the History Tree

To control the view and visibility of an object such as a box or PML, right click on an object in the history tree display the short-cut menu and select **View**. The short cut menu contains the following commands:

- **Fit in Active View**
- **Hide in Active View**
- **Show in Active View**
- **Fit in All Views**
- **Hide in All Views**
- **Show in All Views**

Related Topics

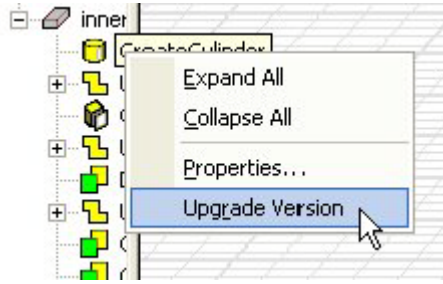
Purge History

Generate History

Selecting Several Objects

Upgrade Version in History Tree Shortcut Menu

By right-clicking on an operation icon in the history tree in the Modeler window, you can see the Upgrade Version command.



The Modeler by default opens the legacy project in previous (what ever that the project was saved with) ACIS version so that you don't see side effects of slight topology changes between ACIS versions. If you want you can use Upgrade Version to move the operation to use latest ACIS code. Typically this applies when a needed fix is available with new ACIS version.

Related Topics

Working with the History Tree

Keyboard Shortcuts for the 3D Modeler Window

By default, the following keyboard shortcuts apply to the 3D Modeler Window

- B: Select face/object behind current selection
- F: Select faces mode
- O: Select objects mode

- E: Select edges mode
- V: Select vertices mode
- M: Multi select mode
- CTRL + A: Select all visible objects
- CTRL + SHIFT + A: Deselect all objects
- CTRL + D: Fit view
- CTRL + E: Zoom in, screen center
- CTRL + F: Zoom out, screen center
- SHIFT + LMB: Zoom in / out
- Alt + double-click to change to one of nine model orientations
- Alt + LMB: Rotate model
- Alt + SHIFT + LMB: Zoom in / out
- Alt + 2xLMB: Sets model projection to standard isometric projections (cursor must be in corner of model screen N/NE/E/SE/S etc)
- F6: Render model wire frame
- F7: Render model smooth shaded

Note LMB means Left Mouse Button

To customize the shortcut assignments, use Tools>Keyboard Shortcuts.

Related Topics

Keyboard Shortcuts for HFSS General Purposes

Custom Keyboard Shortcuts

Using the Password Manager to Control Access to Resources

HFSS lets you specify library resources that require password access, and encryption of those resources. For convenience, the same password can apply to multiple resources. To access the **Password Manager,** click **Tools>Password Manager**. This displays the **Password Manager** dialog.

To Specify a New Password Protected Resource

1. Click **Tools>Password Manager**.

   This displays the **Password Manager** dialog
2. Click the **New** button.

   This opens the **New Encrypted Resource** dialog.
3. Specify the name of the resource that you want to protect and click **OK**.

   This displays the **Enter Passwords** dialog. This dialog has radio buttons to let you:

   - Enter Password and confirm for **Full Access** or for **Execute Only Access**.
   - **Use Ansoft Password** (for execute only). This does not require you to enter a password, but it is still encrypts the library.
4. Once you have selected a radio button, and, if necessary, specified passwords correctly, click **OK**.

   This displays the **Password Manager** with the resource listed.

To Encrypt a Resource

1. Select an existing resource to highlight it and enable the **Encrypt File** button.
2. This displays a File browser window
3. Select the appropriate Files of Type filter.

   The choices are Circuit files (*.lib) and Ansoft Library files. For HFSS, chose Ansoft**hf** Library files. Any existing resources in the selected directory will appear.
4. When you have selected the appropriate resource, click **OK**.

   This encrypts the resource.

Running HFSS from a command line

HFSS includes line arguments that can be included when launching from a command line or terminal prompt. All command-line arguments are case-insensitive. The commands associated batch options can also be used with a Job Management Interface for submitting jobs to ANSYS or RSM and other supported schedulers. See

RSM Integration with Job Management UI

Integration with Microsoft Windows® HPC Scheduler

Integration with Platform's Load Sharing Facility (LSF)

Integration with Grid Engine (GE)

Command-line syntax

hfss <*options*> <*run command*> <*project name/script name*>

Run Commands

The following command line run commands are available in HFSS. Of the commands (BatchSave, BatchSolve, BatchExtract, RunScript, RunScriptandExit), one or none must be used as arguments after **hfss**. Links to the valid options for each run command are listed and/or linked to descriptions.

-BatchSave <*project file name*>

Saves a named project to the current version. This is primarily intended for converting version 9 projects to version 10 when you intend to subsequently run them on a Linux platform. The conversion from version 9 to version 10 must be done under Windows, HP, or Solaris before those projects can run on a Linux system. You can run this command with the **-Iconic** option, the **-Logfile** option, and the **-ng** option (no graphics).

-BatchSolve

By default, solve all adaptive setups, sweeps, as well as Optimetrics setups found in the project file. If parallel solve is possible, you can use the **-Distribute** option in conjunction with **-BatchSolve**. You can run this command with the **-Iconic** option, the **-Logfile** option, the **- ng** option (no graphics), and the **-WaitForLicense** option.

Additional parameters for batch solves include the following. It is good practice to put quotes around the path to the HFSS executable, and the full path to the project. This ensures that spaces in the path or project will not be an issue. The same is true of the design name, if there are indeed spaces. The quotes must enclose the entire argument including the Nominal or Optimetrics part.

[***designName***] - batch solve all setups for design with the name given under the project.

[***designName***]:**Nominal** - batch solve all nominal setups for design with the name given under the project.

[***designName***]:**Optimetrics** - batch solve all Optimetrics setups for design with the name given under the project.

[***designName***]:**Nominal**:[***setupname***] - batch solve the specified nominal setup for design

with the name given under the project. The setupname is case insensitive.

[***designName***]:**Optimetrics**:[***setupname***] - batch solve the specified Optimetrics setup for design with the name given under the project. The setupname is case insensitive.

-Local | -Remote | -Distributed

Perform the **-Batchsolve** on a local machine, a remote machine, or as a distributed solve using a specified machine list (see below). These command line options are mutually exclusive. That is, only one of these options should be specified. The settings persist only for the current session.

If you specify **-Local**, a machine list is not needed.

For **-Remote**, you should provide a machine list with a single hostname.

For **-Distributed**, you should provide a machine list or file path.

-Distributed takes optional arguments which modify the job distribution parameters. When the optional parameters are not present, the behavior is single level distributed solves with no change in order of precedence among possible distribution types.The optional parameters are:

includetypes= <default> | <distribution type 1, distribution type 2, ...>

If the distribution types are specified, only the listed distribution types are enabled. If default is specified, a default set of enabled distribution types will be used.

excludetypes= <default> | <distribution type 1, distribution type 2, ...>

If the distributed types are specified, all distribution types except those listed will be enabled. If default is specified, a default set of enabled distribution types will be used.

maxlevels= 1 | 2

This is the maximum level of job distribution. Right now, only single and double level distribution is supported.

numlevel1= number of level 1 tasks

When two level distribution is enabled, **numlevel1** specifies the number of level 1 tasks.

When using **-Distributed**:

- If neither **includetypes=** or **excludetypes=** are specified, default job distribution types will be used.
- If **maxlevels** is not specified, multilevel distribution will be disabled.

See the **Job Distribution** tab in the **Analysis Configuration** dialog or the -help text for valid values for job distribution types.

The **-Machinelist** command line option is used to specify the DSO Machines. This command line option is only meaningful if **-Distributed** is specified. There are three different formats for the **-Machinelist** command line option. The different formats are described below.

-MachineList list="*<machine1>, <machine2>, ...*"

In this format, the DSO machines machine1,machine2,... are listed explicitly on the command line. The machine names must be separated by commas. The machines may be specified by IP address or by hostname, provided that the hostnames are able to be resolved on the Desktop host. If the list contains any whitespace, it must be enclosed in quotes. The number of distributed COM engines run on each host is equal to the number of times that the hostname appears in the list. That is, if host1 appears in the list once, and host2 appears in the list twice, then one COM engine will run on host1 and two COM engines will run on host2.

list =<machine1, machine2, ...> accepts machine specifiers of the form:

<machine name>:<total number of tasks>:<total number of cores>.

For example:

```
list="Orion:4:8, Aries:3:12, Pluto:6:12"
```

With this form, duplicate machine names are not allowed, and the number of cores must be greater than the number of tasks.

-MachineList file=*"<machine list file path>"*

In this format, the DSO machines are listed in a file. The machine names are listed in the text file, one hostname per line. The pathname of the file is file_path_name. The machines may be specified by IP address or by hostname, provided that the hostnames are able to be resolved on the local host. The number of distributed COM engines run on each host is equal to the number of times that the hostname appears in the list. That is, if host1 appears in the list once, and host2 appears in the list twice, then one COM engine will run on host1 and two COM engines will run on host2.

file=<machine list file path> will also accept machine specifiers in the specified file using the format.

<machine name>:<total number of tasks>:<total number of cores>.

For example:

```
"Orion:4:8, Aries:3:12, Pluto:6:12"
```

With this form, duplicate machine names are not allowed, and the number of cores must be greater than the number of tasks.

You can use either form of the **-MachineList** option to indicate the machine(s) on which to run a distributed batchsolve. The settings persist only for the current session.

When you use a file to define the machines available for a distributed solve you should list the machine addresses or names on separate lines:

```
192.168.1.1
192.168.1.2
(etc)
```

-MachineList num = *<numb distributed engines>*

The "-machinelist num=n" option is now required for batch jobs. This format is used when

a scheduler (such as LSF, PBS, SGE or Windows HPC) is used to manage the jobs sent to a cluster of hosts. In a scheduler environment, you can specify the number of distributed engines that should be used for distributed processing. In this case, you do not specify the machine names after the flag because the names are provided by the scheduler. For example, in the Windows HPC environment, you can write the number of distributed engines as follows.

```
-machinelist num=4
```

The COM engines will be distributed across the hosts allocated to the job by the scheduler.

-batchoptions *'"<option1>' '<option2>'..."*

All options that are specified through **Tools>Options** dialogs go to the user-level registry. You can override such registry entries via the **-batchoptions** command line. These overrides apply only to the current Desktop session. This feature is available for all desktop products. The registry setting overrides may be specified on the command line, or may be in a file with the file pathname specified on the command line. The **-batchoptions** command line option is only valid for batch jobs; it is ignored if neither **-BatchSolve** nor **-BatchSave** command line options are specified.

Examples and Further Explanations of -batchoptions use

BatchExtract <*batchExtractScriptFile*> <*projectFile*>

This command allows the following commands to be executed non-graphically via script and without checking out any GUI licenses: ExportProfile, ExportConvergence, ExportMeshStats, ExportNetworkData, ExportNMFData, ExportEigenmodes, ExportTransientData, Update Reports, ExportToFile. A project file *must* be specified when the command line option **BatchExtract** is used. This means that commands in the <*batch extract script file*> will only be executed in the specified project. The "open/close" project commands are not supported in **BatchExtract** mode.

Note

- **-ng** *must* be used with BatchExtract or it will fail with an error.
- Including unsupported commands in the *batchExtractScriptFile* will terminate the script execution.

Examples:

- **-ng -BatchExtract <*batchExtractScriptFile*> <*projectFile*>**
- **-ng -BatchSolve ... -BatchExtract <*batchExtractScriptFile*> <*projectFile*>**

  The commands in *batchExtractScriptFile* will be executed after **BatchSolve** is done and before the project is saved.

  Note that **BatchSolve** will continue to require solve licenses.

Example Script For Report Export:

```
hfss -ng -batchextract exportToFile.py "C:\Program
Files\AnsysEM\HFSS15.0\Win64\Examples\HFSS\opti_new.hfss"
```

where `exportToFile.py` contains:

```
oDesktop.RestoreWindow()
```

```
oProject = oDesktop.SetActiveProject("opti_new")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oModule = oDesign.GetModule("ReportSetup")
oModule.UpdateReports(["XY Plot 1"])
oModule.ExportToFile("XY Plot 1", "exportToFilePy.csv")
```

-Monitor

You can monitor progress and messages on standard output, during non-graphical analysis. Progress, warning and info messages are logged to the standard output stream. Error and fatal messages are logged to the standard error stream. Schedulers intercept these streams and provide commands for display of this output - see individual scheduler documentation for specifics.

```
Examples:
C:\HFSS\hfss.exe -distributed
-machinelist list="192.168.1.1,192.168.1.2"
-batchsolve design_transient:Optimetrics
"C:\distrib_project.adsn"

C:\HFSS\hfss.exe -batchsolve HFSSDesign1:Nominal
"C:\Project1.hfss"

"c:/Program Files/AnsysEM/HFSS14.0/hfss.exe"
-Iconic -Queue
-LogFile "H:\HFSS\_HFSSQueue\fence-v2__Array with Fence4.log"
-BatchSolve "Array with Fence4:Nominal" "H:\HFSS\fence-v2.hfss"
```

-RunScript <*script file name*>

Run the specified script. You can use the **-ScriptArgs** option to add one or more arguments to this command and can use the **-Iconic** option.

-RunScriptAndExit <*script file name*>

Run the specified script and exit. You can use the **-ScriptArgs** option to add one or more arguments to this command. You can also use the **-Iconic** option, the **-Logfile** option, and the **-WaitForLicense** option. The '**-BatchSolve** <DesignName>' is mutually exclusive with '**-RunScriptAndExit** <ScriptName>'.

<none>

If you do not specify a run command with hfss on the command line, you can still specify the **-Help** and **-Iconic** option.

<project file>

Open the specified project on start up. If **-BatchSolve** is also set, the project will be

solved.

Note The <project file> must be the last command line entry.

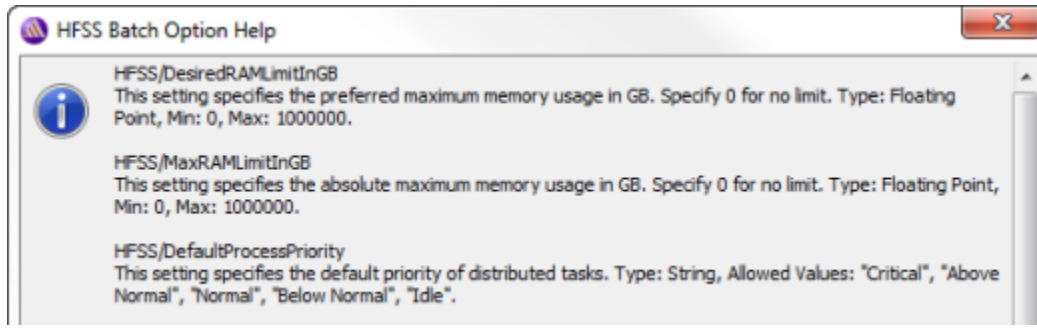
Options

The following options can be associated with one or more of the run commands.

-batchoptionhelp

Open a window listing the -batchoptions help. For instance:

```
E:\Program Files\AnsysEM\AnsysEM15.0\Win64\hfss -
batchoptionhelp
```



-Distribute

Distribute a batch solve to multiple machines. This option must be combined with the **-BatchSolve** run command and must be specified before it in the command line. See Distributed Analysis for more information on distributed analysis.

Example:

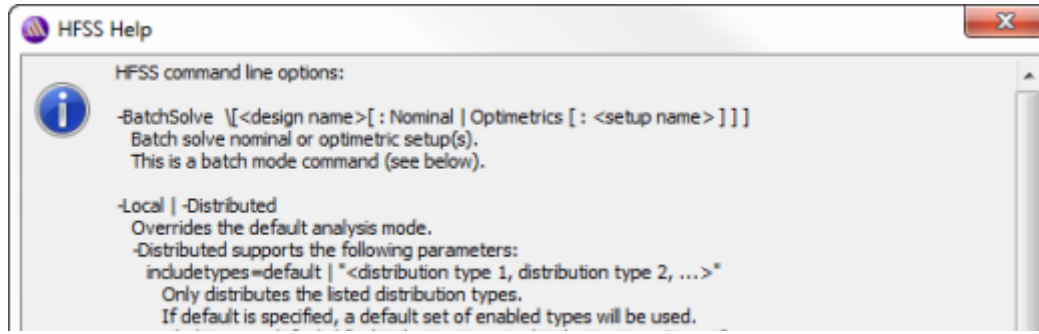
```
C:\HFSS\hfss.exe -distribute -batchsolve
HFSSDesign1:Optimetrics:ParametricSetup1 "C:\Project1.hfss"
```

-Help

Open a window that displays the different command-line options. This is only used when none of the four run commands are used.

For example:

```
E:\Program Files\AnsysEM\AnsysEM15.0\Win64\hfss -help
```



-Iconic

Run HFSS with the window iconified (minimized). This can be used with all or none of the run commands.

-LogFile <*log file name*>

Specify a log file (use in conjunction with **-BatchSave** or **-BatchSolve** or **-RunScriptAndExit** run commands). If no log file is specified, it will be written to the directory in which the script or HFSS project is located, with the name <project_name>.log.

-ng

Run HFSS in non-graphical mode (Use in conjunction with **-BatchSave** or **-BatchSolve** run commands. *Must* be used with **-BatchExtract** or **-BatchSolve** command.).

-WaitForLicense

Wait for unavailable licenses (use along with **-BatchSolve** or **-RunScriptAndExit**).

-ScriptArgs <*scriptArguments*>

Add arguments to the specified script in conjunction with **-RunScript** and **-RunScriptAndExit**.

ScriptArgs looks at the single argument after it and uses those as script arguments. You can pass multiple arguments to scriptargs by surrounding the script arguments in double quotes.

For instance:

```
hfss -scriptargs "HFSSDesign1 Setup1" -RunScriptAndExit
c:\temp\test.vbs
```

Here, HFSSDesign1 is taken into HFSS as the first argument, and Setup1 is the second argument. Without the quotes, HFSSDesign1 is taken as the first argument, and Setup1 will not be understood by HFSS.

```
hfss -scriptargs HFSSDesign1 Setup1 -RunScriptAndExit
c:\temp\test.vbs
```

Example:

```
c:\hfss\hfss.exe -runscriptandexit "c:\project1.vbs"
```

```
    -scriptargs "Setup1"
Example:
    C:\hfss\hfss.exe -RunScriptAndExit C:\scripts\test.vbs
    -scriptargs "arg1 arg2 arg3"
```

Settable Desktop Settings for -batchoptions Use

For -batchoptions Use: Project Directory and Lib Paths

For -batchoptions Use: TempDirectory

For -batchoptions Use: Various Desktop Settings

For -batchoptions Use: HFSS and HFSS-IE Options with Paths

Batch Options Command Line Examples

The Tools>Options>Export Options Files command writes xml files containing the Options settings at all levels to the specified directory (default, ~\Documents\Ansoft\). The **Tools>Options>Export Options** feature is intended to make it easier for different users to use ANSYS Electromagnetics Suite 15.0 installed on shared directories or network drives. The Example Uses for Export Options Features section outlines some use cases enabled by this feature.

Related Topics

Running a Script.

Command Line Enhancements for ANSYS EM Desktop Products

Large Scale DSO for Parametric Analysis

Examples and Further Explanations of -batchoptions use

This section provides further examples and explanations of -batchoptions.

- Example with registry settings specified on the command line
- Example with registry settings specified in a file
- -batchoptions Uses Relative Registry Paths
- When to Use the -batchoptions Desktop Command Line Option

The following examples use hfss, but this feature is available for all desktop products.

- The registry path separator is the slash "/"
- Registry key pathnames are enclosed in single quotes
- Registry string values are enclosed in single quotes
- Backslashes in registry key values must be escaped with another backslash

Example with registry settings specified on the command line

```
hfss.exe -batchsolve -batchoptions
"'Desktop/Settings/ProjectOptions/NumberOfProcessors'=4
```

```
'Desktop/ProjectDirectory'='C:\\projects\test'"
projectname.hfss
```

This command line overrides the values of the Project Options and Desktop/ProjectOptions registry settings.

Notes:

- Multiple registry settings may appear in a single -batchoptions value, separated by whitespace
- The -batchoptions value must be enclosed in double quotes if it contains any whitespace

Example with registry settings specified in a file

```
hfss.exe -batchsolve -batchoptions filename projectname.hfss
```

where file *filename* contains:

```
$begin 'Config'
   'Desktop/Settings/ProjectOptions/NumberOfProcessors'=4
   'HFSS/NumCoresPerDistributedTask'=2
   'Desktop/ProjectDirectory'='C:/projects/test'
$end 'Config'
```

This command line overrides the values of the Desktop/Settings/ProjectOptions/NumberOfProcessors and Desktop/ProjectOptions registry settings. These overrides apply only to the current hfss session.

Note

- The -batchoptions filename value must be enclosed in double quotes if it contains whitespace
- The $begin 'Config' and $end 'Config' lines are required

-batchoptions Uses Relative Registry Paths

When using the -batchoptions command line option, the registry paths specified on the command line or in the batchoptions file are relative paths. The paths are relative to the current version of the current product. If the examples above are used with HFSS 2014, then the following table shows the relative and absolute paths of the registry overrides in the above examples.

Relative Path	Absolute Path for HFSS
Desktop/Settings/ProjectOptions/ NumberOfProcessors	ProgramFiles/AnsysEM/AnsysEM15.0/Win64/HFSS/ Desktop/Settings/ProjectOptions/NumberOfProcessors
Desktop/ProjectDirectory	ProgramFiles/AnsysEM/AnsysEM15.0/Win64/HFSS/ Desktop/ProjectDirectory

For additional options and paths for use with -batchoptions, see

For -batchoptions Use: Project Directory and Lib Paths

For -batchoptions Use: TempDirectory

For -batchoptions Use: Various Desktop Settings

For -batchoptions Use: HFSS and HFSS-IE Options with Paths

When to Use the -batchoptions Desktop Command Line Option

Analysis parameter settings may be done using the GUI. For example, all HFSS options may be set using the **Add Batch Option** dialog box which is brought up through the **Submit Job To** dialog. These parameter settings include the following solver options (not a complete list):

- tempdirectory
- DesiredRAMLimitInGB
- MaxRAMLimitInGB
- DefaultProcess Priority

HFSS and HFSS-IE include a -batch option for distributed memory:

- MPIVendor (either "Intel" or "Platform Computing".

These values of these parameters are saved in the registry when HFSS is not running.

When running a batch analysis, these parameters will take the values from the registry. The **-batchoptions** Desktop command line option allows you to override the parameter values set in the registry with values specified on the command line or in a file. The values specified using the **-batchoptions** command line option only apply to the batch job, and do not affect the parameter values in the registry. For example, you could specify the following command to ensure that this analysis uses 2 processors for distributed processing and 2 processors for non-distributed processing. If the **-batchoptions** value is not specified, then the number of processors for distributed processing or for non-distributed processing could be affected by an interactive HFSS job running on the same host as the same user.

```
"E:\Program Files\AnsysEM\AnsysEM15.0\Win64\hfss.exe"
-distributed includetypes=default maxlevels=1
-machinelist num=4 -monitor -ng
-batchoptions " 'HFSS/DefaultProcessPriority'='Normal'
'HFSS/DesiredRAMLimitInGB'=8
'HFSS/HPCLicenseType'='Pool'
'HFSS/MaxRAMLimitInGB'=16
'HFSS/NumCoresPerDistributedTask'=2
'HFSS-IE/NumCoresPerDistributedTask'=2
-batchsolve E:\Users\kmchrist\Documents\Ansoft\hfss_bpf.hfss
```

Related Topics

Running HFSS from a Command Line

For -batchoptions Use: Project Directory and Lib Paths

The PersonalLib, syslib and userlib settings are a little different from other settings. If the final directory name is different from what is expected, then PersonalLib, syslib or userlib is appended

as a final directory. In addition, these settings may come from a different registry value if the registry values shown above are not set

Path Name	Default Value	Units or Values	Description
Desktop/ ProjectDirectory	subdirectory of user's HOME directory or "My Documents" directory	Directory pathname	Directory where new projects are created
Desktop/ PersonalLib	PersonalLib subdirectory of user's HOME directory or "My Documents" directory	Directory pathname	Directory PersonalLib is appended if final directory is not PersonalLib
Desktop/syslib	syslib subdirectory of installation directory	Directory pathname	Directory syslib is appended if final directory is not syslib
Desktop/userlib	userlib subdirectory of installation directory	Directory pathname	Directory userlib is appended if final directory is not userlib

Related Topics

For -batchoptions Use: TempDirectory

For -batchoptions Use: Various Desktop Settings

For -batchoptions Use: HFSS and HFSS-IE Options with Paths

Running HFSS from a Command Line

For -batchoptions Use: TempDirectory.

Path Name	Default Value	Units or Values	Description
TempDirectory	Set by installer	-	Directory for temporary files

Related Topics

For -batchoptions Use: Project Directory and Lib Paths

For -batchoptions Use: Various Desktop Settings

For -batchoptions Use: HFSS and HFSS-IE Options with Paths

Running HFSS from a Command Line

For -batchoptions Use: Various Desktop Settings

Note that the most of these only affect the GUI.

Note The preferred IP address and preferred subnet address settings are mutually exclusive. If both are specified to be non-empty strings, then the preferred IP address takes precedence, and the preferred subnet address is ignored. This feature is typically used for cluster environments using batch solves. The setting can be made via batchoptions but can also be done via UpdateRegistry.

Path Name	Default Value	Units or Values	Description
Desktop/Settings/ ProjectOptions/ AnimationMemory	200	Megabytes (MB)	Stop animations when available memory falls below this value
Desktop/Settings/ ProjectOptions/ AnsoftCOMPreferredIPAddress	"" (empty string)	IP Address (as a string)	IP Address used to connect from COM Engine to Desktop
Desktop/Settings/ ProjectOptions/ AnsoftCOMPreferredSubnetAddress	"" (empty string)	Subnet address	Specify the subnet that engines use to connect to the Desktop. Allowed formats are: 1) IPv4 dotted decimal network prefix [example: 123.123.123.0] , 2) IPv4 network prefix and network mask separated by a slash character "/" [example: 123.123.123.0/ 255.255.255.0], or 3) IPv4 network prefix and prefix length in CIDR notation [example: 123.123.123.0/24]
Desktop/Settings/ ProjectOptions/ AutoSaveInterval	10	edits	Number of edits to allow between autosaves
Desktop/Settings/ ProjectOptions/ AutoShowMessageWindow	1 (true)	0 (false) or 1 (true)	Show message window on new messages

Path Name	Default Value	Units or Values	Description
Desktop/Settings/ ProjectOptions/ AutoShowProgressWindow	0 (false)	0 (false) or 1 (true)	Show progress window when starting a simulation
Desktop/Settings/ ProjectOptions/ DiskLimitForAbort	0	Megabytes (MB)	A warning is issued when available disk space falls below this value
Desktop/Settings/ ProjectOptions/DoAutoSave	1 (true)	0 (false) or 1 (true)	Enables autosaves if true
Desktop/Settings/ ProjectOptions/ DrawStateIconsInProjectTree3	1 (true)	0 (false) or 1 (true)	Change icon when selection does not match active window
Desktop/Settings/ ProjectOptions/ ExpandMessageTreeOnInsert	1 (true)	0 (false) or 1 (true)	Ensure that new messages are visible in the message window tree
Desktop/Settings/ ProjectOptions/ExpandOnInsert	0 (false)	0 (false) or 1 (true)	Expand project tree on insert
Desktop/Settings/ ProjectOptions/HighlightActive ContextInProjectTree2	1 (true)	0 (false) or 1 (true)	Emphasize active command context (menu and toolbars)
Desktop/Settings/ ProjectOptions/ SavePreviewImagesInProject File	1 (true)	0 (false) or 1 (true)	Save preview images in project file
Desktop/Settings/ ProjectOptions/ UpdateReportOnFileOpen	0 (false)	0 (false) or 1 (true)	Update reports on file open

Related Topics

For -batchoptions Use: Project Directory and Lib Paths

For -batchoptions Use: TempDirectory

For -batchoptions Use: HFSS and HFSS-IE Options with Paths

Running HFSS from a Command Line

For -batchoptions Use: HFSS and HFSS-IE Options with Paths

Here are the HFSS Options and HFSS IE Options with the complete path names:

Path Name	Default Value	Units or Values	Description
Hfss/ HPCLicenseType	"pack"	"pool" or "pack"	For HFSS Designs: "pool" = multiprocessing enabled by HPC licensing; "pack" = multiprocessing enabled by HPC Pack licensing
Hfss/ MaxRAMLimitInGB	0 (No Limit)	GB	This setting specifies the absolute maximum memory usage in GB. Type: Floating Point, Min: 0, Max: 1000000.
Hfss/ DesiredRAMLimitInGB	0 (No Limit)	KB	Preferred memory use in GB. Type: Floating Point, Min: 0, Max: 1000000.
Hfss/ NumCoresPerDistributedTask	1	-	Number of cores that used per distributed task when using a machine list provided by the scheduler. Type: Integer, Min: 1, Max: 1000.
HFSS/MPIVendor		"Platform Computing", "Intel".	Which MPI libraries to use for remote communication. Type: String,
HFSS/ AllowGPUForTransient	0	0 (False), 1 (True).	Allow GPU to be used for transient solves.
HFSS/ SolveAdaptiveOnly		0 (False), 1 (True).	If true, only the adaptive portion of the setup will be solved
HFSS-IE/ HPCLicenseType	"pack"	"pool" or "pack"	For HFSS-IE Designs: "pool" = multiprocessing enabled by HPC licensing; "pack" = multiprocessing enabled by HPC Pack licensing
HFSS-IE/ MaxRAMLimitInGB	0 (No Limit)	GB	Maximum RAM Limit

Path Name	Default Value	Units or Values	Description
HFSS-IE/ DesiredRAMLimitIn GB	0 (No Limit)	GB	Desired RAM Limit
HFSS-IE/ NumCoresPerDistrib utedTask	1	-	For HFSS-IE Designs: number of cores that will be used per distributed task when using a machine list provided by the scheduler
HFSS-IE/MPIVendor		"Platform Computing", "Intel".	Which MPI libraries to use for remote communication. Type: String,

Related Topics

For -batchoptions Use: Project Directory and Lib Paths

For -batchoptions Use: TempDirectory

For -batchoptions Use: Various Desktop Settings

Running HFSS from a Command Line

Running from a Windows Remote Terminal

When running HFSS or HFSS-IE from a remote terminal, there are some performance and behavior issues to consider. These issues are due to the interaction of bandwidth/opengl drivers/remote-terminal-protocol

- Showing axes when interactively drawing objects will slow the performance.
- Remote OpenGL performance will be slower in general. Graphics card and driver quality helps.
- All 3D windows will be closed when you switch from remote PC to a console or from a console to remote. This is to avoid display/opengl instability during the switch.
- Grid will not be turned off while viewing a plot from a remote desktop. The mouse over highlights on 2D plots may appear as not totally overlapping the line color or as thin dotted lines.

Related Topics

Remote Solve

Modifying the Model View

Windows HPC Commands

HPC Integration allows you to submit jobs directly using ANSYS Electromagnetics command line arguments for batchsolves. The supported HPC software is described in the ANSYS Electromagnetics Installation Guide. ANSYS Electromagnetics products must be accessible from the same directory on all machines. The ANSYS Electromagnetics command line syntax is documented here. You must pass in a -distributed flag as part of the ANSYS Electromagnetics command line arguments if you want to run a distributed simulation.

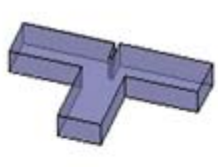
Before running a job you must you **Tools>JobManagement>Select Scheduler** and use the dialog to designate the head node of a cluster. You can then click **Tools>Job Mangement>Submit Job** to submit the batch commands for the job.

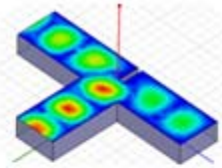
Related Topics

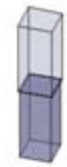
High Performance Computing (HPC) Integration

Getting Started Guides

The HFSS installation includes getting started guides that describe the following designs.

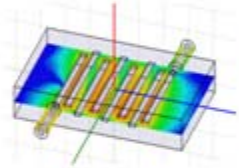

A Waveguide T-Junction

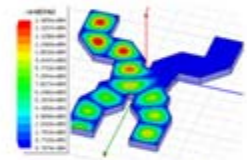

Optimizing a Waveguide T-Junction

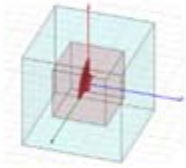

Floquet Ports


A Dilectric Resonator Antenna

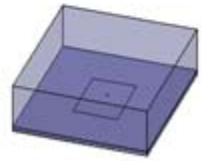

A Coax Connector


Bandpass Filter

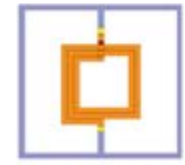

Waveguide Combiner


Radar Cross Section


UHF Probe


Patch Antenna


Coax Tee


Spiral Inductor

A list of example projects included with the HFSS installation is located here.

A brief Application Specific Modeling Guide: Antennas.

An Application Specific Guide: Spiral Inductors on Silicon Substrate:

Guides for HFSS-IE are here.

Guides for HFSS Transient are here.

For an example patch antenna project emphasizing Keyboard Accelerators:
Open the PDF:

.

A 20 GHz Waveguide Combiner

Open the PDF:

.

This *Getting Started* guide is written for HFSS beginners as well as experienced users who are using HFSS for the first time. This manual guides you through the setup, solution, and analysis of a two-way, low-loss waveguide combiner.

By following the steps in this guide, you will learn how to perform the following tasks in HFSS:

- Draw a geometric model.
- Modify a model's design parameters.
- Assign variables to a model's design parameters.
- Specify solution settings for a design.
- Validate a design's setup.
- Run an HFSS simulation.
- Create a 2D x-y plot of S-parameter results.
- Create a field overlay plot of results.
- Create a phase animation of results.

Antennas: Application Specific Modeling Guide

Open the PDF:

.

This Application Specific *Modeling Guide* is written for HFSS beginners as well as experienced users who are using HFSS for the first time. This brief manual provides guidelines for modeling antennas in HFSS.

Related Topics

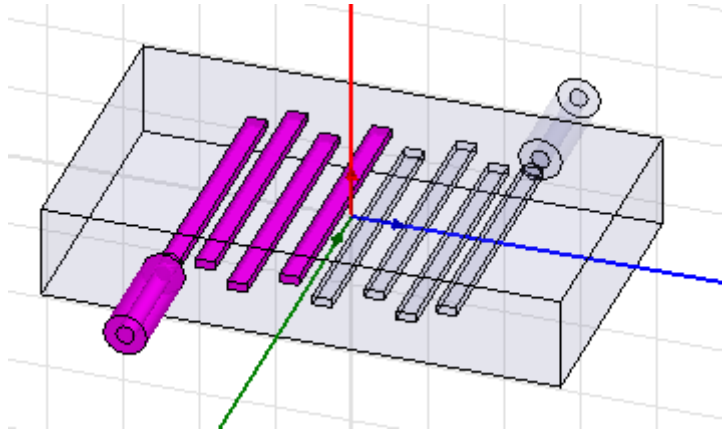
Antennas in HFSS

A Bandpass Filter

Open the PDF:

.

This *Getting Started* guide assumes some familiarity with HFSS. It guides you through the process of creating a bandpass filter. It includes the use of duplicate around axis commands.

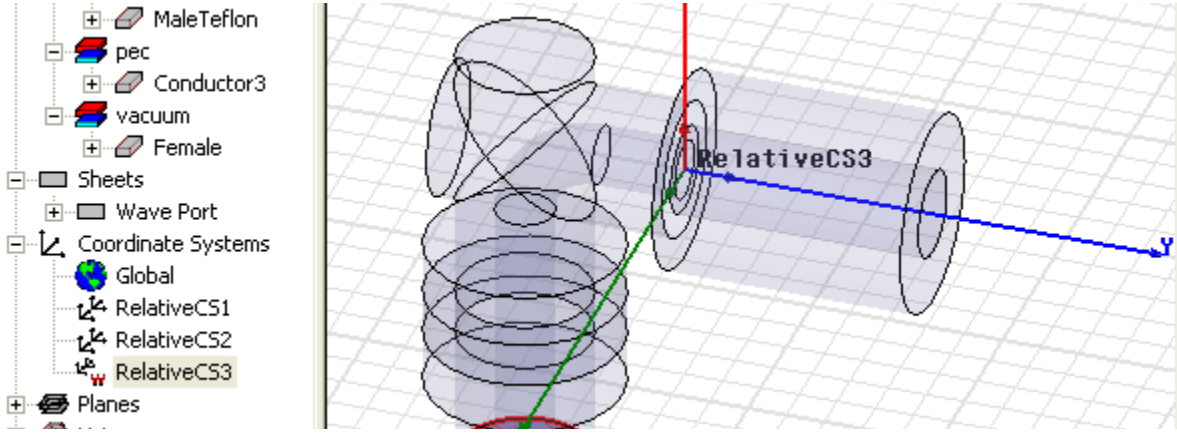


This design has a frequency sweep. If you want to distribute the frequencies for efficient simulation you can use HPC. For more information about how to setup HPC, see the **Add HPC Analysis Setup** section in the **Getting Started Guide for Bandpass Filter.** You can access the design from the Examples folder which also contains a short description on the HPC setup in the associated PDF.

A Coax Connector

Open the PDF:

This *Getting Started* guide assumes some familiarity with HFSS. It guides you through the process of creating a coax connector. This includes the use of relative coordinate systems as an aid to building the model. It also uses boolean and sweep operations.



Coax Tee Model

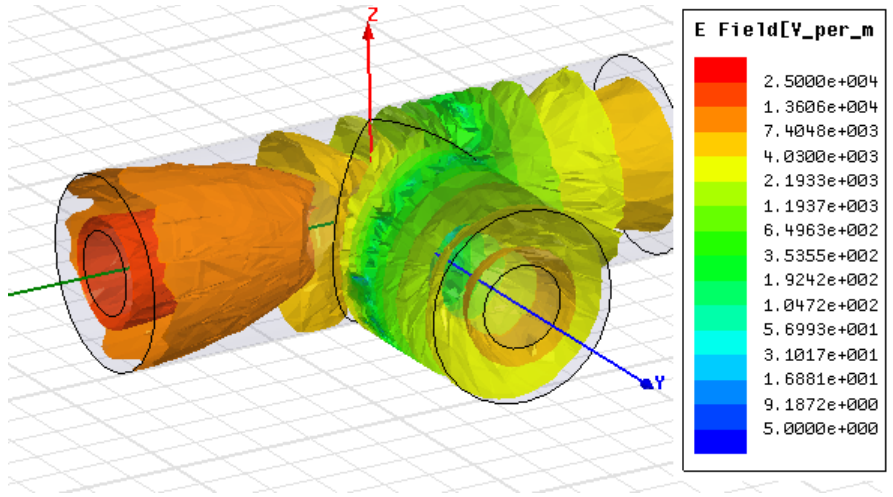
Open the PDF:

.

This *Getting Started* guide assumes some familiarity with HFSS. It guides you through the steps of creating and simulating a coax tee. The simulation solves for the fields in an arbitrary volume.

Coax Dielectric

- Coax Center Pin
- Outer Boundary
- Coax Shield



A Dielectric Resonator Antenna

Open the PDF:

.

This *Getting Started* guide is written for HFSS beginners as well as experienced users who are using HFSS for the first time. This guide leads you step-by-step through creating, solving, and analyzing the results of a dielectric resonator antenna problem.

By following the steps in this guide, you will learn how to perform the following tasks in HFSS:

- Draw a geometric model.
- Modify a model's design parameters.
- Assign variables to a model's design parameters.
- Specify solution settings for a design.
- Validate a design's setup.
- Run an HFSS simulation.
- Create a 2D x-y plot of S-parameter results.

- Create a field overlay plot of results.
- Create a phase animation of results.

Floquet Port Models

Open the PDF:

.

This *Getting Started* guide is written for HFSS beginners as well as experienced users who are using HFSS for the first time. This manual guides you through the setup, solution, and analysis of two different models using Floquet ports.

By following the steps in this guide, you will learn how to setup Floquet ports in HFSS.

Optimizing A Waveguide Tee Junction

Open the PDF:

.

This *Getting Started* guide is written for Optimetrics beginners as well as experienced users who are using Optimetrics for the first time. You must have completed *Getting Started with HFSS: A Waveguide T-Junction* before you begin this guide.

You will use ANSYS Electromagnetics Optimetrics software to find an optimal position for the septum. Prior to performing the optimization, you will set up and solve a parametric analysis.

By following the steps in this guide, you will learn how to perform the following tasks in HFSS using Optimetrics:

- Create a basic parametric setup.
- Solve a parametric analysis.
- Create a 2D x-y plot of S-parameter results.
- Create a 2D x-y plot of power distribution results.
- Create a geometry animation.
- Specify a variable to be optimized.
- Create an optimization setup, which includes defining a cost function and setting the range of variable values for an optimization.
- Solve an optimization analysis.
- During an optimization analysis, view a plot of cost values versus solved iterations.
- Run an HFSS simulation using the optimal variable value.
- Update an existing field overlay plot with new results.

Related Topics

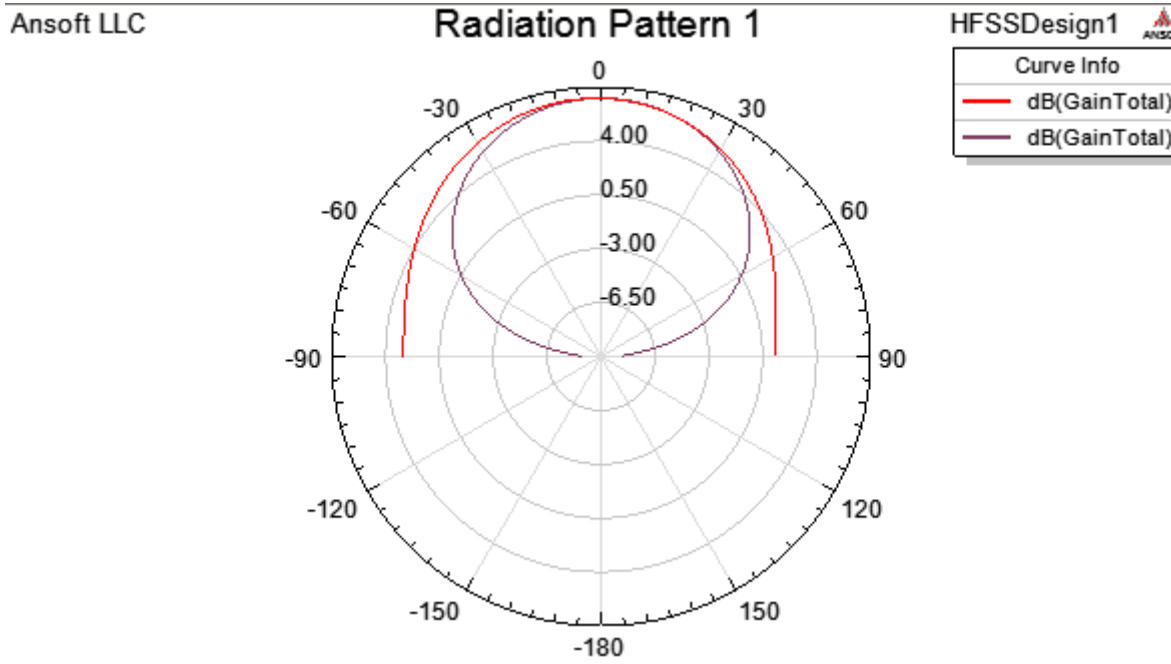
Optimetrics

A Patch Antenna

Open the PDF:

.

This *Getting Started* guide assumes some familiarity with HFSS. It includes the use of Perfect E and Radiation boundaries and a Radiation Pattern plot.



Related Topics

Example Projects

Getting Help

RCS Model

Open the PDF:

.

This *Getting Started* guide is written for HFSS beginners as well as experienced users who are using HFSS for the first time. This manual guides you through the setup, solution, and analysis a simple radar cross section (RCS) problem. By following the steps in this guide, you will learn how to perform the following tasks in HFSS:

- Draw the geometric models.
- Create the Perfectly Matched Layer (PML) Boundaries

- Add the Excitation
- Setup Mesh Operations
- Specify solution setting for the design.
- Validate the design setups.
- Run HFSS simulations.
- Create the geometry setups for monostatic and bistatic infinite spheres.
- Create plots for these geometries for a Normalized Bistatic RCS and Monostatic RCS.

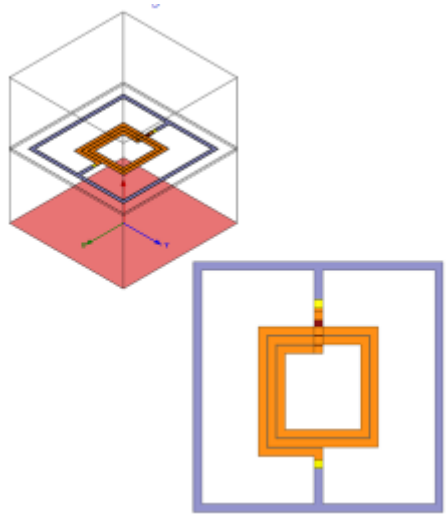
By following the steps in this guide, you will learn how to setup RCS problems.

Silicon Spiral Inductor

Open the PDF:

.

This *Getting Started* guide assumes some familiarity with HFSS. It includes the use of Perfect E and Radiation boundaries and Output Variables in generating plots.



Related Topics

Example Projects

Application Specific Modeling Guide: Spiral Inductors on Silicon Substrate

Spiral Inductors on Silicon Substrate: Application Specific Modeling Guide

Open the PDF:

.

This Application Specific *Modeling Guide* is written for HFSS beginners as well as experienced users who are using HFSS for the first time. This brief manual provides guidelines for modeling spiral inductors in HFSS.

Related Topics

Drawing a Spiral Using User Defined Primitives

Drawing a Spiral

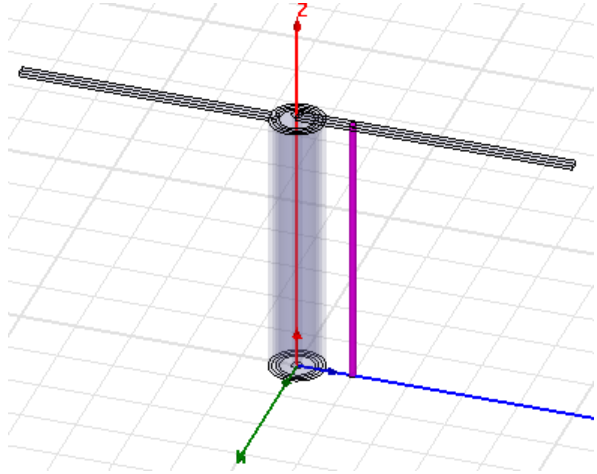
Setting the Reference Point

A UHF Probe

Open the PDF:

.

This *Getting Started* guide assumes some familiarity with HFSS. It includes the use boolean operations, and the use of a ground plane and radiation boundaries.



A Waveguide Tee Junction

Open the PDF:

.

This *Getting Started* guide is written for HFSS beginners as well as experienced users who are using HFSS for the first time. This guide will lead you step-by-step through creating, solving, and analyzing the results of a T-shaped waveguide with an inductive septum. This type of structure is used to split an incoming microwave signal into two outgoing signals. The waveguide's transmission and reflection of the signal will depend on the position of the septum.

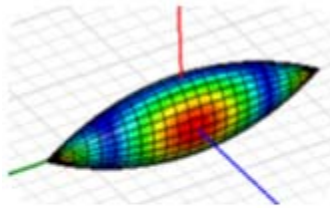
By following the steps in this guide, you will learn how to perform the following tasks in HFSS:

- Draw a geometric model.

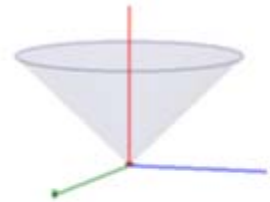
- Modify a model's design parameters.
- Assign variables to a model's design parameters.
- Specify solution settings for a design.
- Validate a design's setup.
- Run an HFSS simulation.
- Create a 2D x-y plot of S-parameter results.
- Create a field overlay plot of results.
- Create a phase animation of results.

HFSS-IE Getting Started Guides

The HFSS-IE installation includes the following getting started examples:



Ogive RCS



Monocone Antenna

Open the PDF:

.

This *Getting Started* guide is written for HFSS-IE beginners as well as experienced users who are using HFSS for the first time. This manual guides you through the setup, solution, and analysis a simple radar cross section (RCS) problem. By following the steps in this guide, you will learn how to perform the following tasks in HFSS-IE:

- Draw the geometric models.
- Add the Excitation
- Specify solution setting for the design.
- Run HFSS-IE simulations.
- Create the geometry setups for infinite spheres.
- Create plots for RCS.

By following the steps in this guide, you will learn how to setup RCS problems.

Open the PDF:

.

This *Getting Started* guide is written for HFSS-IE beginners as well as experienced users who are using HFSS for the first time. This manual guides you through the setup, solution, and analysis a simple monocone antenna problem. By following the steps in this guide, you will learn how to perform the following tasks in HFSS-IE:

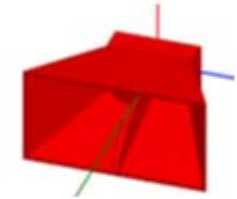
- Draw the geometric models.
- Add the excitation and infinite ground plane.
- Specify solution setting for the design.
- Run HFSS-IE simulation.
- Create the geometry setups for infinite spheres.

- Create a plot for the radiation pattern.

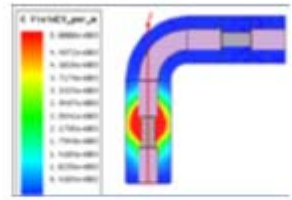
By following the steps in this guide, you will learn how to setup HFSS-IE antenna problems.

HFSS Transient Getting Started Guides

The HFSS Transient installation includes the following getting started examples.



Horn Antenna Model



TDR for Coax Bend

Ball Grid Array

Open the PDF:

.

This *Getting Started* guide is written for HFSS Transient beginners as well as experienced users who are using HFSS for the first time. This manual guides you through the setup, solution, and analysis of a transient horn antenna problem. By following the steps in this guide, you will learn how to perform the following tasks in HFSS:

- Import the model
- Add the boundaries and the excitation
- Specify solution setting for the design.
- Run HFSS Transient simulations.
- View reports
- Create the geometry setups for infinite spheres.
- Create field plots.

By following the steps in this guide, you will learn how to setup transient antenna problems.

Related Topics

Transient Solution Type

Open the PDF:

.

This *Getting Started* guide is written for HFSS Transient beginners as well as experienced users who are using HFSS for the first time. This manual guides you through the setup, solution, and analysis of a time delayed response problem for a coax bend. By following the steps in this guide, you will learn how to perform the following tasks in HFSS:

- Draw the geometric models.
- Add the boundary and excitation.

- Specify solution setting for the design.
- Run HFSS transient simulation.
- Create a plots and animate a field overlay.

By following the steps in this guide, you will learn how to setup HFSS transient problems.

Related Topics

Transient Solution Type

Open the PDF:

.

This *Getting Started* guide is written for HFSS Transient beginners as well as experienced users who are using HFSS for the first time. This manual guides you through the setup, solution, and analysis of a transient simulation. By following the steps in this guide, you will learn how to perform the following tasks in HFSS:

- Draw the geometric models.
- Add the boundaries and excitation.
- Specify solution setting for the design.
- Run HFSS transient simulation.
- Create a plots, including the use of Output Variables.

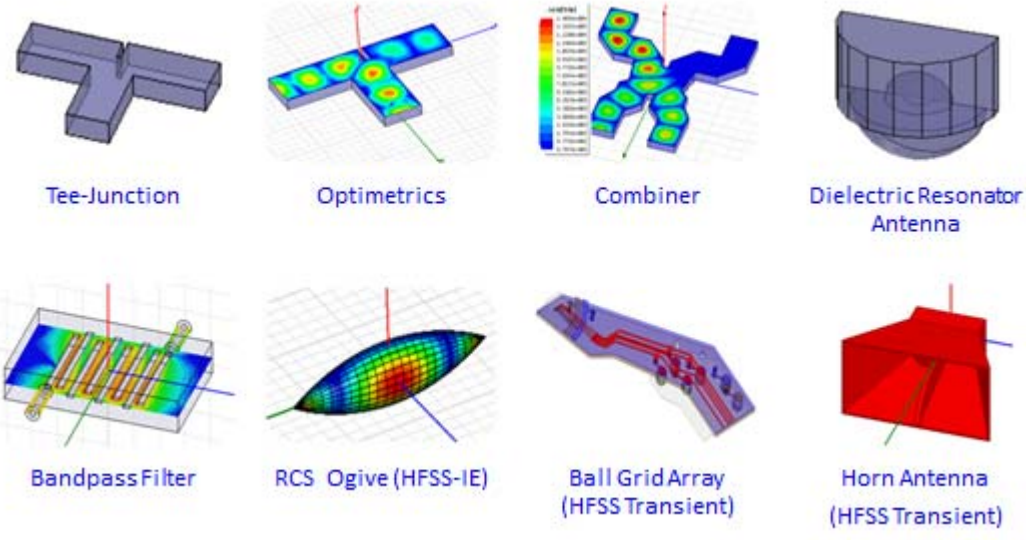
By following the steps in this guide, you will learn how to setup HFSS transient problems.

Related Topics

Transient Solution Type

Example Projects

Your HFSS installation includes an example directory containing a projects folder including the following projects associated with Getting Started Guides: (See associated Getting Started Guides)



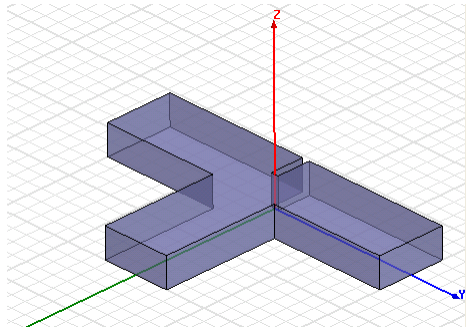
Several other example projects don't have full getting started guides but do include descriptions in the online help. See Other Examples with Brief Descriptions in the Online Help

Related Topics

Opening Example Projects

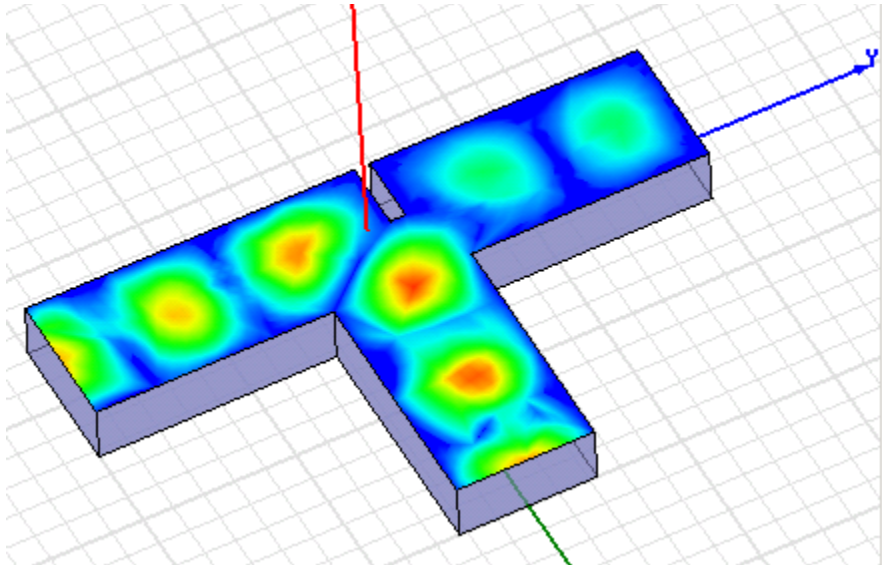
Tee and OptimTee Waveguide Projects

The Getting Started folder in the Examples/Projects folder contains the versions of the waveguide t-junction modal solution project described in *Getting Started with HFSS: A Waveguide T-Junction*, and *Getting Started with HFSS: Optimizing a Waveguide T-Junction Using HFSS with Optimetrics*.



The waveguide T-junction illustrates the basic HFSS features, including :

- the Modeler
- parameterization of a design feature
- setup and analysis
- the use of the Reporter and field animation.

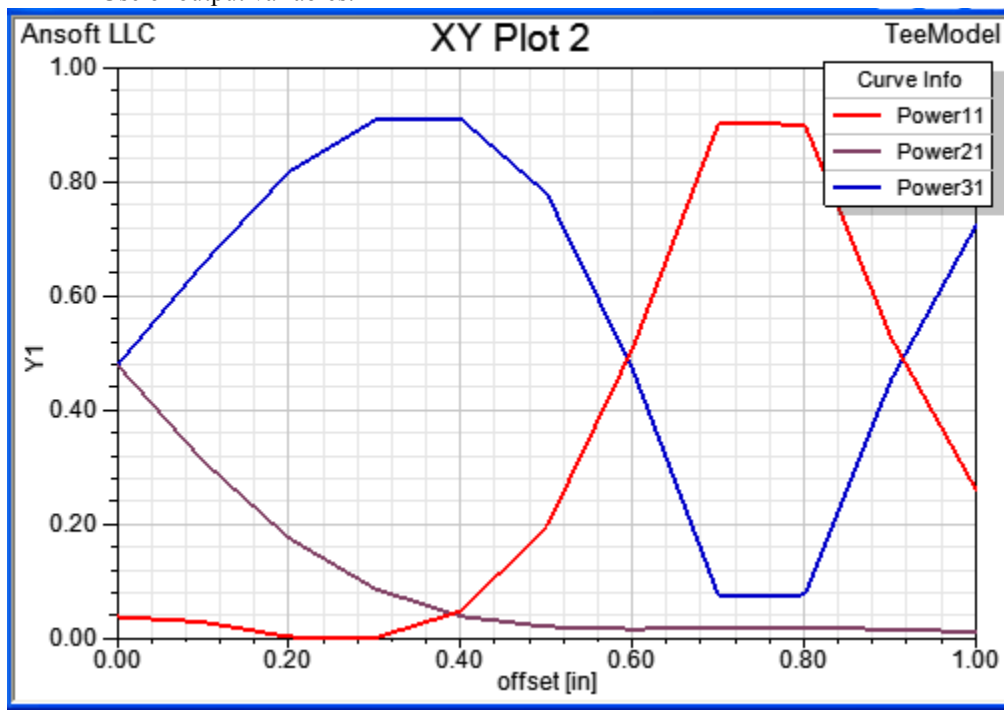


The animated *Mag_E1* plot of the E-field when the septum is located 0.2 inches closer to Port 2.

The second version of the wave guide t-junction demonstrates the use of the Optimetrics.

- parametric analysis
- variable for optimization
- an optimization setup
- a cost function
- Optimization analysis.
- plot of cost values versus solved iterations.

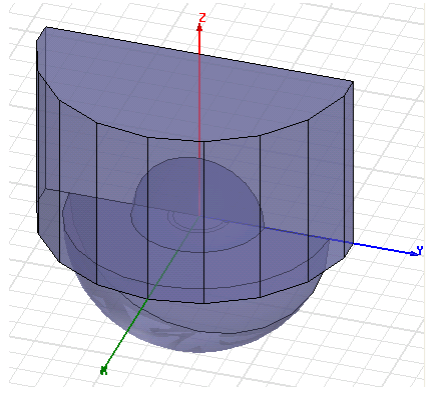
- Use of output variables.



See Getting Started Guides.

Dielectric Resonator Antenna

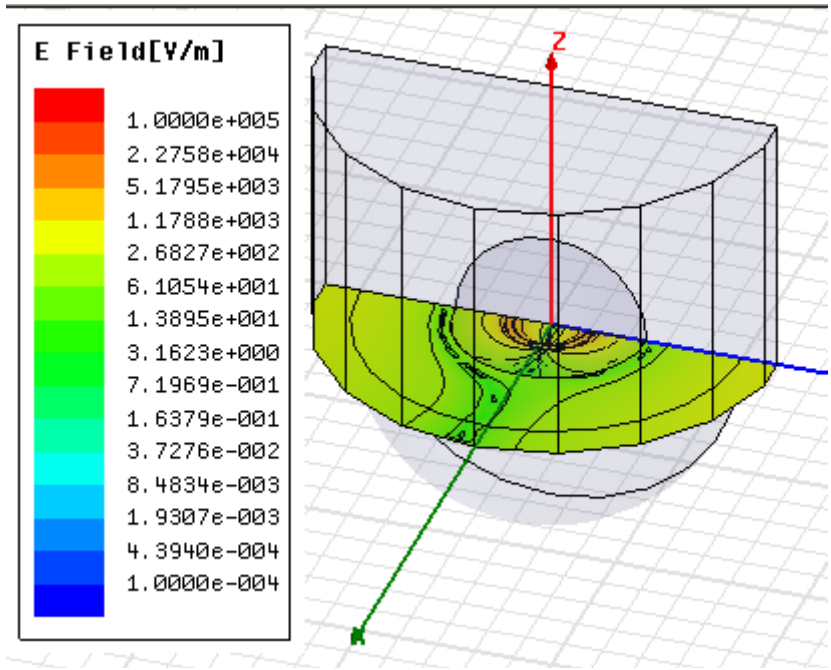
The dra_diel directory in the Example/Projects folder contains the modal solution project described in *Getting Started With HFSS: A Dielectric Resonator Antenna*. See Getting Started Guides.



This design demonstrates the use of:

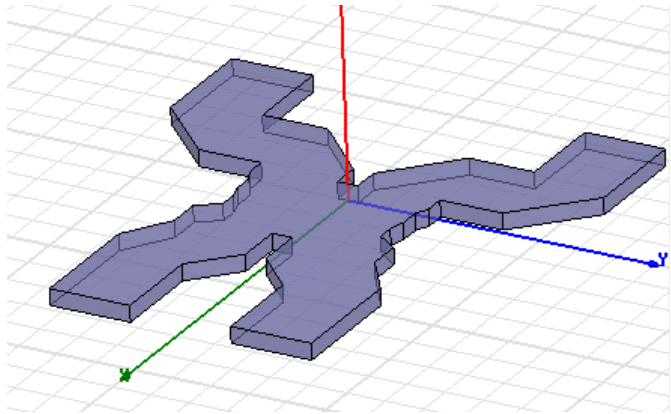
- boolean operations on geometries,

- the use of symmetry and radiation boundaries
- mesh operations
- lumped ports
- modifying the impedance multiplier because of symmetry
- animation of a field plot
- setting up an infinite sphere and computing antenna parameters



Waveguide Combiner Example Project

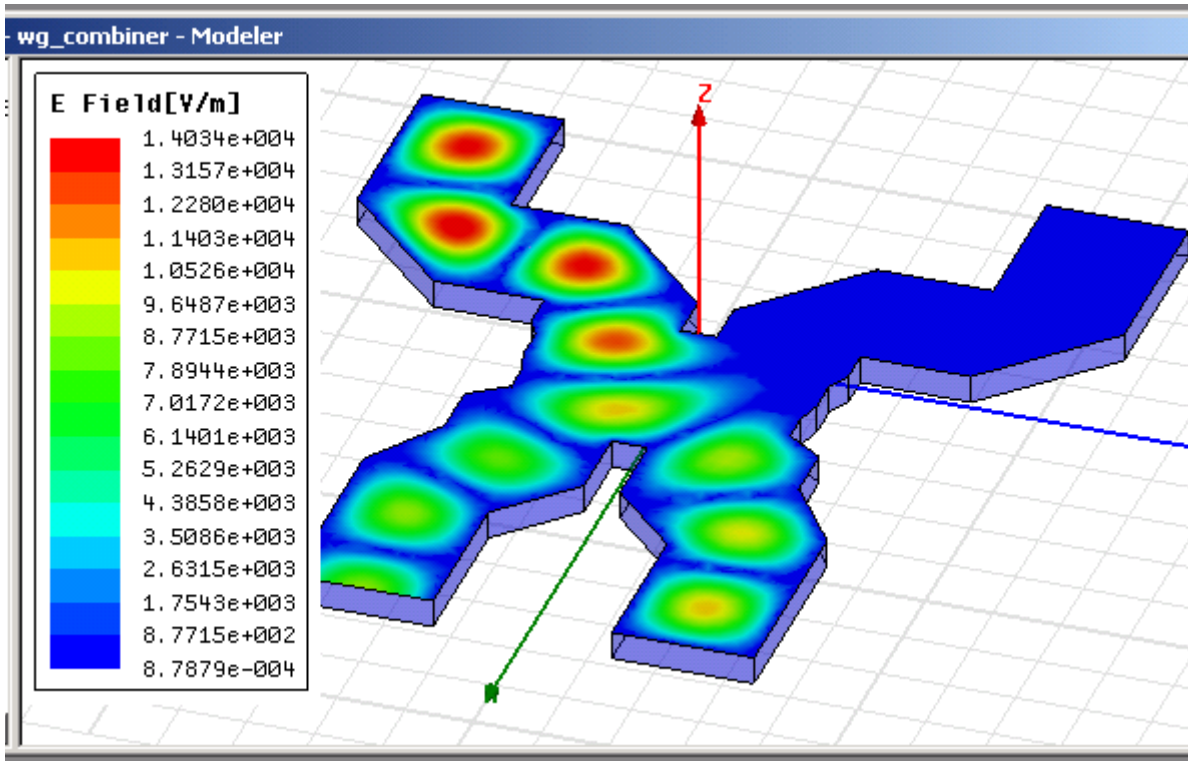
The wg_combiner project is located in the Example/Project folder. This project has an associated Getting Started Guide. See Getting Started Guides.



The waveguide combiner project demonstrates:

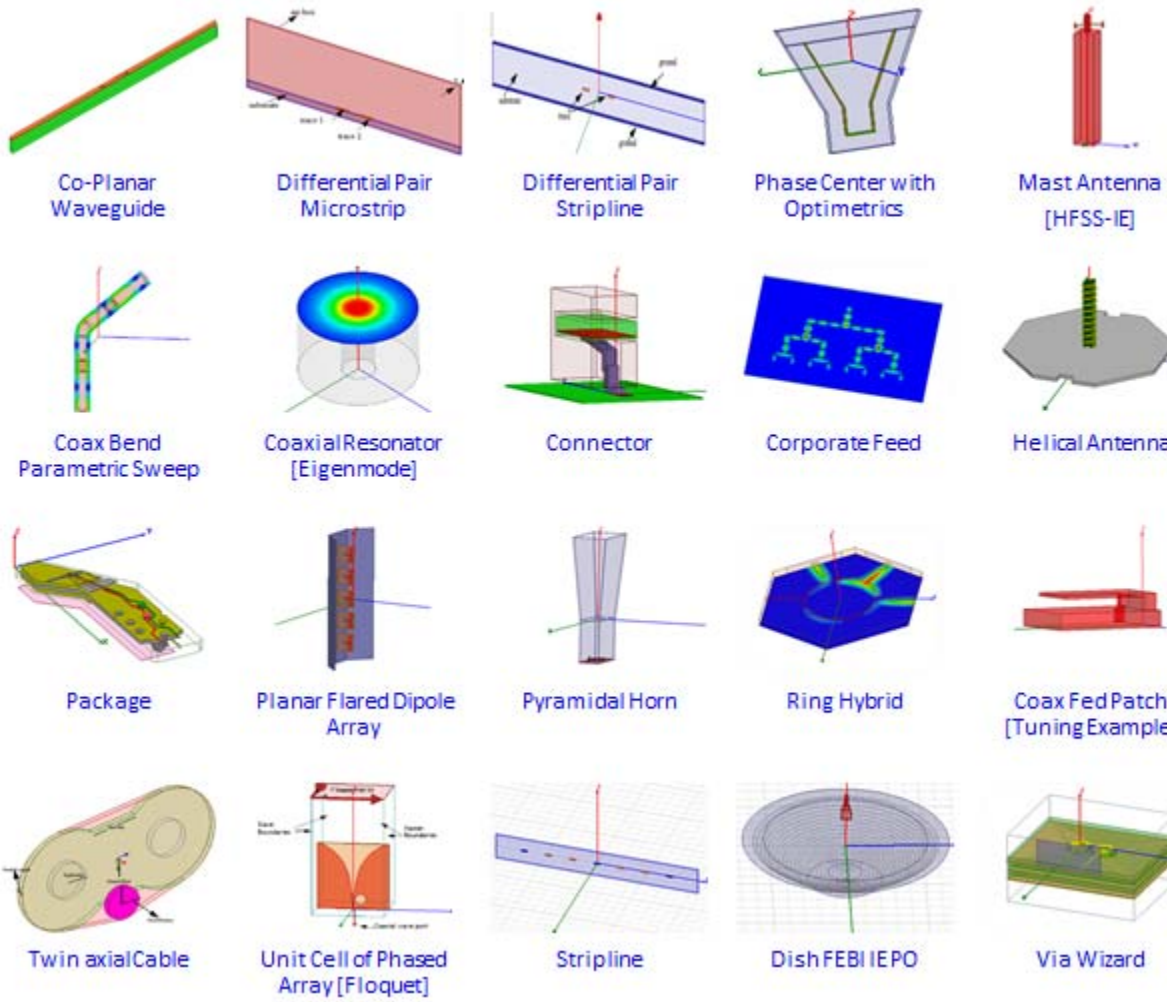
- finite conductivity boundary condition
- symmetry boundary condition
- wave ports
- integration lines in wave ports.
- solution data
- plot creation and analysis

- a phase animation.



Other Examples with Brief Descriptions in the Online Help

The following example projects have brief descriptions in the online help.



For further examples, see Getting Started Guides and look at the ANSYS Website

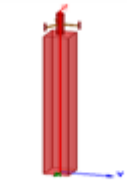
Related Topics

Example Projects

Getting Started Guides

Antennas on Mast

Description - Three dipole antennas mounted on an aluminum mast, simulated in HFSS-IE. The mast is mounted on an infinite ground plane.



Model - The mast is 3.1 m high. The dipoles are modeled using 2D objects with a PerfE boundary. The excitations are lumped ports and the mounting structures are modeled by the polystyrene rods. Note under Boundaries in the Project tree, InfGndPlane1.

Setup - Adapt at 0.9 GHz.

Note To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed in the Properties window. Selecting an object in the History tree will also display its properties.

Post Processing Antennas Mounted on a Mast

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

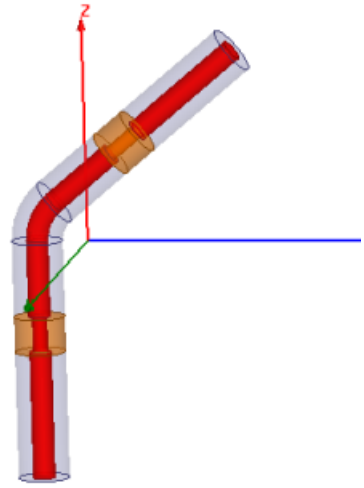
Before viewing the fields make sure all three antennas are excited. Go to **HFSS-IE>Fields>Edit Sources** and uncheck all Terminated options. Set Scaling Factor to 1 for all three ports.

To view the radiation pattern shown, double click on Radiation Pattern 1 under Results in the Project tree. This is the $\theta = 90^{o}$ pattern cut.

To view the induced currents on the mast, double click on Mag_J1 under **Field Overlays>J Fields**.

Parametric Sweep of a Coax Bend

Description - This is a model of an air-filled coax bend. Two teflon supports with inner radius compensation are included in the model. The variable here is bend_angle and it controls the angle of the top half of the bend.



Model - The walls of the coax use the default outer PerfE boundary. The inner conductor is copper. The ports are assigned to the faces of the coax so port2 will automatically move with the coax body as the angle is changed.

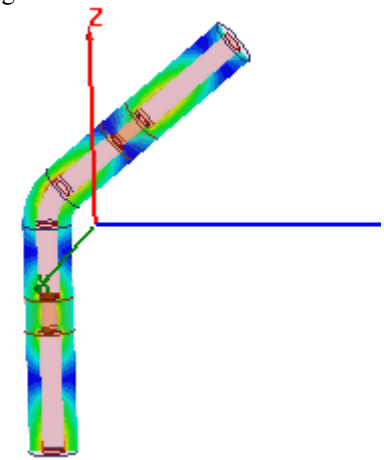
Setup - Adapt at 5 GHz and do an interpolating sweep from 0.1 to 5 GHz. The defined Parametric sweep in bend angle sweeps from 50 - 90^{o} every 10^{o}.

Note To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed in the Properties window. Selecting an object in the History tree will also display its properties.

Post Processing Coax Bend

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and

Mesh Statistics. To view the solution data for different bend angle values, click on the ellipsis [...] net to the design variation box and select the variation of interest.



To view the S Parameter frequency plots for all five bend_angle variations, double-click on **XY Plot 1** under Results in the Project tree.

To see the shade plots of the electric field, double-click on Mag_E1 under **Field Overlays>E Field**. To view all variations right-click on Mag_E1 and select **Animate**. In the **Select Animation** window select New, and then select bend_angle for the swept variable in the setup pane. Then click OK.

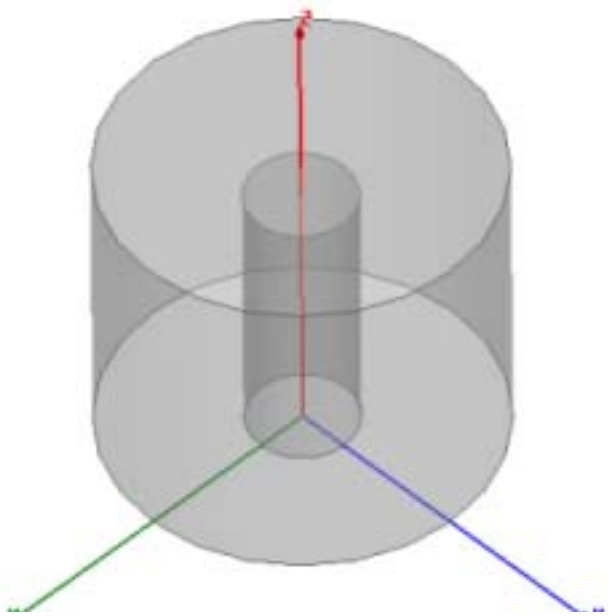
Coaxial Resonator

Description - A coaxial resonator model showing how to use the Eigenmode solver. The eigen solver computes the resonant frequency and Q of the model. This example was taken from *Microwave Circuit Modeling Using Electromagnetic Field Simulation* (D. Swanson Jr., W. Hoefer).

Model - A coaxial cavity. Walls are defined to have $\sigma = 6.17 \times 10^7$ mho/m.

Setup - There are no defined sources in an eigen solution so you need only select the number of modes to compute and the convergence criteria. For this model, only the first mode is computed. For maximum accuracy, we need to use curvilinear elements. To verify that this has been set for the

model, go to **HFSS>Mesh Operations> Initial Mesh Settings**, and make sure "Apply Curvilinear Elements" is checked.



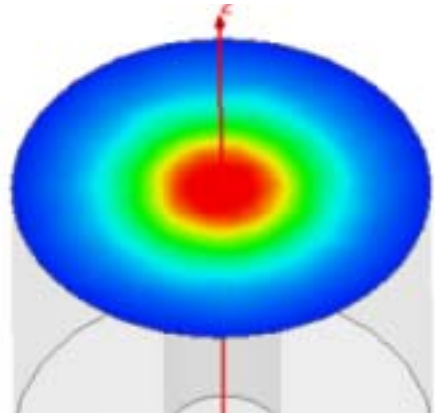
Note Selecting an object in the History tree will also display its properties.

Coaxial Resonator Post Processing

After solving, you can view solution data by right-clicking on Setup1 and selecting **Eigenmode Data** to display the **Solution** dialog. You also view the **Solution** tabs for **Profile**, **Convergence**, and **Mesh Statistics**.

To view the resonant frequency and Q, select the **Eigenmode Data** tab on the Solution dialog.*

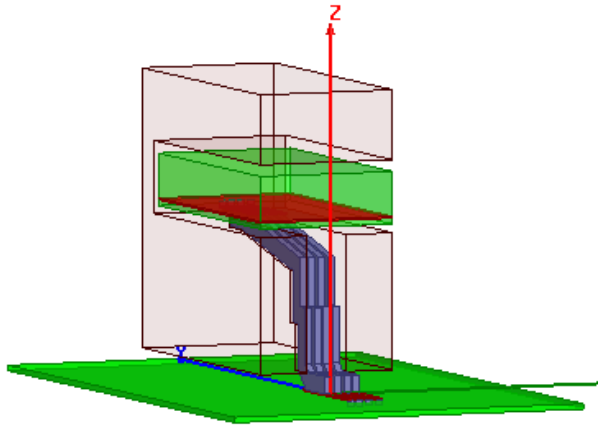
To view the shade plot, right-click on E Field under Field Overlays in the **Project tree**, and select **Update Plots**.



* Data computed using a mode matching program are given in the reference. The results presented are f_0 = 1.87 GHz and Q = 5592.

Connector - Terminal Example

Description - a simplified model of a four pin section of a connector. This is a driven terminal design.



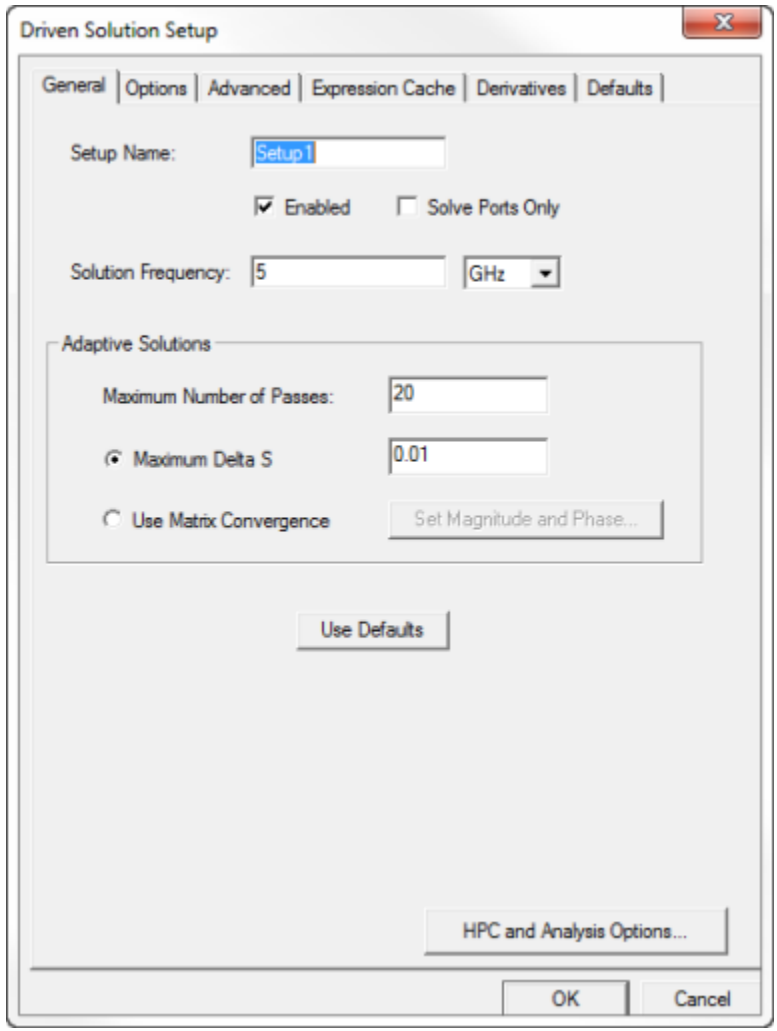
Model - the connector is configured with lumped ports on each end of the two inner pins. The two outer pins are each grounded at both ends. The boards are FR4 and the connector body is modified epoxy. A radiation boundary is applied to the surrounding airbox.

Setup - Driven Terminal Solution with adapt at 5 GHz. An interpolating sweep is also included that has an upper frequency of 5 GHz and uses DC extrapolation at the low end.

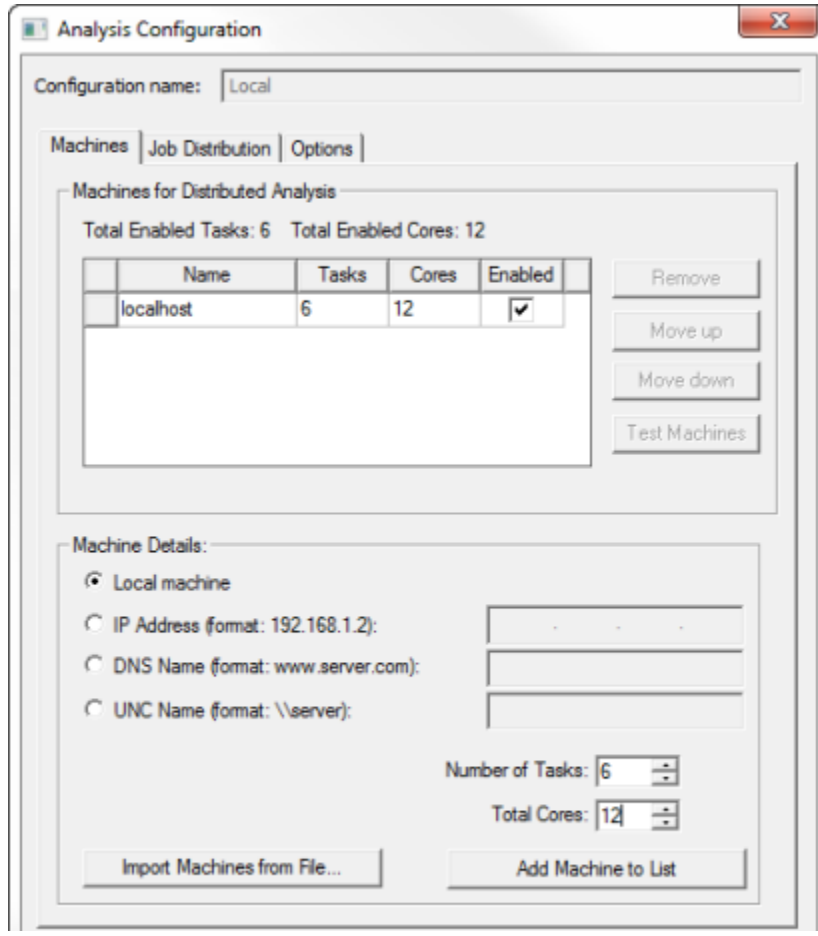
Note To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed in the Properties window. Selecting an object in the History tree will also display its properties.

HPC Analysis Setup

You can set an adapt frequency at 5 GHz with an interpolating frequency sweep from 0 to 5 GHz. Since several frequencies are being solved in this design, you can set up an HPC Analysis to distribute the frequencies resulting in efficient simulation.



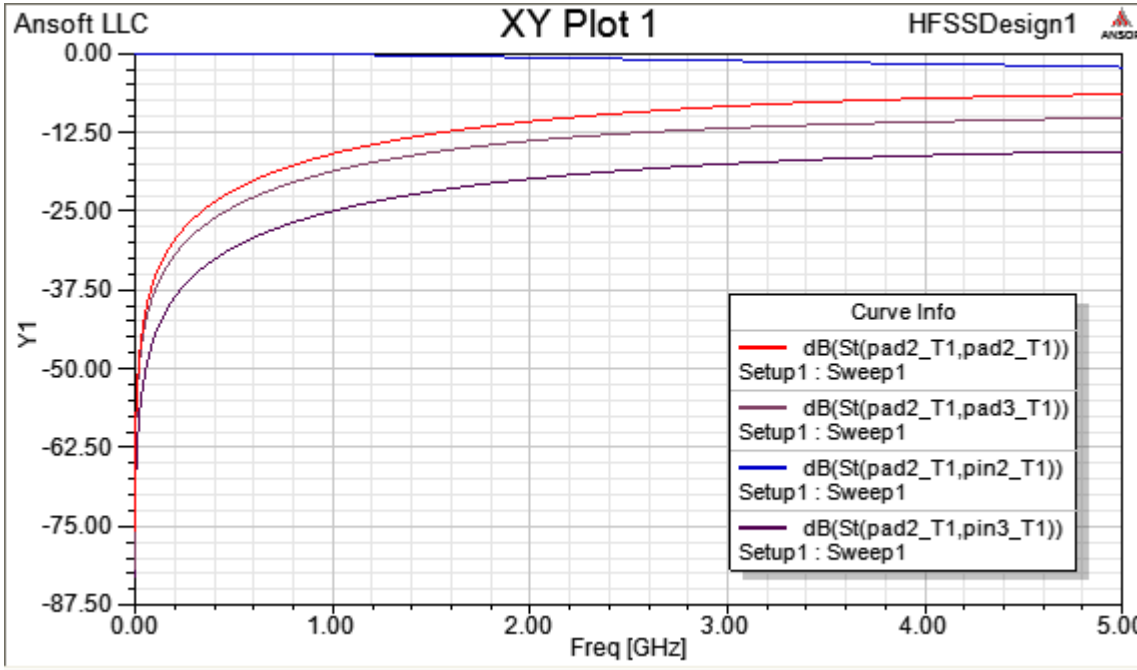
During the simulation, adaptive mesh refinement uses the **Total Cores** that are configured in the HPC setup, while the number of cores used to solve each frequency point is determined by **Total Cores**/**Number of Tasks** configured in the **Analysis Configuration**. On the **Driven Solution Setup** dialog box, if you click the **HPC and Analysis Options** button and click **Add**, you can set the **Number of Tasks** and the **TotalCores** on the **Analysis Configuration** window.



Connector Post Processing

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

To view the S parameter plot show below, double click on XY plot1 in the Project Tree under Results.



Co-Planar Waveguide (Driven Terminal)

The coplanar waveguide CPW consists of a signal trace sandwiched between two coplanar ground conductors. The width of the signal trace and the gap between the trace and the ground conductors affect the characteristic impedance. Model a short length as shown below and to obtain a longer length of the model you can deembed out of the port.

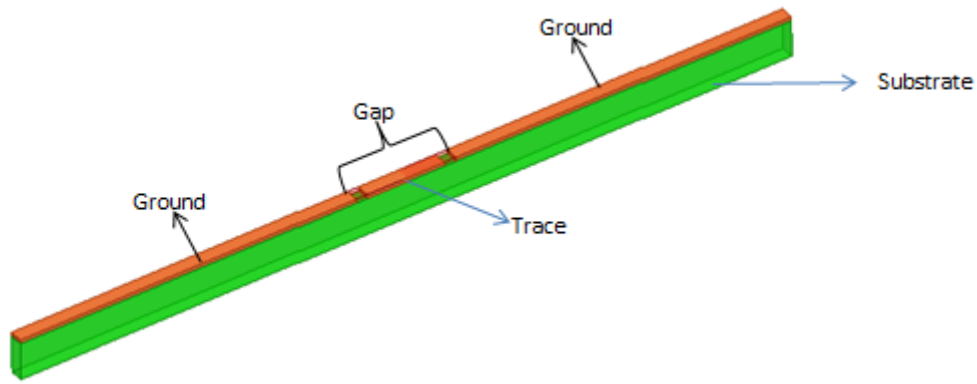


Figure 1 CPW (air box + ports hidden)

Define the ports such that only their faces touch the air box. The edges of the ports should not touch the edges of the air box.

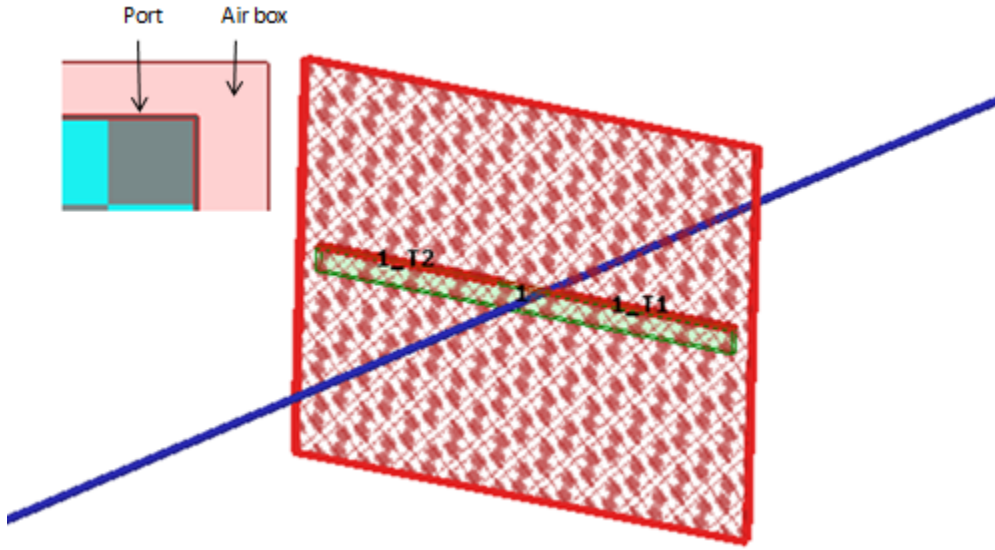


Figure 2 ports with dembeddding

Define the Radiation boundary only along the thickness of the air box. Assigning a radiation boundary on all surfaces of the air box in this model can make the port boundary to be conducting.

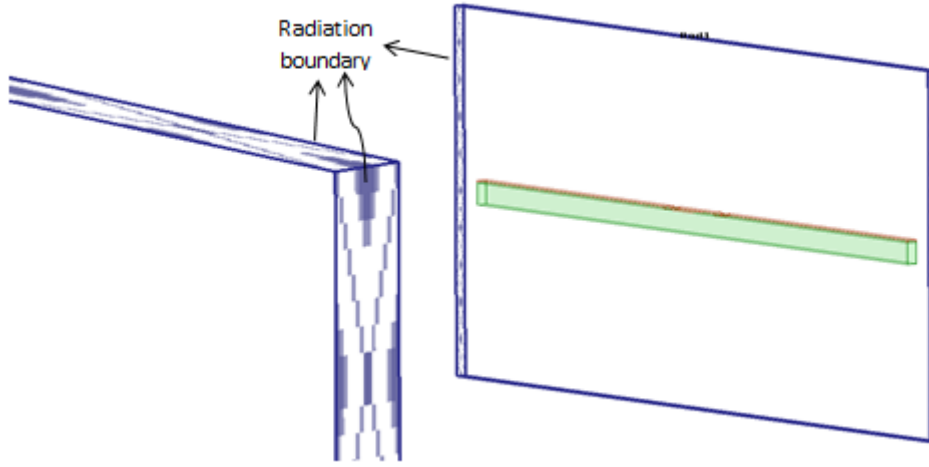


Figure 3 Radiation Boundary

Define a perfect H boundary on the air box. The wave port touches a perfect H boundary and therefore becomes an open circuit.

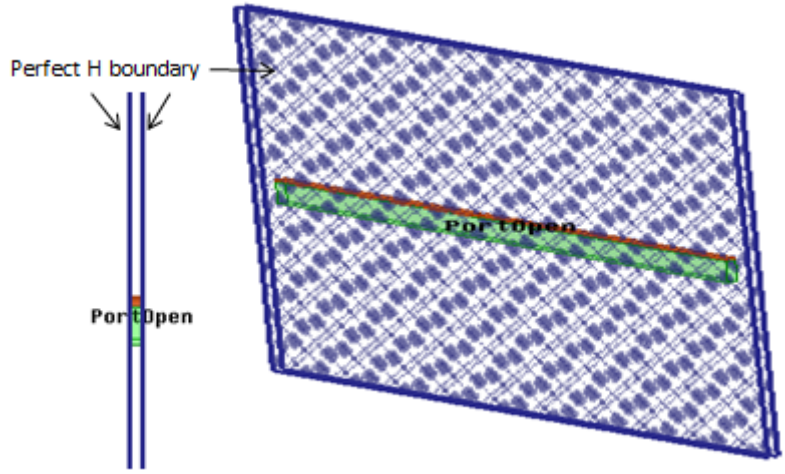


Figure 4 Perfect H boundary

So, the port boundary does not stay as a conductor anymore and almost mimics a "perfect open." This is because with the application of the perfect H boundary "behind" the wave port, the port boundary becomes an open and will no longer be one of the conductors associated with the port. Now with three conductors namely, the two grounds and the trace, there are two possible modes that this structure can carry. Obviously, for the CPW structure we are interested in the center conductor excited at a voltage with reference to the two sides (or what we arbitrarily call "ground") conductors at zero potential. Since voltage values can be arbitrary, this same mode could also be considered as the center conductor at "0" volt with the two side conductors at some equal voltage offset from the center conductor. In the terminal framework such a mode can be described as the center conductor labeled "reference" conductor with the two outside conductors considered to be the "terminals." Then, by placing those two conductors at equal potential with respect to the center conductor they can be defined as differential pair whose common mode is the aforementioned mode of interest.

The Edit Post Process Sources dialog is shown below.

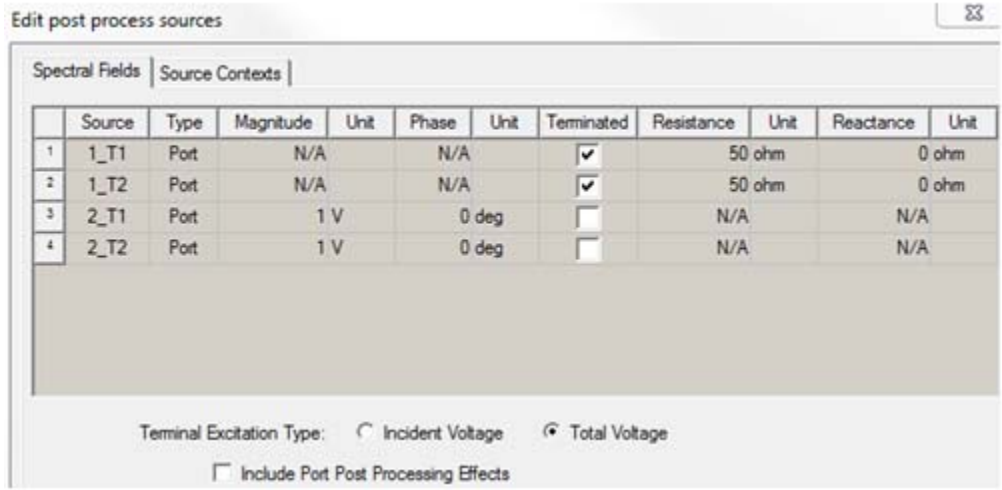


Figure 5 Edit Port Process Dialog

Notice from the plots below how the field gets trapped in the signal trace and dielectric.

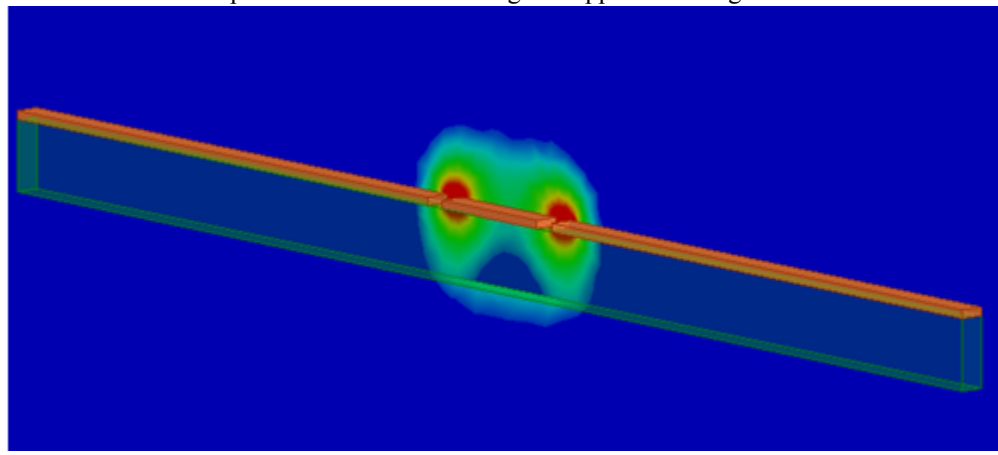


Figure 6 E field Plot

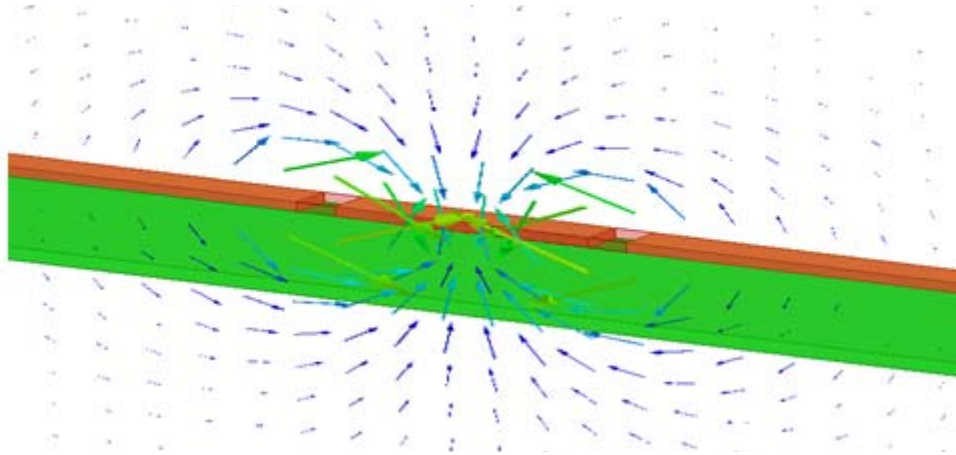


Figure 7 : Vector Field

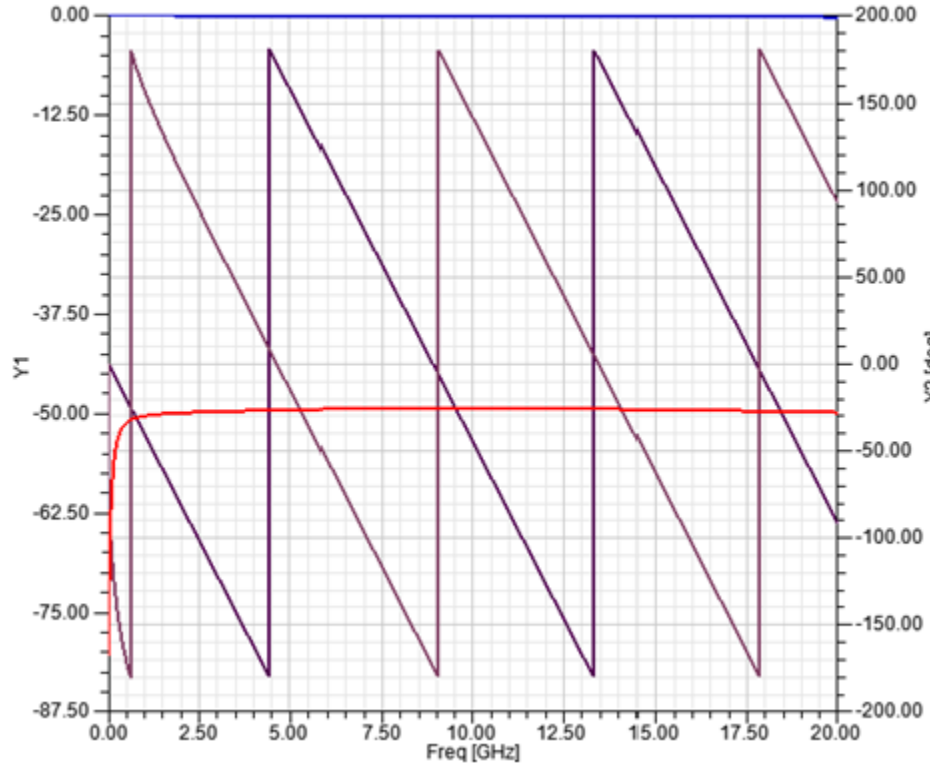


Figure 8 S-Parameter plot for the differential pair (legend shown below)

Curve Info	Y Axis
dB(St(D1,D1)) 5GHz : DC_20GHz	Y1
dB(St(D2,D1)) 5GHz : DC_20GHz	Y1
ang_deg(St(D1,D1)) 5GHz : DC_20GHz	Y2
ang_deg(St(D2,D1)) 5GHz : DC_20GHz	Y2

Related Topics

Coplanar Waveguide with Ground

Coplanar Waveguide with Ground (Driven Terminal)

The coplanar waveguide (CPW) with ground plane is shown below.

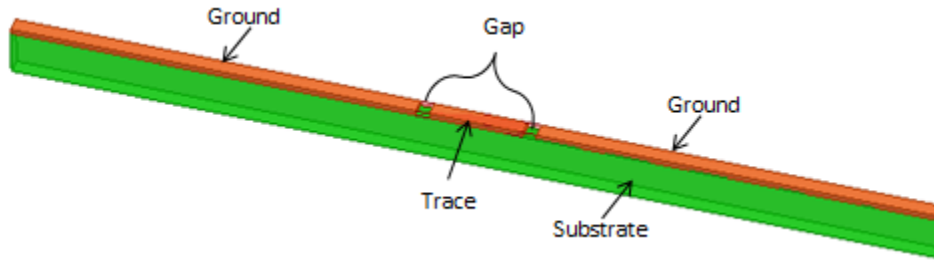


Figure 1: Coplanar Waveguide with Ground

When using HFSS to model a transmission line it is not necessary to model the entire length explicitly. Instead create a short section and to generate the desired length for the model you only need to de-embed the appropriate distance from the wave port. For this model the explicitly solved length was chosen as a dimension typical of the cross section such as spacing between conductors or a conductor thickness. This is done to keep the 3D mesh as close to a high quality as possible. So for a model of explicit length X and a desired simulation length Y one can de-embed from each port (Y-X)/2.

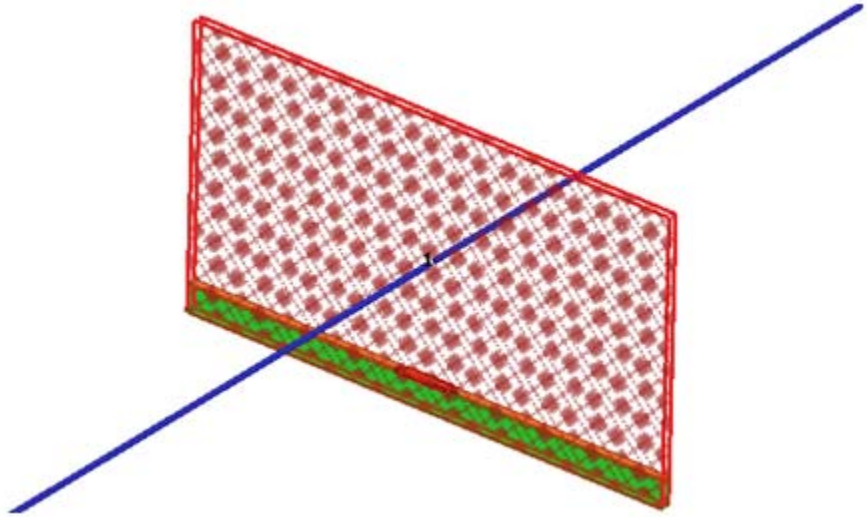


Figure 2: Ports with Deembedding

Deembedding greatly reduces the simulation effort.

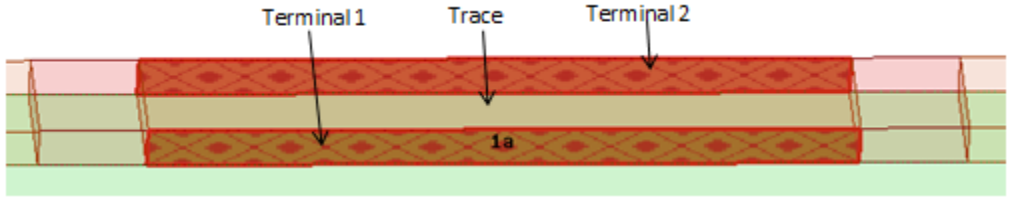


Figure 3: Terminals

The resulting E field and vector fields are shown below.

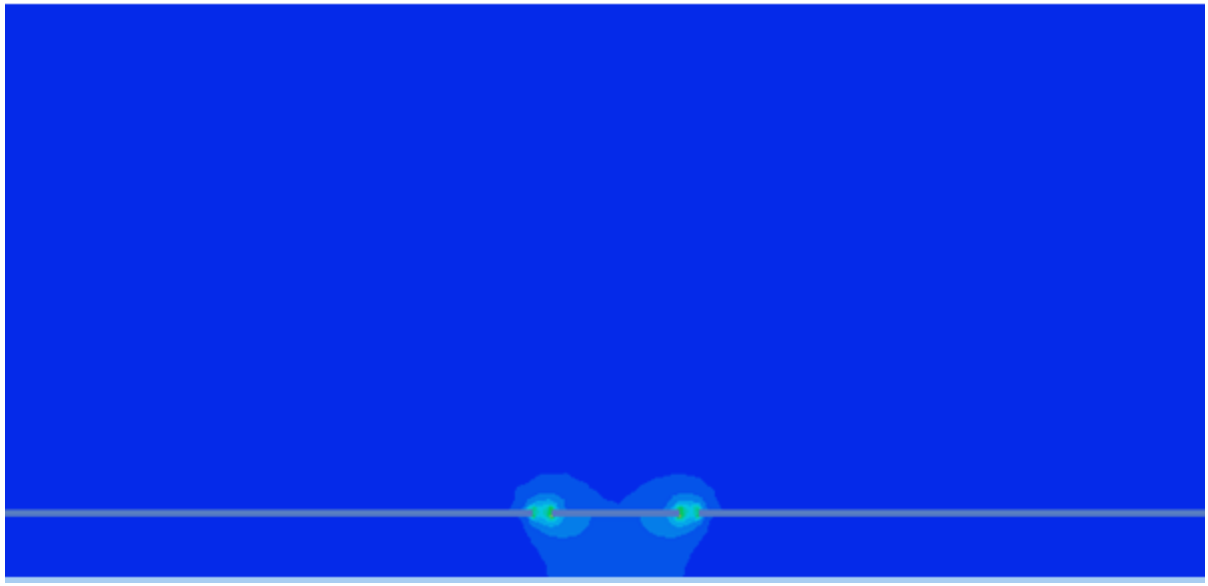


Figure 4: E Field

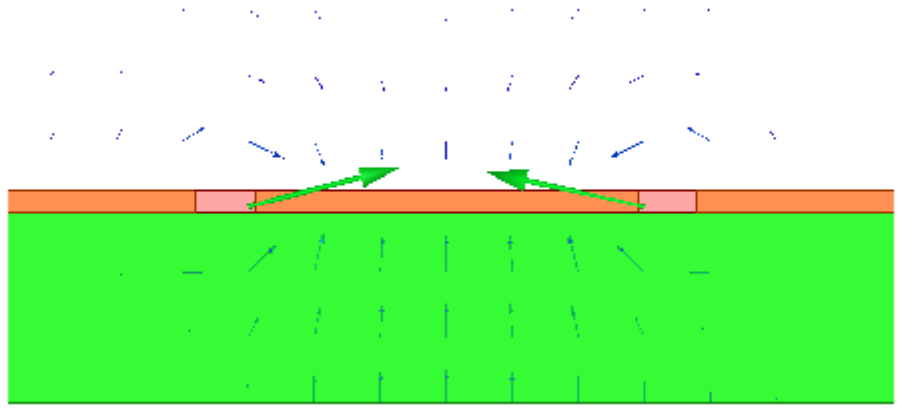


Figure 5: Vector Field

The S-parameter plot is shown below.

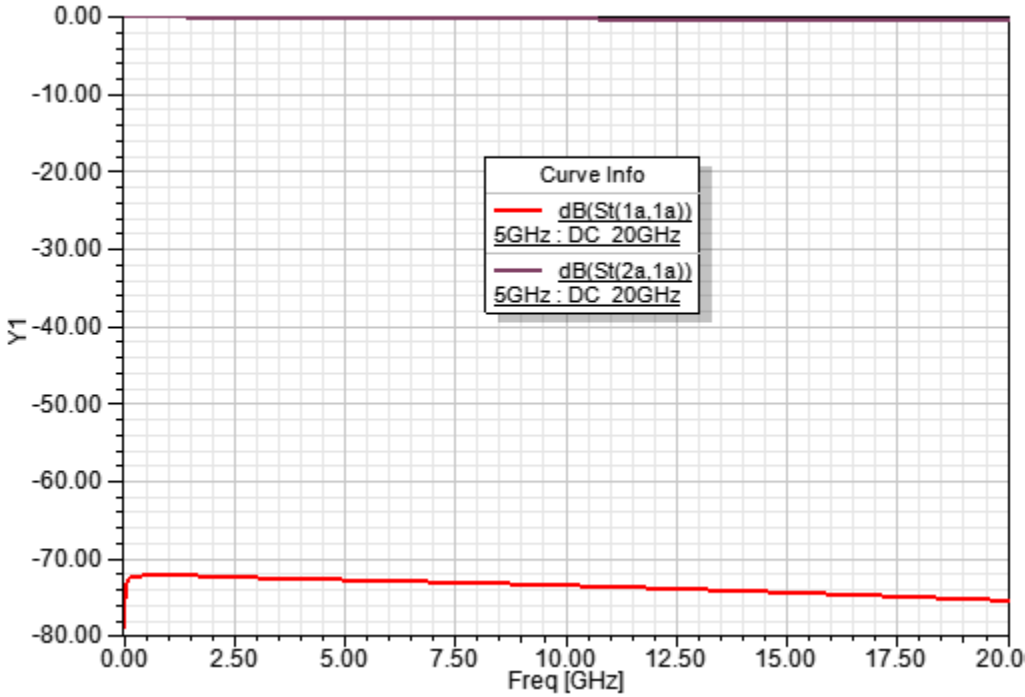
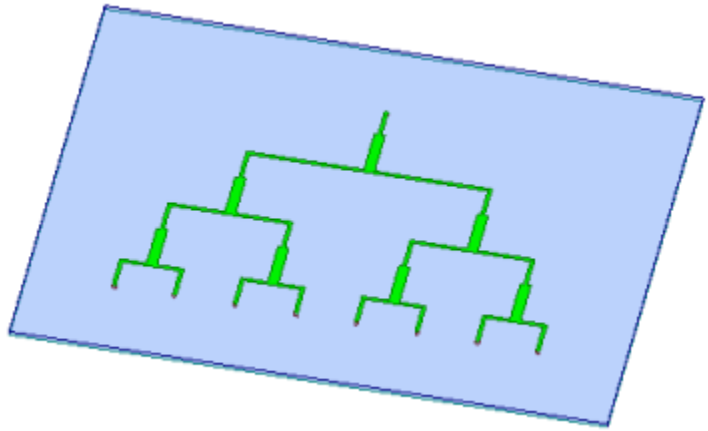


Figure 6: S-parameter Plot

Corporate Feed

Description - A microstrip 8 way corporate feed. The model was originally created in Ansoft Designer. Export to HFSS was used to create this project. It is a Driven Terminal design.



Model - The traces are defined as 2D objects with a finite conductivity boundary of copper. The sources are all lumped ports with a 50 Ohm impedance. There is a surrounding air box with a radiation boundary on the top and sides.

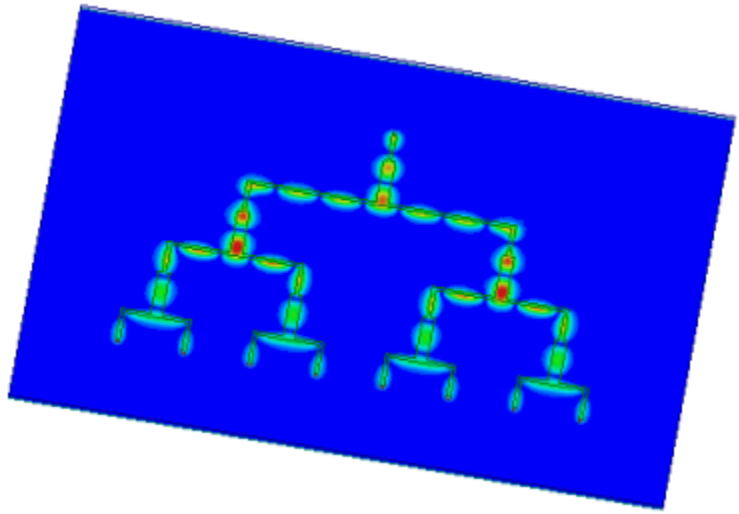
Setup - Adapt at 7.5 GHz and use Mixed Order for the Order of Basis function.

Note To view a port or boundary, select the desired item in the Project Tree. It is highlighted in the Model window and the properties will be displayed. Selecting an object in the History tree will also display its properties.

Corporate Feed Post Processing

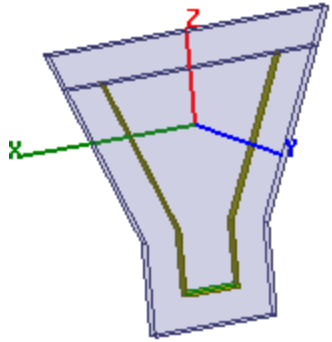
After solving, you can view solution data by right-clicking on the Setup and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

To view the shade plot of the E field, shown below, double click on MagE1 in the Project tree under **Field Overlays>E Field**. To see a phase animation of this field plot, right click on MagE1, and select **Animation**, Then click OK in the dialog.



Determining Phase Center Using Optimetrics

The phase center is the point from which the electromagnetic radiation spreads spherically outward, with the phase of the signal being equal at any point on the sphere. This example illustrates a technique for using an optimization setup to search for the phase center.



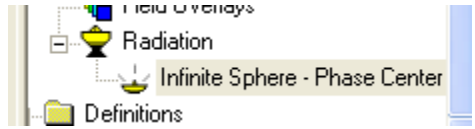
The technique involves the following tasks.

1. Define a relative coordinate system that uses a post processing variable.

   This allows the variable value to change the definition of the coordinate system and thereby change a solved solution without invalidating it.

2. Use a relative coordinate system in a Far field setup calculation.

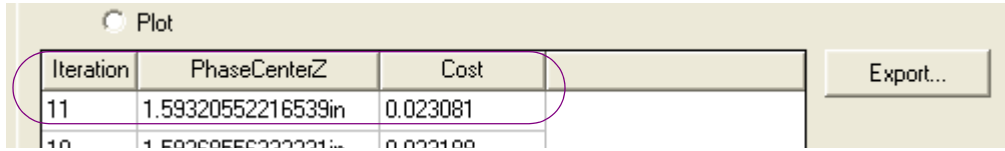
This allows far field quantities to be re-calculated when the relative coordinate system is repositioned (demonstrated in the example by changing the value of the PhaseCenterZ variable), without re-simulation.



The far field setup will also limit the angular range of the sweeping intrinsic (phi or theta).

3. Create an optimization setup to search for the phase center. The search will be efficient if you have a rough idea on the location of the phase center. The expression to be optimized is the peak to peak continuous angle of the quantity rEPhi. (The electric field of this antenna is Phi polarized.)
4. Analyze the Optimization setup and view results via the **View Analysis Result** dialog.

   The Z coordinate of the phase center will be the value of the post processing variable when Cost is at minimum.



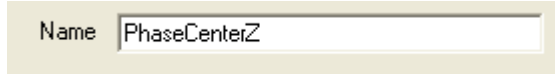
The following sections detail these steps for setting up an Optimetrics project to calculate the Phase Center.

- Define the Relative Coordinate System using a Post Processing Variable
- Use the Relative Coordinate System in a Far Field Setup Calculation
- Create an Optimization Setup to Search for the Phase Center
- Analyze and View the Optimization Results

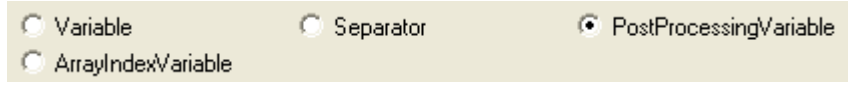
Define the Relative Coordinate System using a Post Processing Variable

First create a post processing variable that you can use to define a relative coordinate system.

1. Use **HFSS>Design Properties** to display the **Design Properties** dialog.
2. Click **Add Variable**, to display the **Properties** dialog for defining a Design Variable. The example project uses the name "PhaseCenterZ".



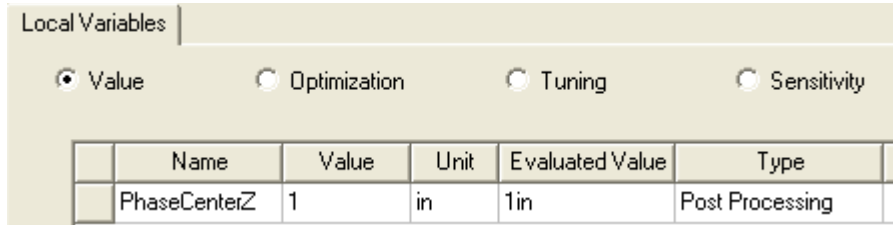
Designate the type as Post Processing variable:



Set the Unit Type as Length, the Units. This example uses "in" for the units and the value as 1.

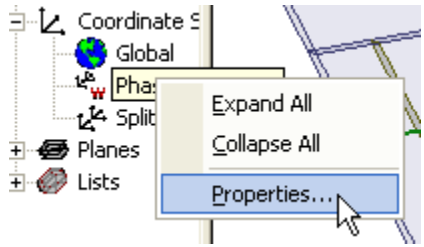
Click **OK**.

The PhaseCenterZ variable is listed as a Local variable with the Post Processing Type.

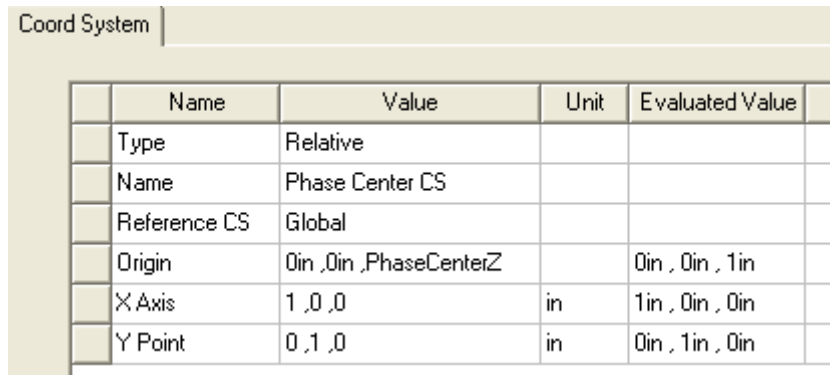


3. You can use the variable to define an offset relative coordinate system.

   Use **Modeler>Coordinate System>Create>Relative CS>Offset** to create a coordinate system, using the variable as the z coordinate. You can verify the Properties.



In the example, the PhaseCenterZ variable defines the z coordinate of the origin for Phase Center CS.



This variable allows you to change definition of the coordinate system during the optimization process without invalidating solved solution.

Next , Use the Relative Coordinate System in a Far Field Setup Calculation.

Related Topics

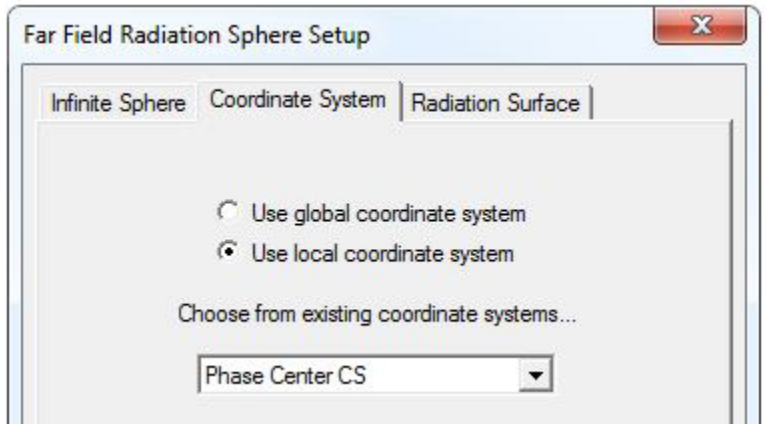
Determining Phase Center using Optimetrics

Use the Relative Coordinate System in a Far Field Setup Calculation

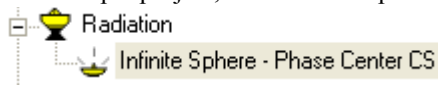
1. Click **HFSS>Radiation>Insert Far Field Setup>Infinite Sphere**, or right-click on **Radia-**

tion in the Project tree and on the shortcut menu, click **Insert Far Field Setup>Infinite Sphere**. This displays the dialog.

2. Set the Infinite Sphere values appropriately, and on the Coordinates tab select the Relative coordinate system that uses the Post Processing variable.



In the example project, the radiation sphere setup is named Infinite Sphere - Phase Center CS.



This allows far field quantities to be re-calculated when the relative coordinate system is repositioned (by changing the value of the PhaseCenterZ variable), without re-simulation.

The far field setup will also limit the angular range of the sweeping intrinsic (phi or theta). rEPhi and rETheta are the quantities of interest, depending on the radiation direction of the antenna.

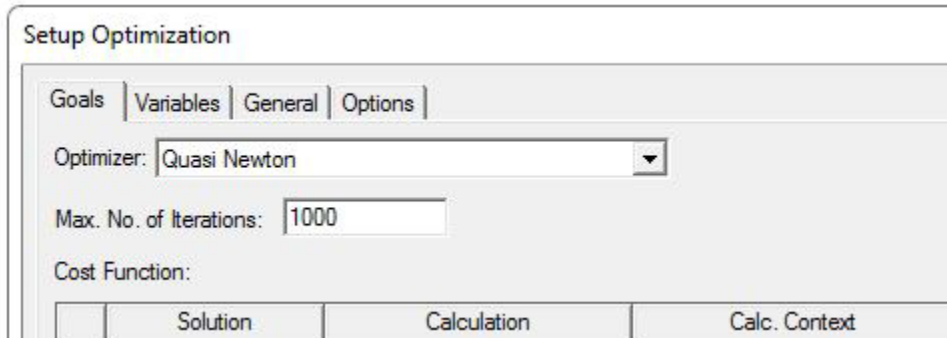
Next, Create an Optimization Setup to Search for the Phase Center

Related Topics

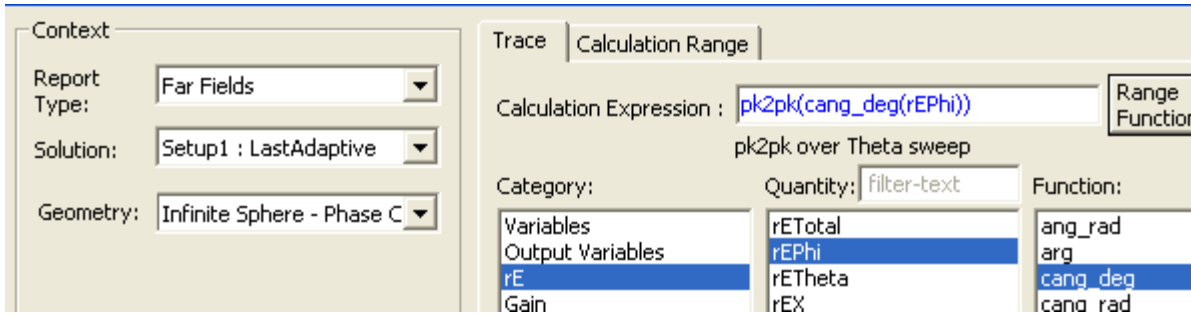
Determining Phase Center using Optimetrics

Create an Optimization Setup to Search for the Phase Center

1. Add an Optimization setup.

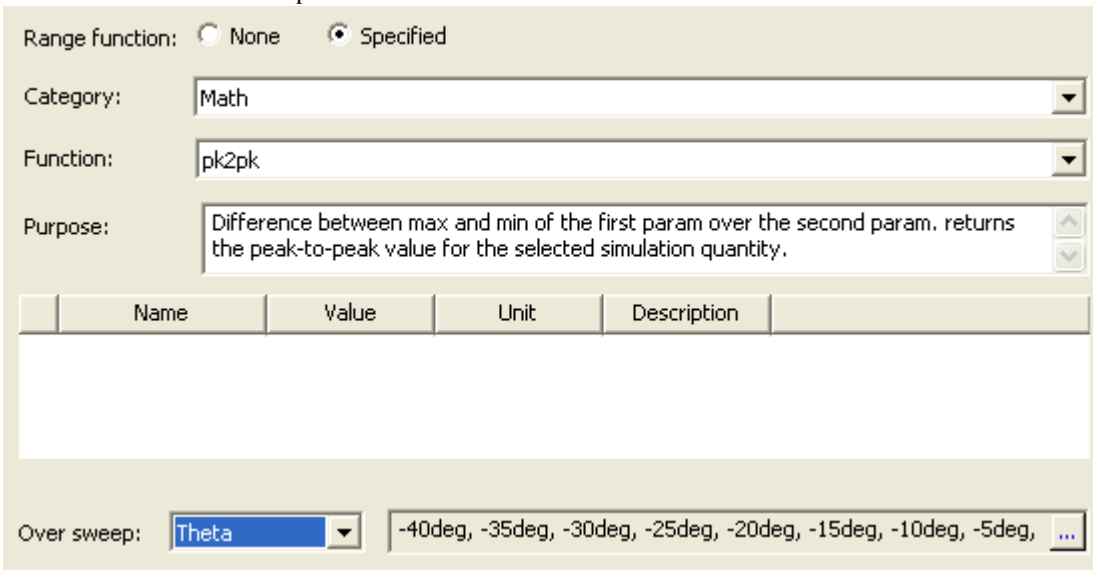


2. Click the Setup Calculations button to open the dialog for defining the Calculation Expression.
   You will define Optimization Goals designed to search for the phase center.
   The expression to be optimized is the peak to peak continuous angle of the quantity rEPhi.

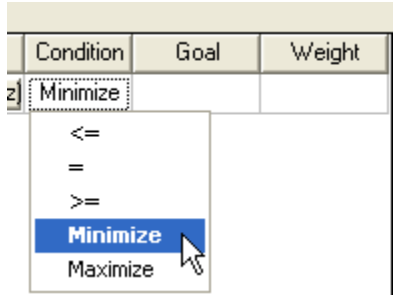


- The Report Type is Far Fields.
- The Solution is LastAdaptive.
- The Geometry is the Infinite Sphere defined using the PhaseCenterZ variable.
- The Calculation Expression includes the rEPhi Quantity with the cang_deg function. The antenna is Phi polarized for the Phi=0 plane (E-field along the y-axis).
- You then click Range Function button to select Math and pk2pk to apply over the Theta

sweep.



- The Over sweep field defines the range of the sweeping intrinsic, with Theta from -40 deg to +40 deg. 0 deg is selected for the value of Phi.
- Return to the Setup Optimization dialog. Here you may need to scroll to the right to see and set the Goal weight Condition to Minimize.



The "Minimize" goal condition doesn't require user to setup the goal value/weight and can be more natural for antenna related application. However, users need to be aware of below restrictions.

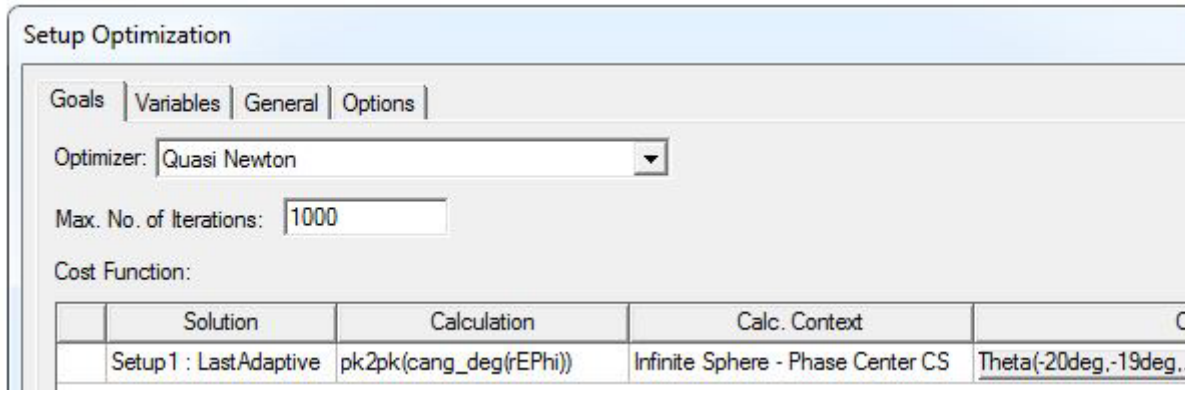
- "Minimize/Maximize" can only be used when a single calculation quantity is being optimized.
- Optimetrics only supports one dimension of range reduction, that is, no nested range function

The above restrictions implies "Minimize/Maximize" choices can only be used if

- Only one sweep in the Calculation Range has multiple values, and
- A Range reduction function is used in the Calculation Expression

In the example project, OptimizationSetup2 uses the Minimize selection. For comparison, The OptimizationSetup1 uses the <= Condition with a Goal of 0 and a weight of 1, subject to the Noise setting of 0.0001. The final values for PhaseCenterZ in the two setups are very close.

The completed Goals for the Optimization setup2 for the example display as shown:



For the **Variables** tab, you should check to include the variable used to define the relative coordinate system. The search is restricted to a minimum of 5 in and a maximum of 10 in along the Z coordinates of the relative coordinate system.

For the **General** tab, uncheck Update design parameters value after optimization. This permits you to view a plot of the progress of the simulation.

OK the completed Optimization Setup.

Next, you Analyze and View the Optimization Results.

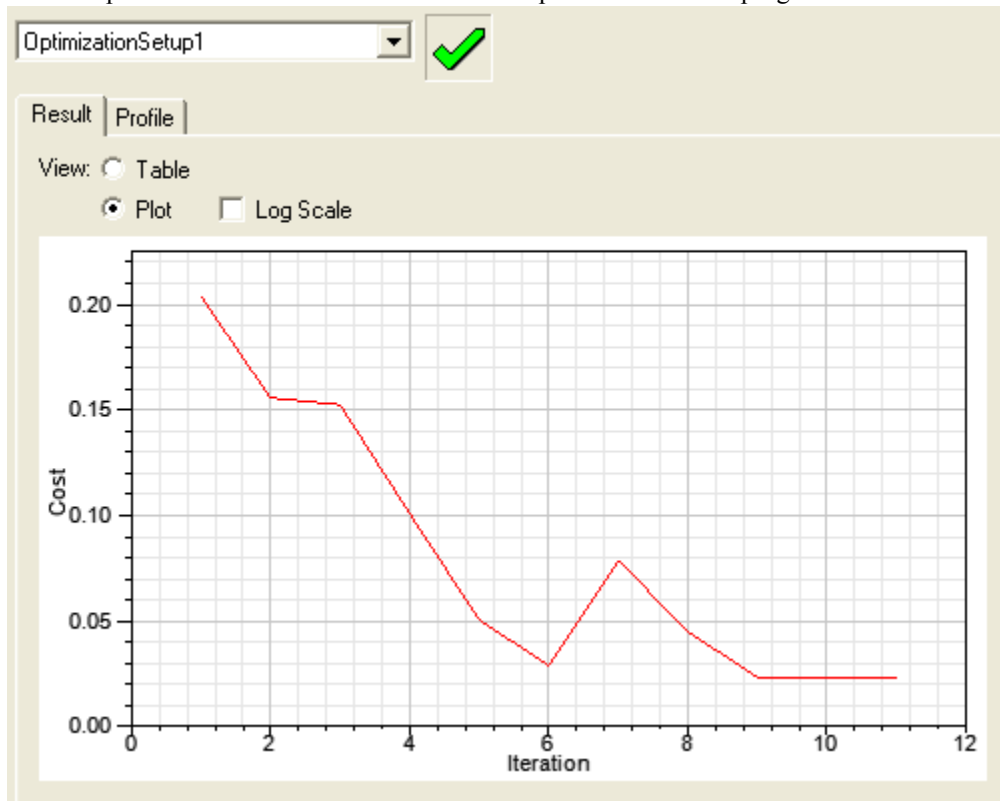
Related Topics

Determining Phase Center using Optimetrics

Analyze and View the Optimization Results

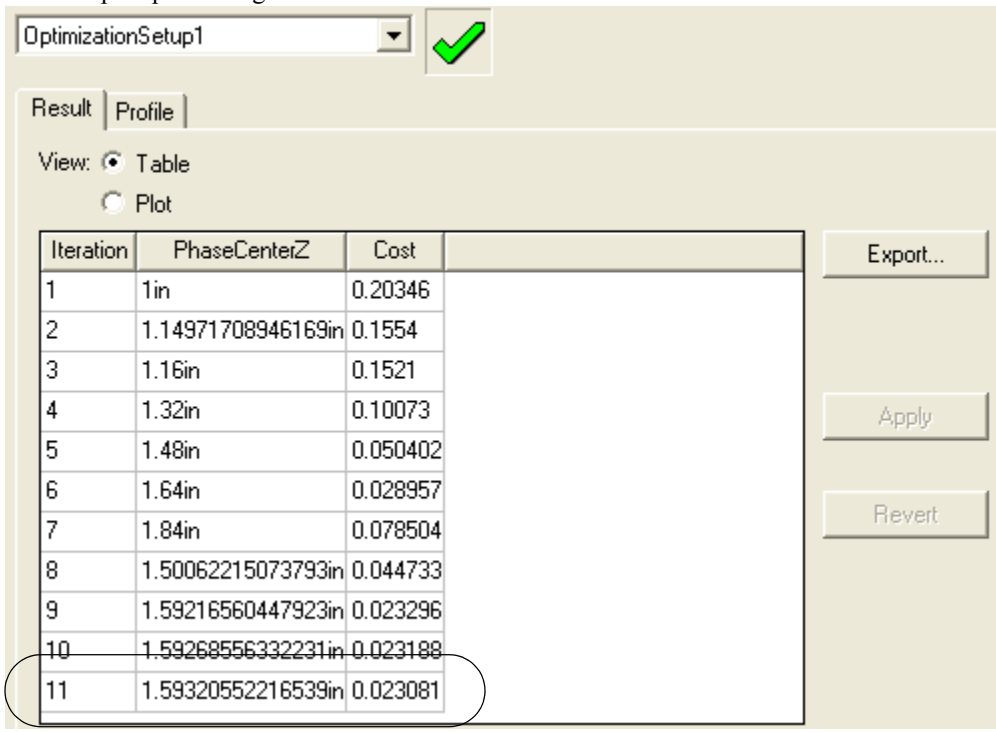
Right-click on the Optimization Setup to view the short-cut menu and click **Analyze** to start the simulation. Then right-click again, and select View Analysis Result to view the Post Analysis dialog. Select the Plot radio button to watch the progress of the simulation.

The plot form of the results shows how the Optimization search progresses towards the lowest cost.



The table form of the Results shows that the Z coordinate of the phase center is the value of the

post processing variable when Cost is at minimum.



Related Topics

Determining Phase Center using Optimetrics

Dish FEBI IE PO

This project shows the same simulation of the dish antenna by using the variations in the HFSS solver over designs with different levels of approximations in the geometries. First we will describe the HFSS designs followed by the HFSS-IE designs.

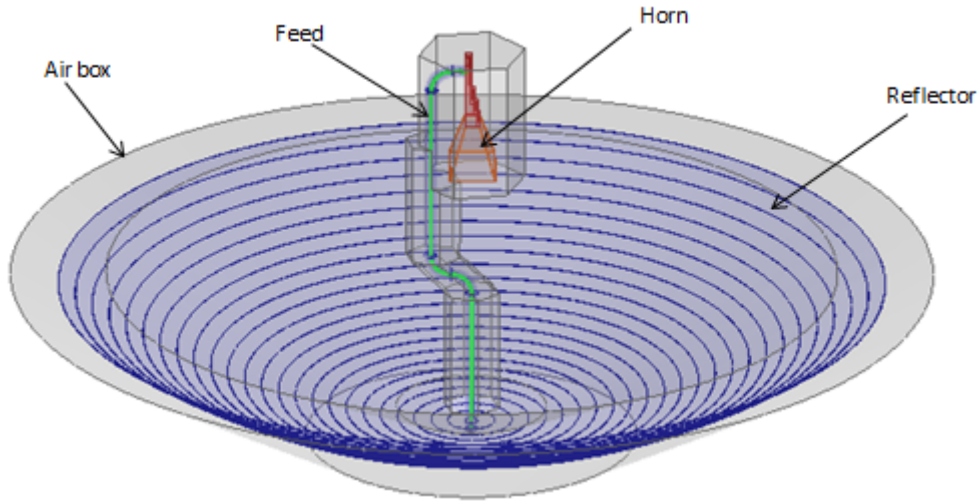


Figure 1: Dish FEBI with Feed Network

In this set-up a FEBI radiation boundary is applied on the air box that encloses the entire structure. This radiation boundary uses integral equations in its solution. See the settings in the Radiation Panel below. The Model exterior option is selected which uses FEBI method in the solution.

Figure 2: FEBI Boundary and Radiation Panel

The air box is close to the structure; in the initial mesh settings the curve linear elements are turned off. Typically, this design requires more RAM and CPU for its simulation than the other designs in this project, and the simulation produces accurate results and takes longer.

Dish FEBI with horn only using IE Regions [Driven Modal]

In this design there is no feed network and we are using the FEBI boundary on the horn only while the dish is in the IE region.

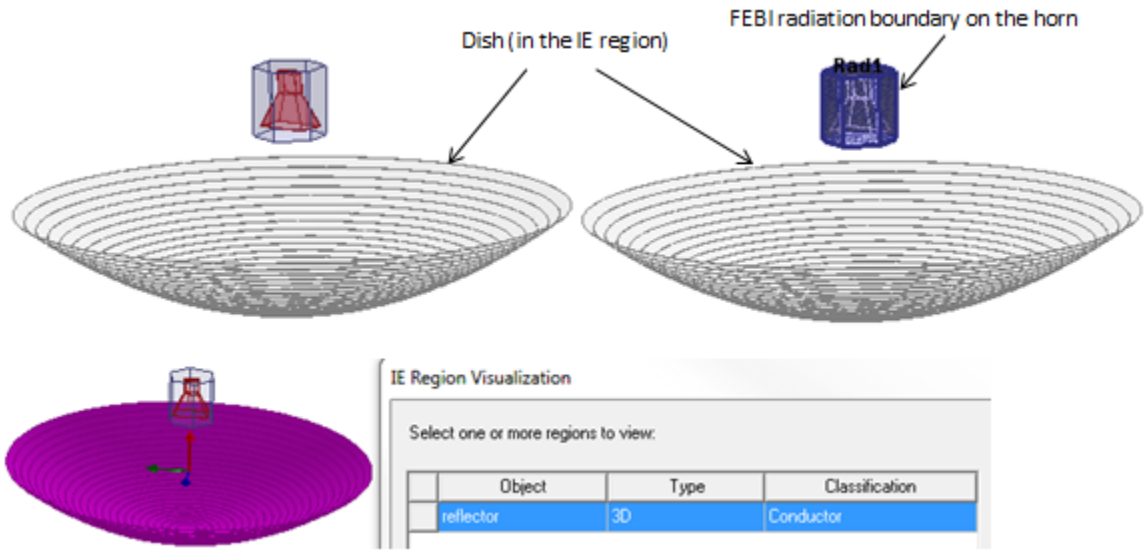


Figure 3 Dish FEBI with Horn Only Using IE Regions

Dish FEBI with horn only [Driven Modal]

In this design instead of making the dish an IE region, a FEBI boundary is assigned on the air boxes enclosing the dish and the horn.

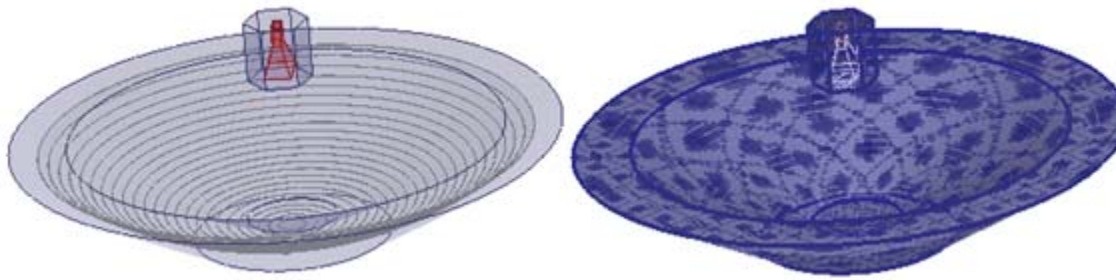


Figure 4 FEBI Boundary Enclosing Dish and Horn

Horn Source Project for PO Solver [Driven Modal]

The last HFSS design is the source project with the horn only. Default radiation boundary is used. So, HFSS will use scattered field formulation since Radiation Only is selected. A wave port is

assigned on this horn as shown below and then, the design is solved to generate the fields which are used in the IE designs explained in the next two sections that follow.

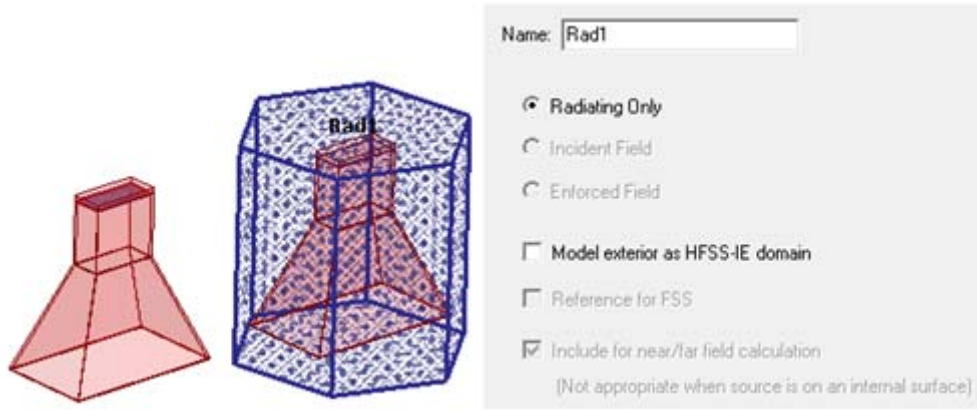


Figure 5 Horn Source Project

Dish IR Solver Imported Far Fields

Here we increase the level of approximation in comparison with designs (1), (2), and (3) by excluding even the horn (and the feed network). Using HFSS' Far Field Link as an Excitation option, we can simply impose the fields from the Horn_Source_Project (i.e. design 4) onto the dish without the coupling. As shown in the figure below we have retained only the dish in this design. Although there is no horn, interaction between the horn and the dish is kept intact by way of Far Field Incident Wave from the Horn_Source_Project. See the Excitation settings and the dish below.

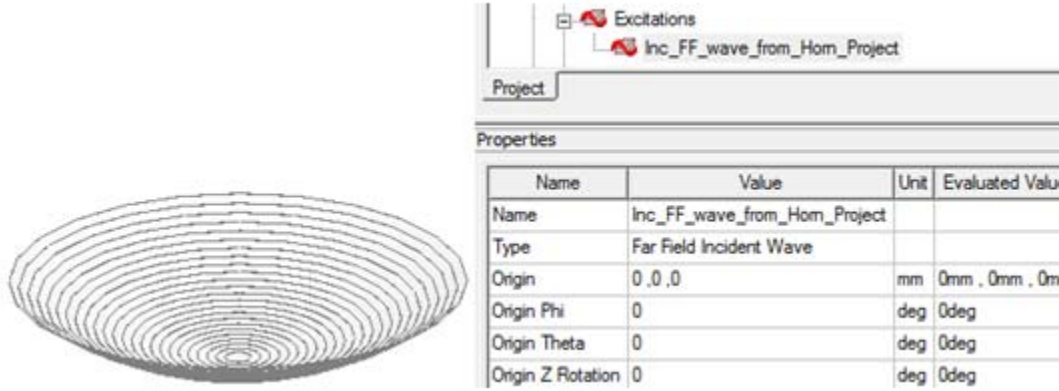


Figure 6: Dish with Solver Imported Far Fields

See the IE Solution Setup panel for the General and Options tabs shown side by side. Note the solver options used.

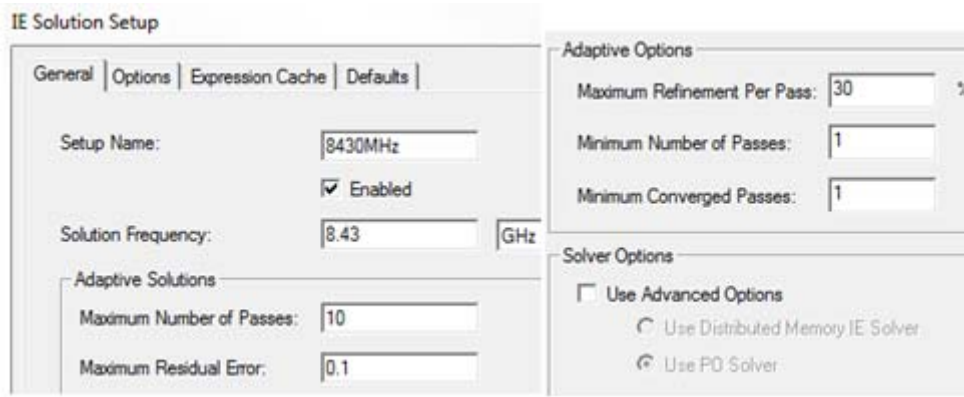


Figure 7: Setup Link Dialog

Dish Physical Optics Solver Imported Far Fields

In this design a similar level of approximation as design 5 is used i.e. we retain only the dish and impose the Far Field Incident Wave from the Horn_Source_Project. The geometry is similar to that in design 5 except that we use a different solver for the simulation. Notice the settings in the IE Solution Setup dialog box where we have selected the Use PO Solver radio button. Obviously, we are using the Physical Optics solver.

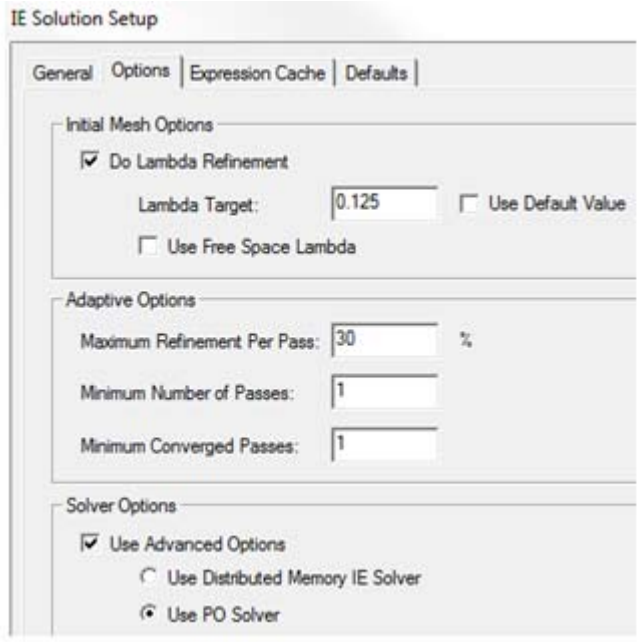


Figure 8: Dish PO Solver Imported Far Fields

Results

The E-field plot for the first project with the horn, dish, and the feed network is shown below.

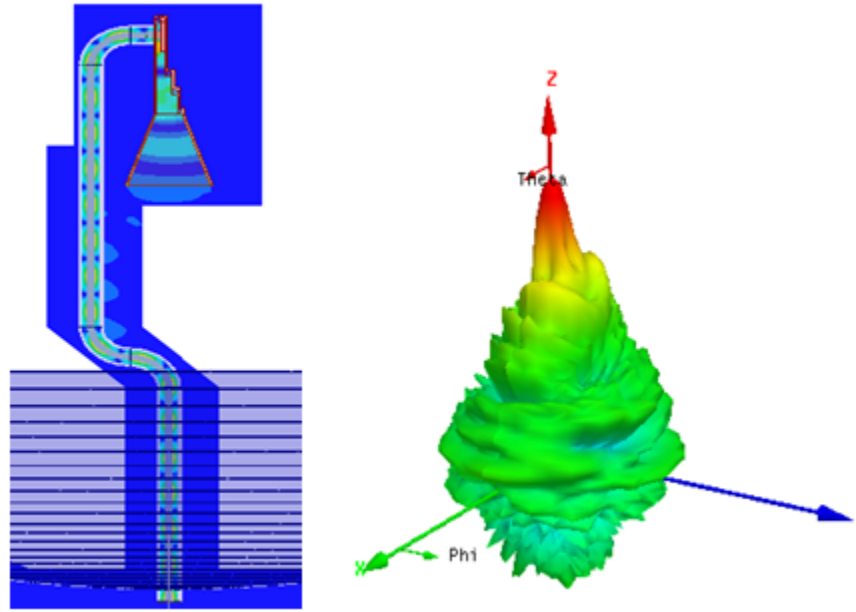


Figure 9: E-Field Plot on the Feed and the 3D Polar Plot for the First Design

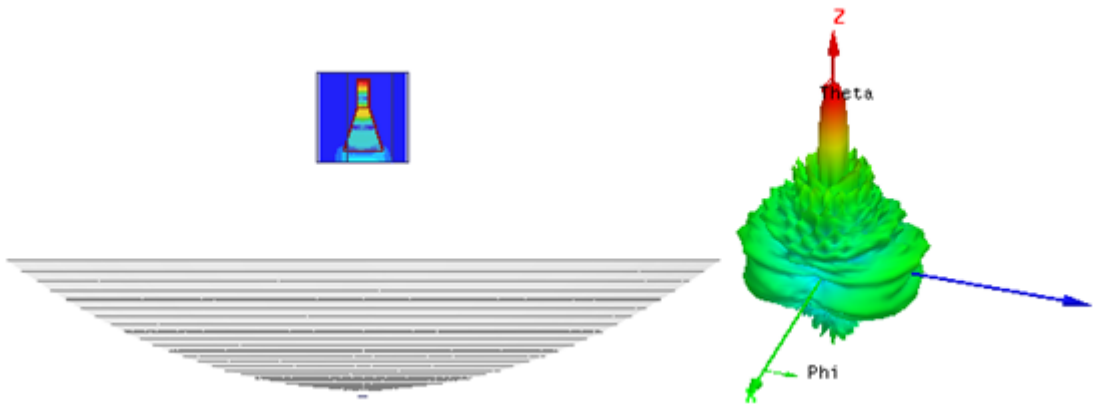


Figure 10: Dish DEBI with Only IE Using Horn and the 3D Polar Plot for the Second Design

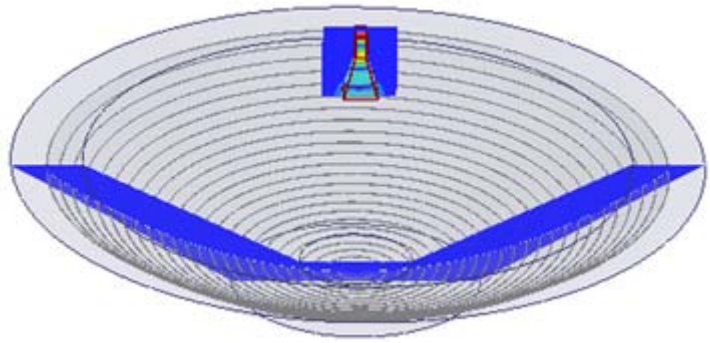


Figure 11 E-Field Plot for Dish FEBI with Horn Only

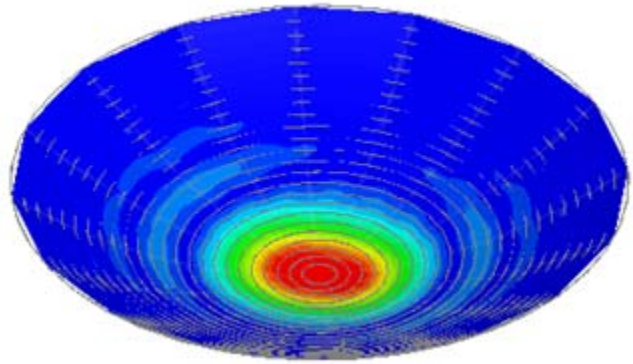


Figure 12 E Field for the Dish IE Solver Imported Far Fields

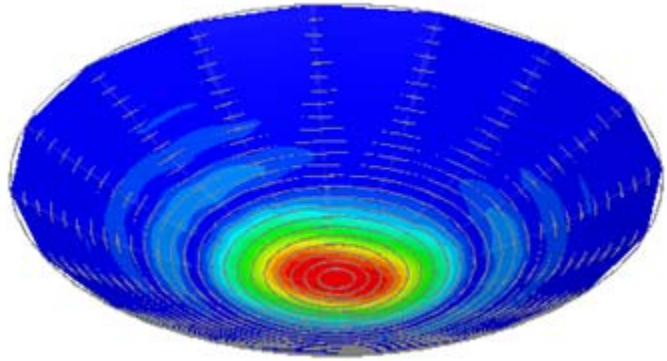


Figure 13: Dish PO Solver Far Fields Imported

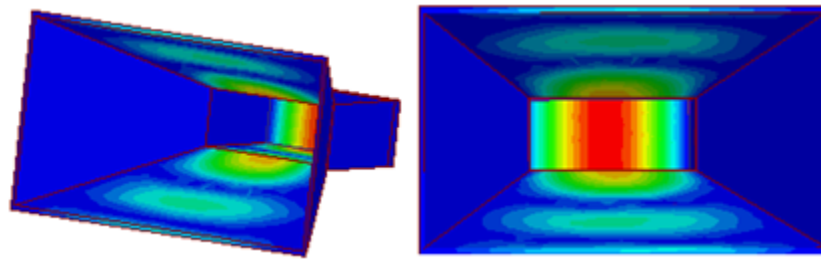
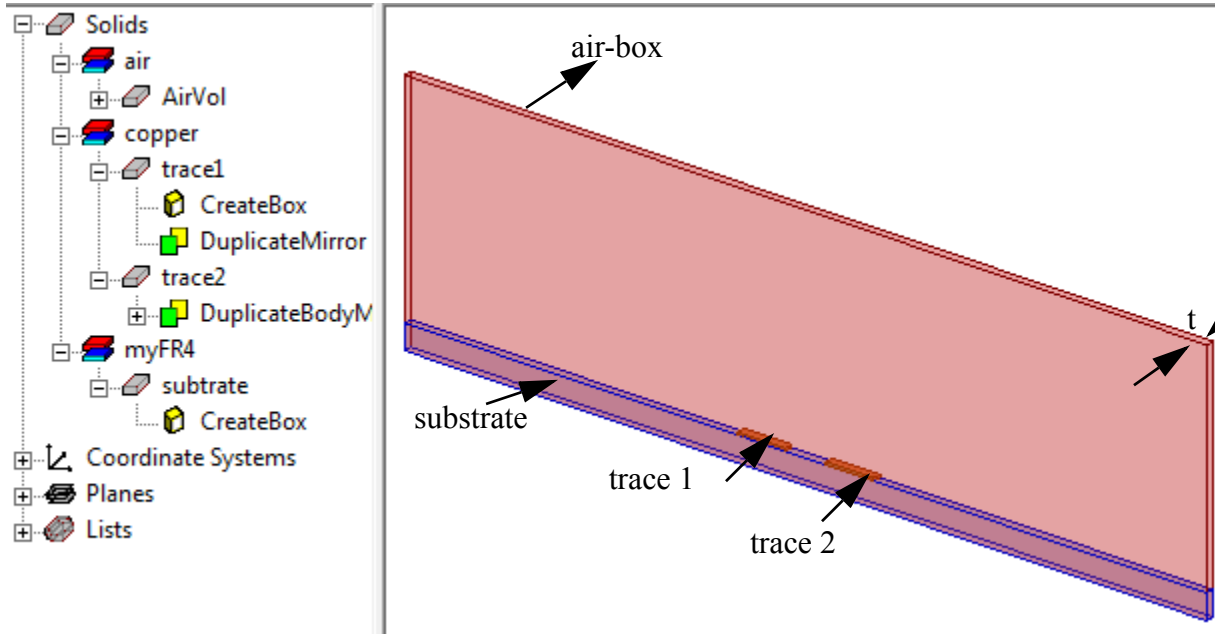


Figure 14: E-Field in Horn Antenna

Differential Pair Microstrip

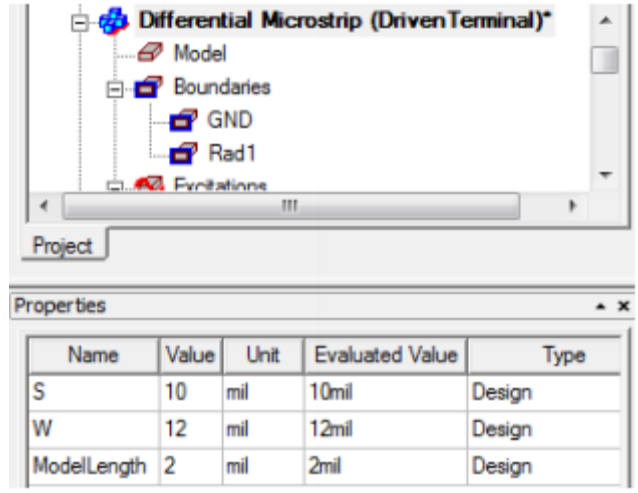
Description

The differential microstrip design comprises two copper traces and an FR4 substrate enclosed in an air-box. The design in the following figure describes an efficient way to model a long transmission line without explicitly drawing the desired length of the microstrip model. HFSS offers a post processing feature called *Deembedding* that can be used to calculate the transmission line characteristics by moving the reference plane of the wave port by the specified deembedding distance.



Differential microstrip

The dimensions of the copper traces, air-box, and substrate are defined by using variables. If you click the design name on the **Project Manager** window, the **Properties** window displays all the design variables.

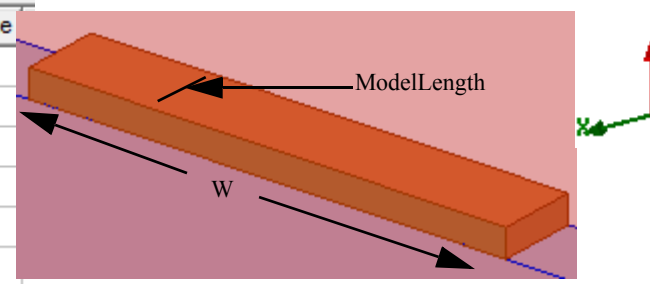


Name	Value	Unit	Evaluated Value	Type
S	10	mil	10mil	Design
W	12	mil	12mil	Design
ModelLength	2	mil	2mil	Design

Variable list

Double-click **CreateBox** under **trace1** on the history tree to see the dimensions of the copper trace.

Name	Value	Unit	Evaluated Value
Command	CreateBox		
Coordinate Sys...	Global		
Position	0mil ,S/2 ,0		0mil , 5mil , 0
XSize	ModelLength		2mil
YSize	W		12mil
ZSize	0.7	mil	0.7mil



trace1

In this model, trace2 was created from trace1 by right-clicking **trace1** and selecting **Edit > Duplicate > Mirror.** In the coordinate text boxes of the status bar, starting co-ordinates for the base position was set to 0, 0, 0 and the normal position was set to 0,-1,0.

Name	Value
Command	DuplicateMirror
Coordinate Sys...	Global
Base Position	0 ,0 ,0
Normal Position	0mil ,-1 ,0mil

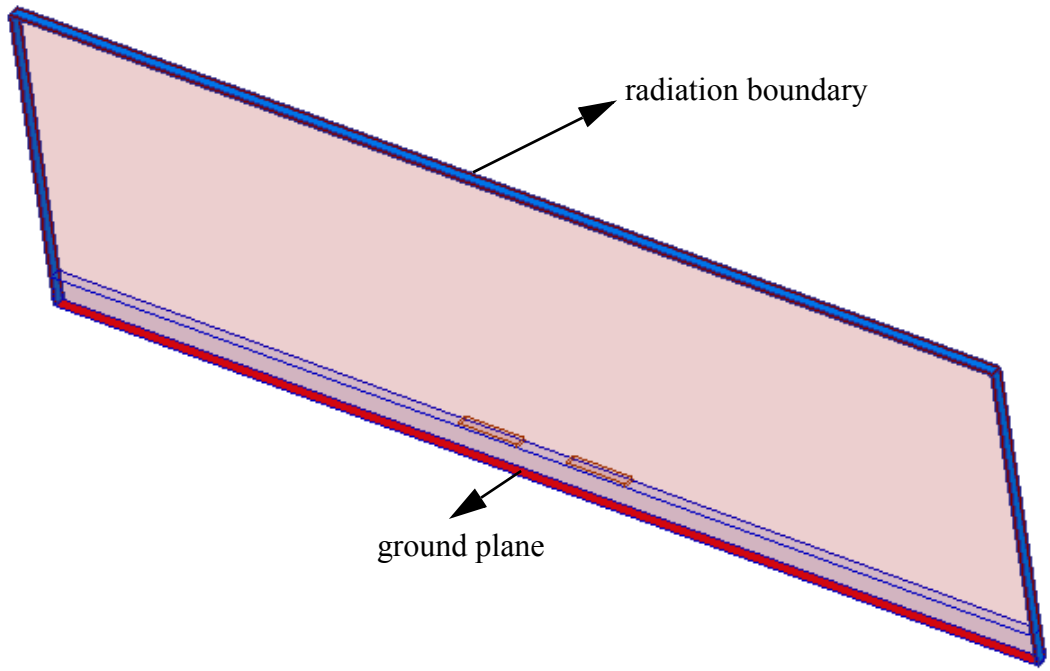
Boolean operation command window

By parameterizing with common variables and taking advantage of the model tree you can create efficient designs in HFSS. Changing the value of one or more common variables ensures individual objects to track with the geometry of the entire model appropriately.

Boundaries and Excitations

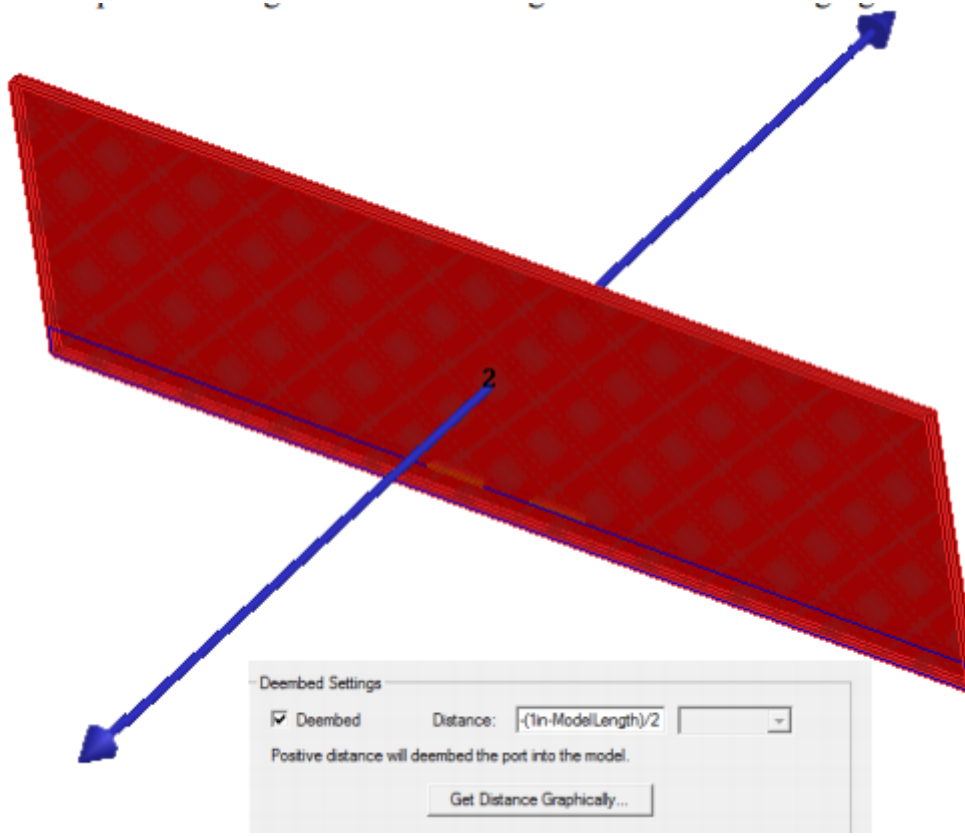
On the **Project Manager** window, click **GND** and **Rad1** to see the ground assigned on the bottom face of this design, and the radiation boundary assigned on the top face and along the ZX faces as shown in the following figure.

Note: You can also select the **Boundary Display (Solver View)** option from the **HFSS** menu item and select the boundaries, ports, and terminals that you want to see in the design. In the following figure, the ground and the radiation boundary appear highlighted since they are selected on the **Boundary Display (Solver View)** dialog box.



ground and radiation boundary

Observe how the two ports are assigned in the remaining faces in the following figure.



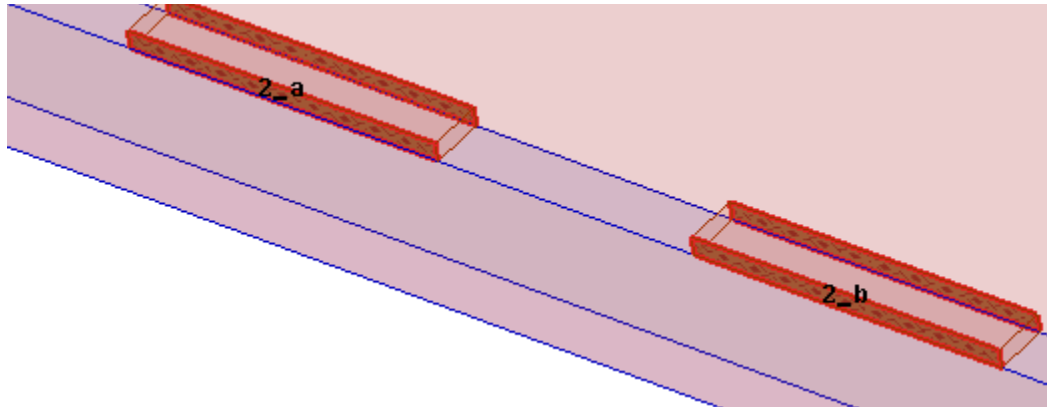
wave ports and deembedding

Note For guidelines on defining port size, see the section **Assign Wave Ports for Terminal Solutions** in the online help.

The ports are deembedded with a negative distance outwards from the structure. The intent is to solve the model of this minimal length and then deembed outwards from the ports using a negative sign for the deembed distance to effectively add the extra length that you want to represent the actual length of the model.

Note For more information about modeling long transmission lines, see the **Applications for Deembedding** section in the **Assign Excitation** material in the online help.

There are 4 terminals assigned as shown in the following figure.



terminals

Right-click **Excitations** and select **Differential Pairs** to access the **Differential Pairs** dialog box.

Differential Pairs

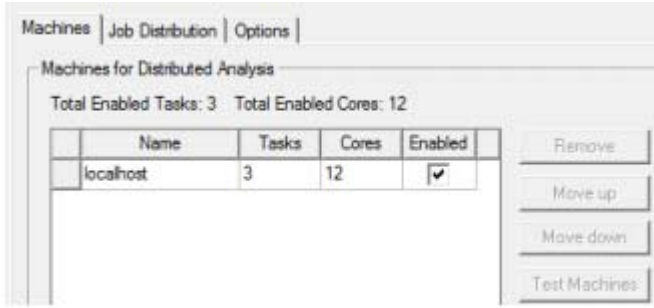
Terminals:				Differential Mode:		Common Mode:	
Positive	Negative	Enabled	Matched	Name	Ref. Z	Name	Ref. Z
1_a	1_b	☑	☐	D1	100ohm	C1	25ohm
2_a	2_b	☑	☐	D2	100ohm	C2	25ohm

differential pairs

HPC Analysis and Solution Setups

Run this design at an adapt frequency of 20 GHz. Since a parametric sweep (look for **Spacing-Sweep** under **Optimetrics** on the **Project Manager** window) is defined, this design is a good choice for which HPC can be set up. On the **Solution Setup** dialog box, click the button to open the **HPC and Analysis Options** window. Click **Add** and set the number of tasks and cores.

For example in HPC setup in the following figure, the design was simulated on a machine with 12 cores on it and the Number of Tasks is 3. In such a setup, 3 frequency points are solved in parallel with 4 cores of matrix multiprocessing per frequency point.



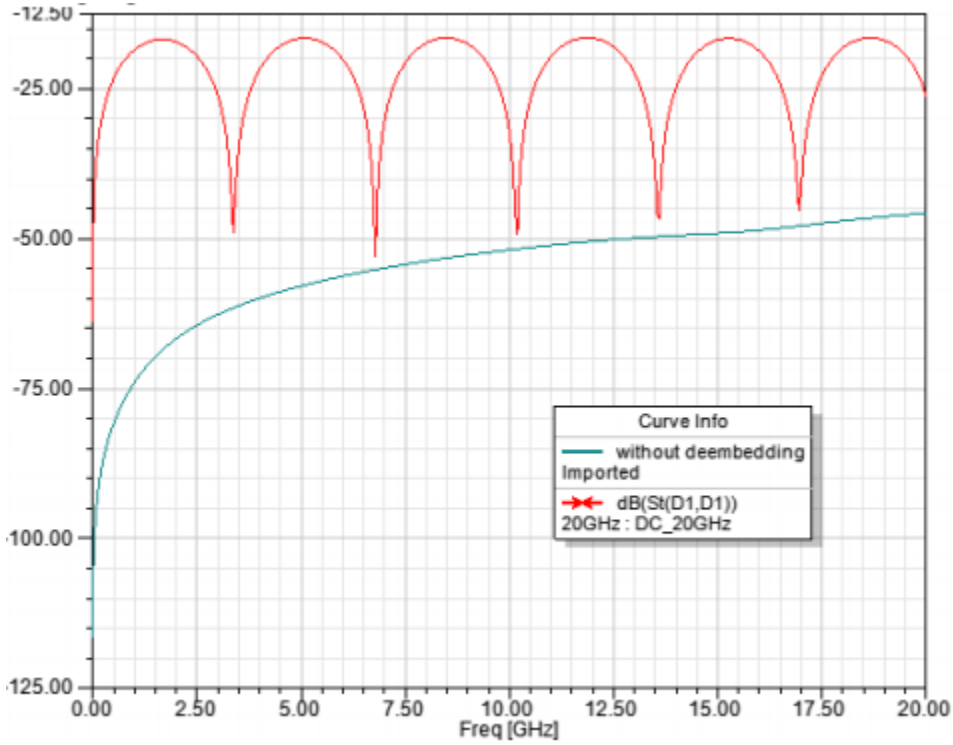
HPC setup

Note For more information about HPC, see **HPC and Analysis Configuration Options** section in the online help.

While deembedding simplifies modeling long differential striplines and makes the solution process efficient, the HPC setup further accelerates the simulation process.

Results

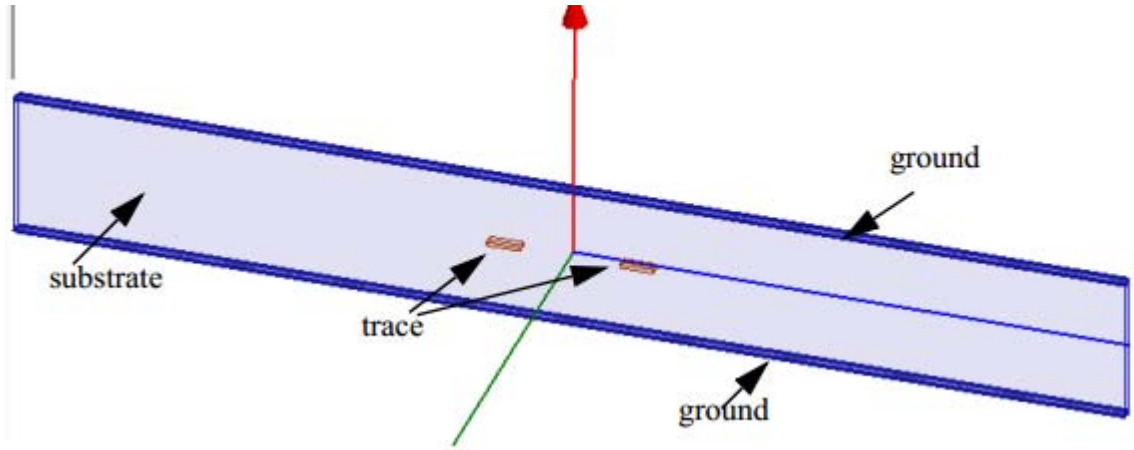
The results of the S-parameter plots with and without deembedding are shown below. The deembedding operation adds the effect of phase delay and additional dielectric and conduction losses to the resulting S-parameter calculated from this model.



results

Differential Pair Stripline

The following figure shows a differential pair stripline design, where two copper traces are embedded in the substrate, which in turn is sandwiched between two ground conductors. Select **GNDs** on the project tree of the **Project Manager** window to highlight the ground planes assigned on the top and bottom faces of the stripline. These top and bottom ground planes are equipotential surfaces.

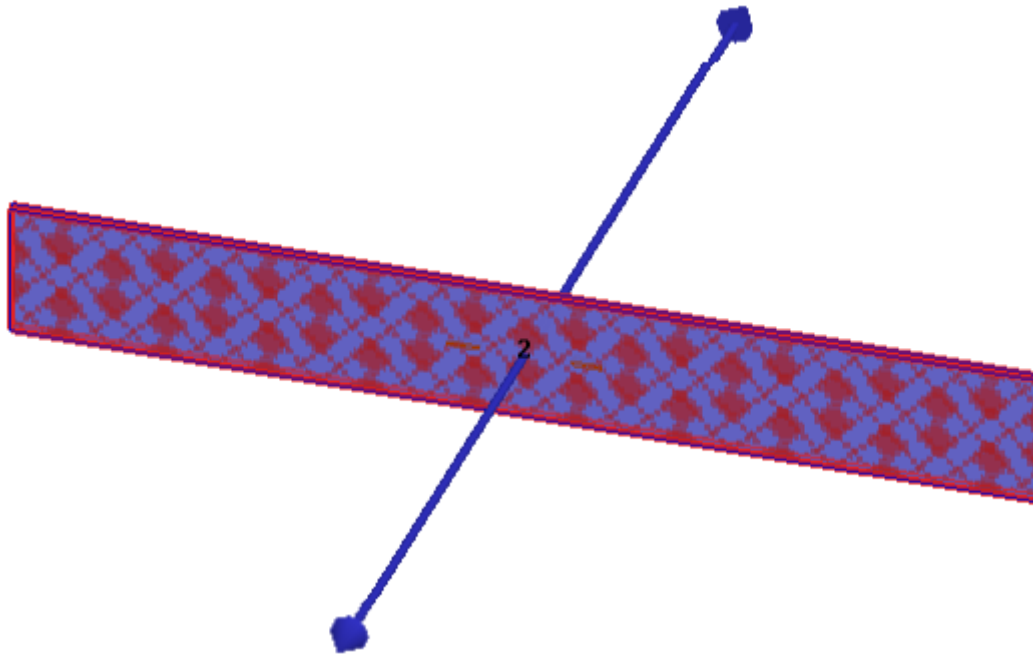


stripline

The intent of this design is to draw only minimal lengths of the differential traces containing the two conductors adjacent to each other (and the equipotential ground conductors) without explicitly drawing the actual length throughout the entire trace route. By using a post processing feature of *Deembedding*, transmission line characteristics can be calculated by moving the reference plane of the wave port to desired locations along the trace route, depending upon the specified value of the deembed distance. Such a design is an effective approach to simulate the actual model length. It saves simulation time and uses minimal computational resources.

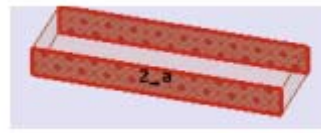
Excitations

To see both excitations at the same time, right-click the **Excitations** option on the **Project Manager** window. Select **List** from the shortcut menu, and click the two wave ports listed in the **Design List** dialog box. The waveports assigned in the model with the deembedded lengths appear highlighted in the design as shown in the following figure.



deembedded wave ports

The deembed arrows point outwards from the structure since negative deembedding value of -(1in - ModelLength)/2 was specified for each wave port. The Deembed Distance value is set on the **Post Processing** tab of a wave port dialog box. The main purpose of such a design is to solve the model of minimal length and then, by deembedding outwards from the ports, to represent the actual length of the model. The terminals on a trace are shown in the following figure.

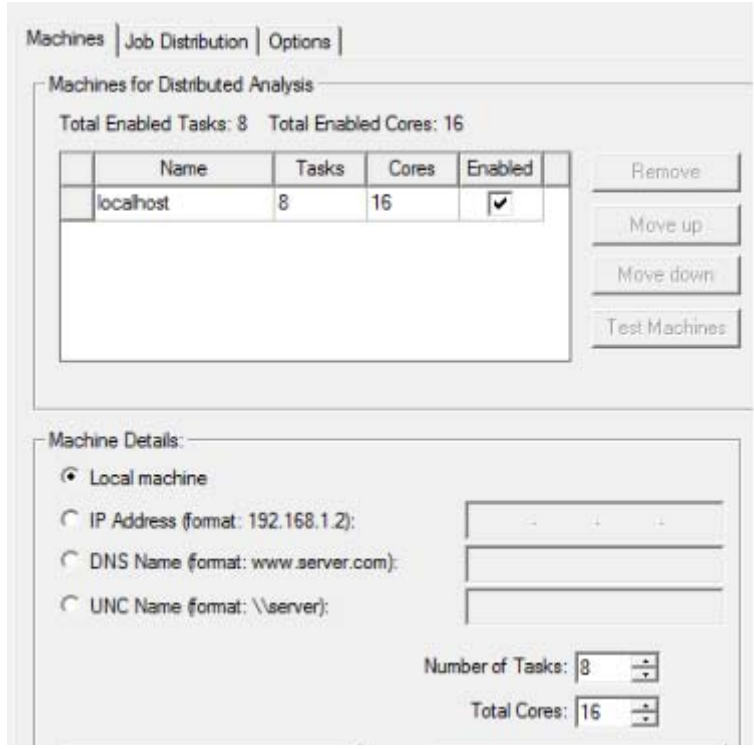


terminals on a trace

Solution Setup and HPC Analysis

Run this design at a solution frequency of 20 GHz. Since the design has a parametric sweep (with the trace edge-to-edge spacing defined by the variable S), it is a good choice for setting up HPC analysis. From the **Solution Setup** dialog box, click the button to open the **HPC and Analysis** win-

dow. Click the **Add** button to open the **Analysis and Configuration** window, where you can set the number of available cores to use for this design.



HPC setup on the Analysis and Configuration window

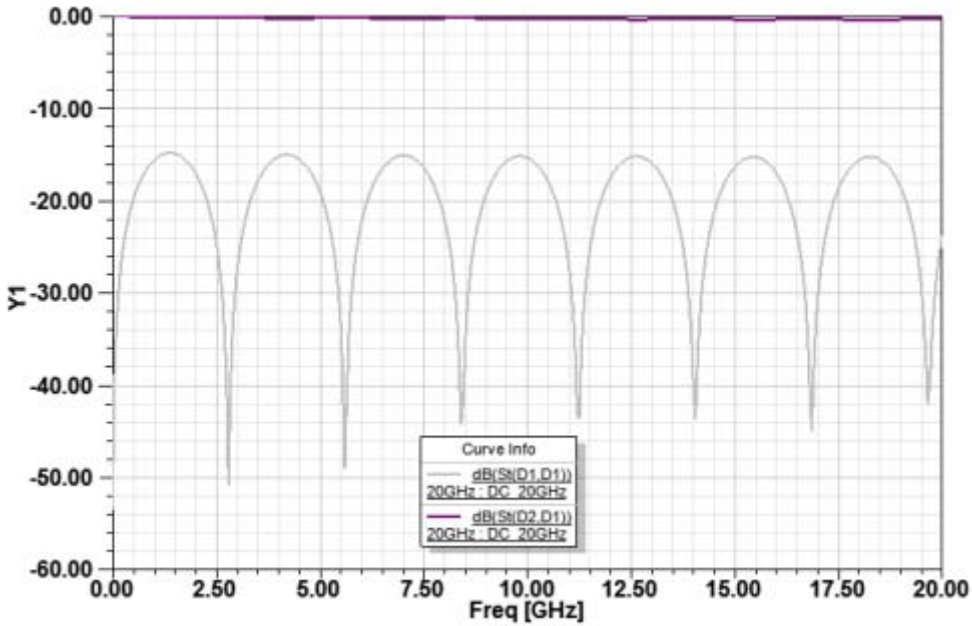
For example, in figure above, 16 cores are available on the machine in which the design was simulated and number of tasks is 8. In such a setup, the sweep is run with 8 frequency points solved in parallel by using two cores of matrix multiprocessing for each frequency point. When such an analysis is executed on a single machine, the simulation is very efficient if the machine has enough shared memory to accommodate 8 simultaneous solves. Otherwise the analysis can be performed across multiple machines (that have HFSS installed in them) without requiring any additional HFSS license for each machine

Note For more information about HPC, see **HPC and Analysis Configuration Options** section in the online help.

While deembedding simplifies modeling long differential striplines and makes the solution process efficient, the HPC setup further accelerates the simulation process.

Results

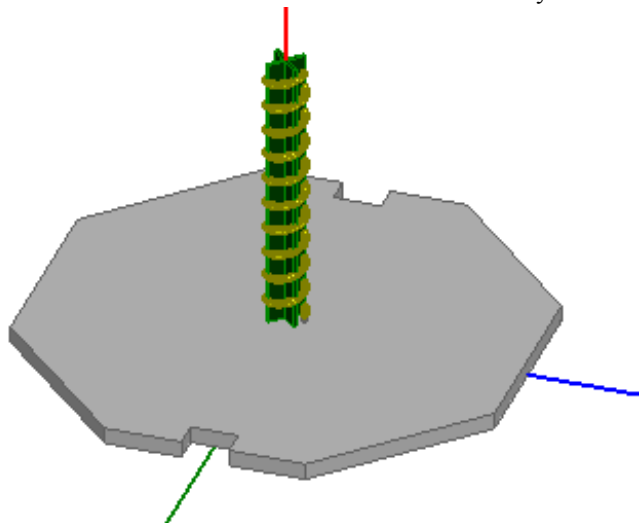
The following figure shows the S-parameter plots for the stripline..



s-parameter plot

Helical Antenna

Description - a coax fed helical antenna with a dielectric support on a finite ground plane. The antenna is designed to run at 3.5 GHz. A smaller virtual object is defined as the integration surface for radiated field calculations. This is surrounded by an air box with a radiation boundary.*



Model - the support is made of Teflon and the ground has as thickness of 0.5 in. The coax port is internal and is capped by a conducting object. You can create a helix similar to this by using **Draw>User Defined Primitive>SysLib>Segmented Helix.**

Setup - adapt at 3.5 GHz and use mixed order for Order of Basis function. Since this model has open air regions and the tightly spaced helix it is a good choice for mixed order.

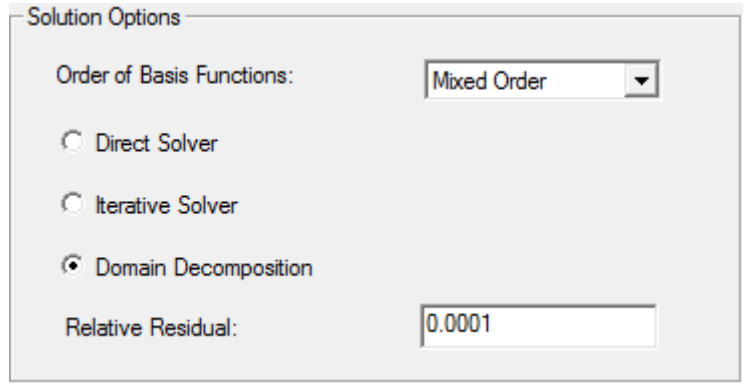
Note

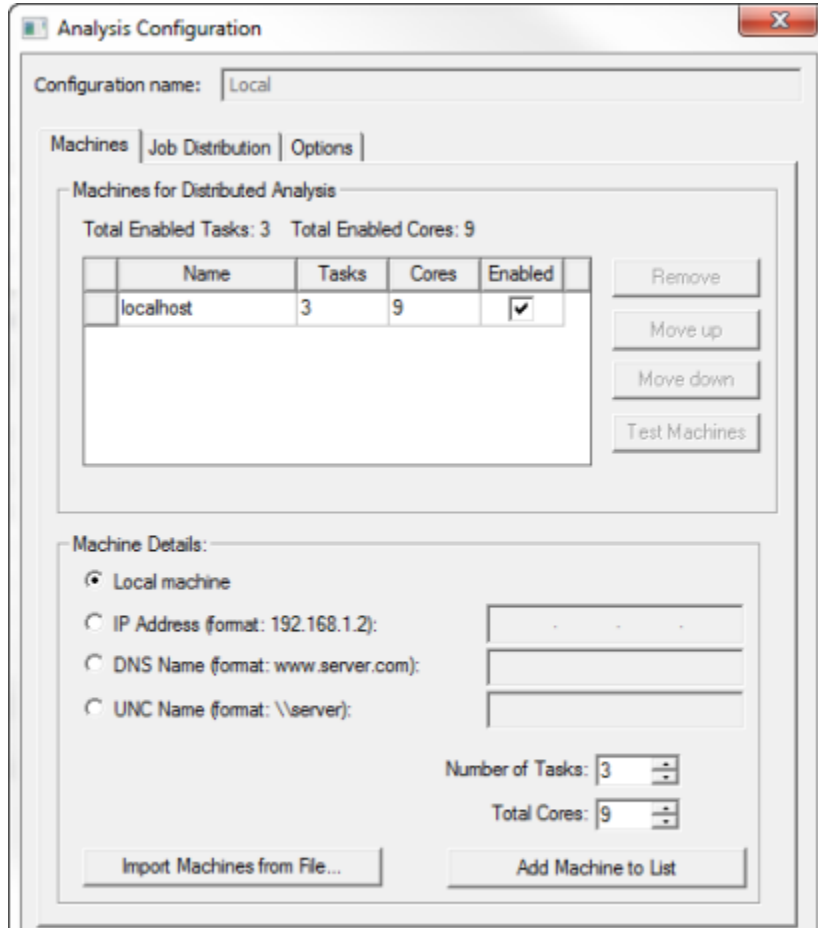
To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed in the Properties window. Selecting an object in the History tree will also display its properties.

For a further discussion on using integration surfaces and for more on creating sample antenna designs, see the antenna design kit at https://support.ansys.com/AnsysCustomerPortal/en_us/Downloads/Current+Release/Tools.

HPC Analysis and Solution Setup for Helical Antenna

Solve this design at an adapt frequency of 3.5 GHz and use **Mixed Order** for **Order of Basis Functions**. This model has open air regions and the tightly spaced helix, so it is a good choice for mixed order. Since this is a large problem, you can enable **Domain Decomposition** so that HFSS automatically partitions the design into domains and solves them by separate processes. These processes can be executed on separate networked machines, allowing the problem to be solved with distributed memory. Before enabling solver domains, set up HPC on the **Analysis Configuration** window so that there are at least 3 tasks dedicated for this simulation. You can access the window from **Tools > Options > HPC and Analysis Options** and clicking the **Add** button.



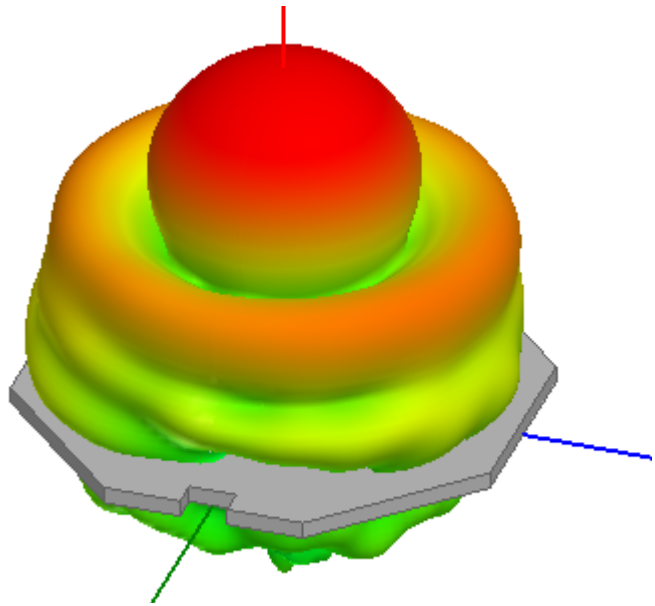


Note For details see the Enable Domain Decomposition and Setting HPC and Analysis Options sections in the online help.

Post Processing

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

To view 3D plot of the antenna gain, look in the Project Tree under Results and double click on 3D Polar Plot 1. To overlay the 3D plot on the model, click **HFSS>Fields>Plot Fields>Radiation Field** to display the **Overlay radiation field** dialog. Check **Visible** for 3D Polar Plot 1, and set the transparency and scale as desired.



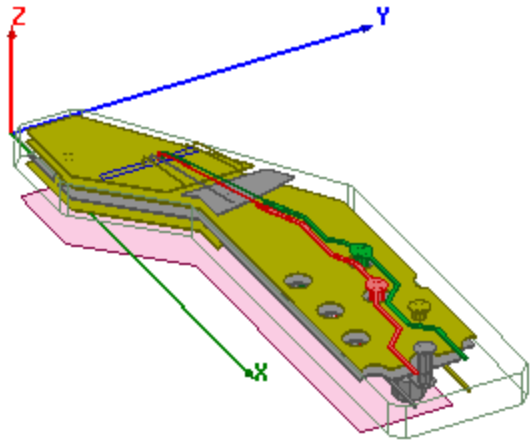
To view a 2D plot of the total gain, in the Project tree, double-click on Results - Radiation Pattern 1.

To view a 2D plot of the circular polarization pattern for this antenna in the $\phi = 0^{o}$ cut, in the Project tree, double click Results - Radiation Pattern 2.

You can add markers to the Radiation Pattern plots by right-clicking on the plot window and choosing **Marker>Add Marker**.

Package Section

Description - a model of a section of a complex package. The model was created using Ansoft Links from an ANF file and is a driven terminal design.



Model - the model is two traces of a larger package. At one end are the bond wires that are excited by lumped ports. The opposite ends terminate in ports at the solder balls. The substrate is FR4 and a radiation boundary has been applied to the surface of the surrounding airbox.

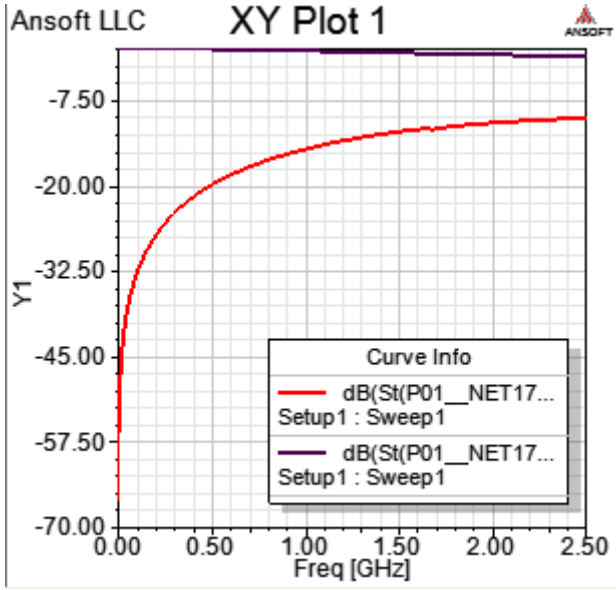
Setup - driven terminal setup with an adapt frequency of 2.5 GHz. An interpolating sweep is also included that has an upper frequency of 2.5 GHz and uses DC Extrapolation at the lower end.

Note To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed in the Properties window. Selecting an object in the History tree will also display its properties.

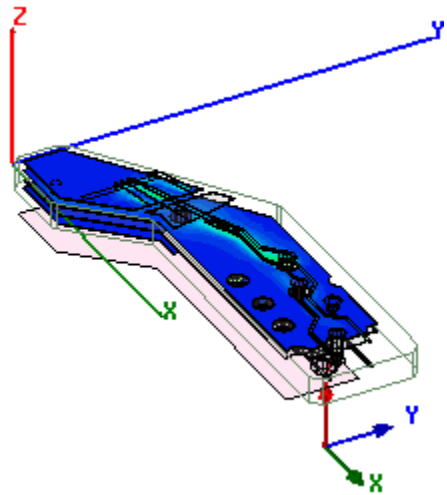
Package Post Processing

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

To view the S parameter plot show below, double click on XY plot1 in the Project Tree under Results.

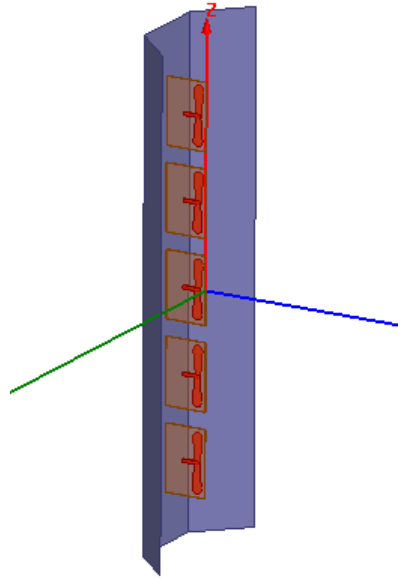


To view the shade plot of jsurf, the surface current density, on one of the package planes, right click on Jsurf under Field Overlays in the Project Tree and select update.



Small Array of Planar Flared Dipole Antennas

Description - A five element array of flared dipole antennas with a trough reflector. A post processing variable has been defined that controls the element to element source phases, allowing the resulting beam to be easily steered.



Model - The elements use a teflon substrate with 1.6 mm thickness. The excitations are lumped ports. A radiation boundary is defined on an air box that is not show here. The post processing variable is called phase_shift.

Setup - Adapt at 1.9 GHz.

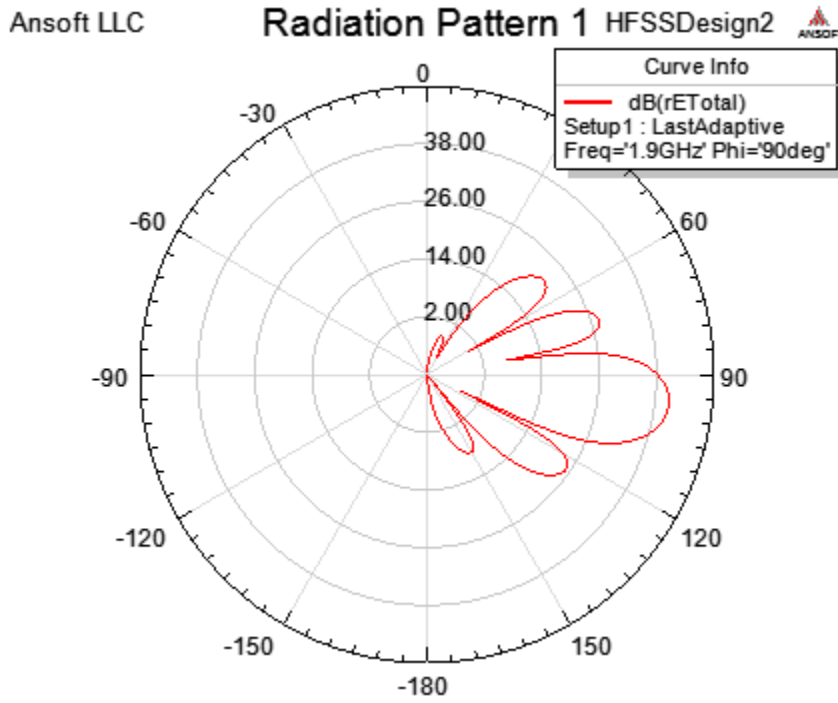
Note To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed in the Properties window. Selecting an object in the History tree will also display its properties.

Post Processing Small Array of Planar Flared Dipole Antennas

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

To view the $\phi = 90^{o}$ radiation pattern double click on Radiation pattern 1 under Results in the Project tree. As stated the relative element-to-element phase shift in degrees for this array is controlled

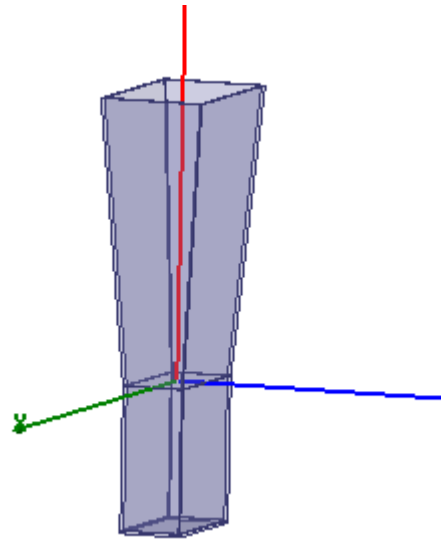
by the phase_shift post processing variable. To verify this, go to **HFSS>Fields>Edit Sources** where you will see the offset phase, defined as multiples of the variable phase_shift.



The easiest way to adjust the phase_shift variable is to click on the design name, HFSSDesign2, in the Project tree. You will see the design variable displayed in the Properties window. You can change the value of phase_shift in this window. The plot shown here is for a setting of 30^{o}.

Pyramidal Horn

Description - A pyramidal horn designed to operate at 10 GHz. The feed is an x-band waveguide.This project uses PMLs for its radiation boundary.



Model - The horn has 0.02" wall thickness and is defined as PEC. The port is internal to the solution region and is capped by a PEC object (cap). PMLs are included in the model and are defined on the faces of air_box. Their visibility, by default, has been turned off.

Setup - Adapt at 10 GHz.

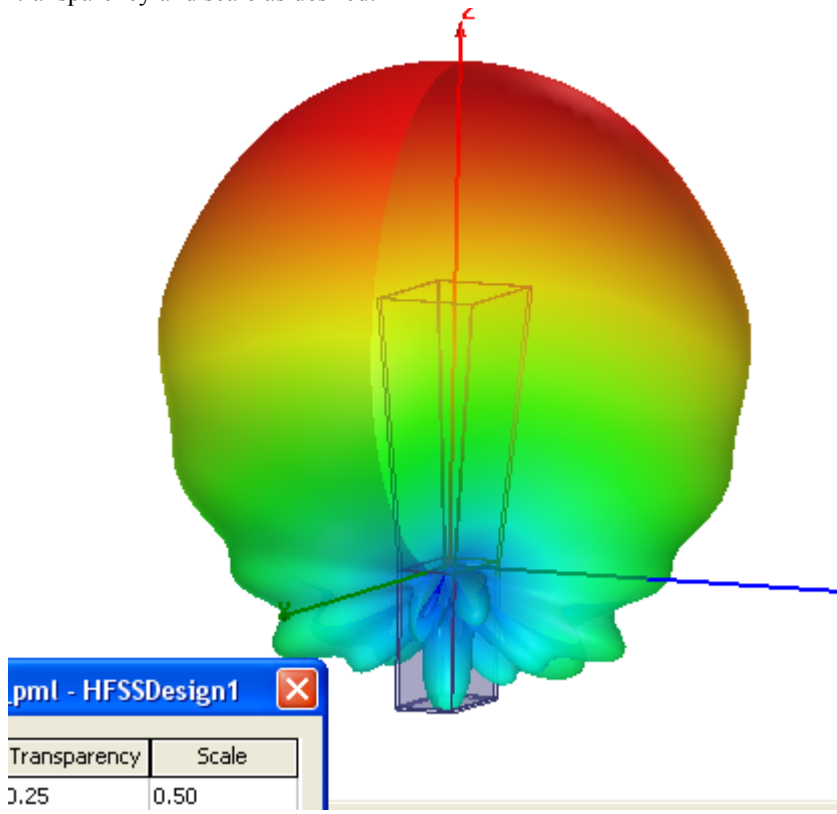
Note To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed in the Properties window. Selecting an object in the History tree will also display its properties.

Pyramidal Horn Post Processing

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

To view 3D plot of the antenna gain, look in the Project Tree under Results and double click on 3D Polar Plot 1. To overlay the 3D plot on the model, click **HFSS>Fields>Plot Fields>Radiation**

Field to display the **Overlay radiation field** dialog. Check **Visible** for 3D Polar Plot 1, and set the transparency and scale as desired.



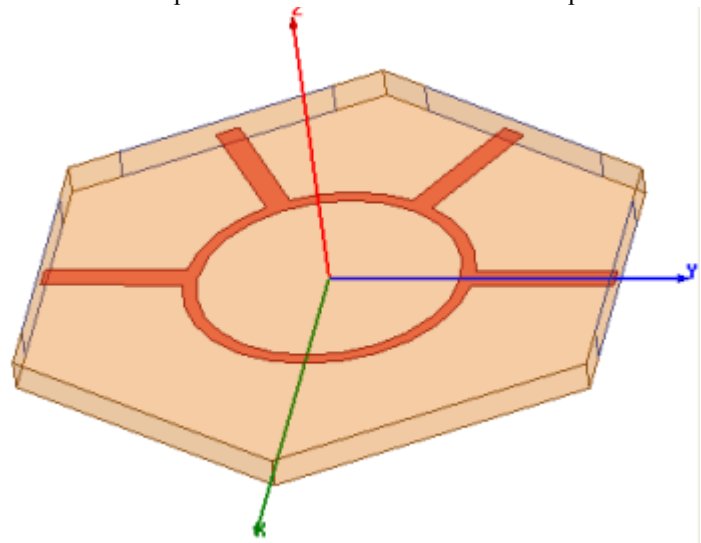
To view a 2D plot of the dB gain total, look in the Project Tree under Results, and double click on Radiation Pattern 1

A shade plot showing the fields on the Y-Z plane has also been created and can be viewed by right-clicking on E Field under Field Overlays and selecting **Update**.

Ring Hybrid

Description - a ring hybrid that can be used as a splitter. Power input to the Sum port splits equally between Ports 2 and 3 in phase while power input to Difference Port splits with 180^{o} phase shift. It

can be used as a power combiner with power input to Ports 2 and 3. In that case, the sum of the two inputs will be seen at Sum port and the difference at Difference port.



Model - This is a hybrid designed to operate at 10 GHz using stripline transmission lines. The substrate is 40 mil Duriod 5880. The ports are defined on four of the faces of the substrate. Default outer boundary boundary (PerfE) is defined on the remaining faces. The trace is a 2D object with a PerfE boundary.

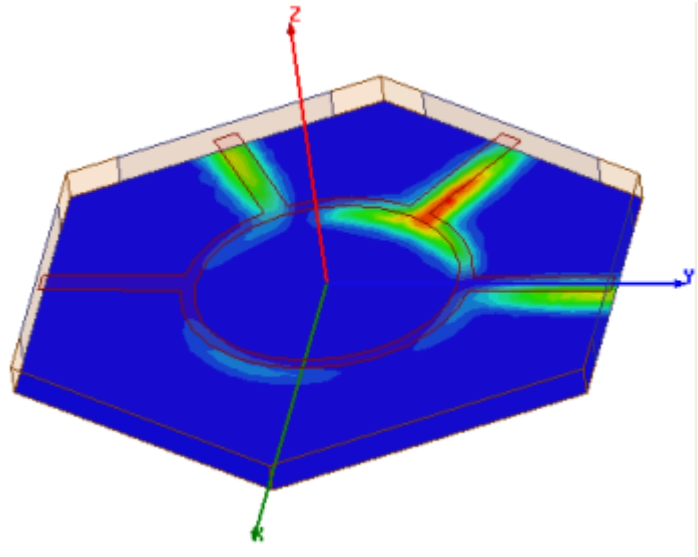
Setup - Adapt at 10 GHz with interpolating frequency sweep from 8 to 12 GHz. Also, the model includes an output variable called Diff_port_phase that computes the difference in phase between ports 2 and 3 when the Difference port is excited.

Note To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed. Selecting an object in the History tree also displays its properties.

Ring Hybrid - Post Processing

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

To view plots of S parameter data you can open XY Plot 1. For a plot of Diff_port_phase, open XY Plot 2. To view a shade plot, double click on Mag_E1 under E Field under Field Overlays.



To view the shade plot as shown, change excitations so that ports 2 and 3 are excited by equal amplitude and phase, using **HFSS>Fields>Edit Sources**.

To view a phase animation of the resulting shade plot, right click on MagE1 and choose **Animation** from the menu.

Stripline (Driven Terminal)

The following figure shows the HFSS model of the GSSGSSG Stripline.

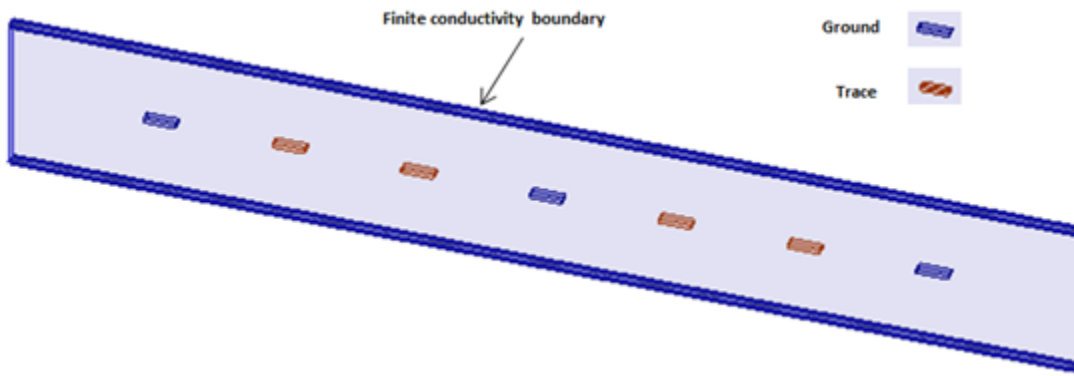


Figure 1: GSSGSSG Stripline

The figure illustrates a set of seven striplines composing a GSSGSSG dual differential pair configuration. To reduce the coupling between the differential pairs there are 3 ground conductors. Notice how the terminals are defined in the model shown below.

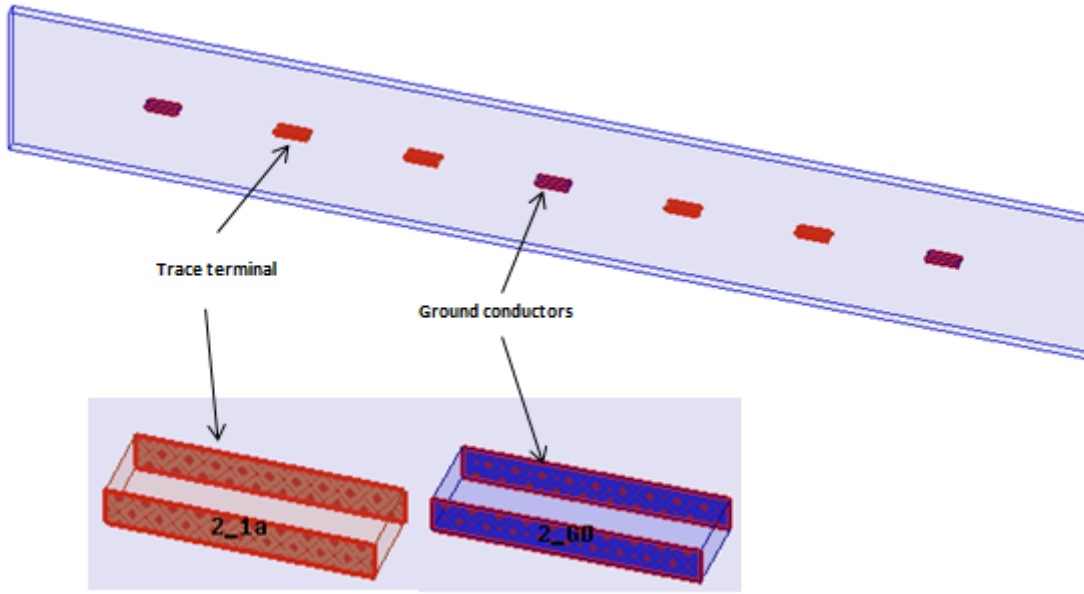


Figure 2: Terminals

The grounding is accomplished when you renormalize the impedance of the ground conductors to a very small value like 5e-6 (6 orders of magnitude below the other terminals set to 50 ohms). Notice the contrast in the values of the terminal renormalizing impedance of a trace and a ground conductor in the dialogs below.

Figure 3 Terminals Dialogs

To model a longer length, just deembed the ports as shown below.

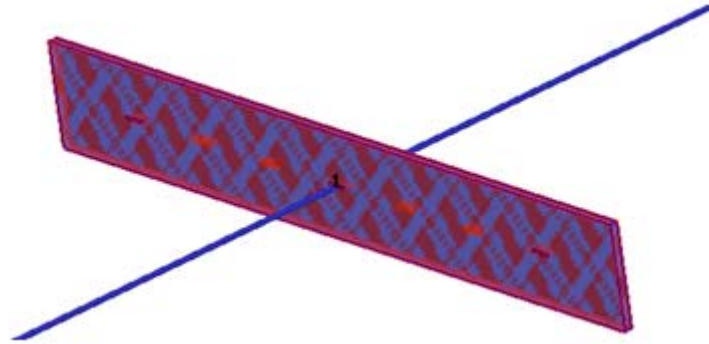


Figure 4: Ports with Deembedding

In the Post Processing tab of the Wave Port dialog box the deembedding distance = + (Simulation Length - Model Length)/2 where Simulation Length and Model Length are the design variables.

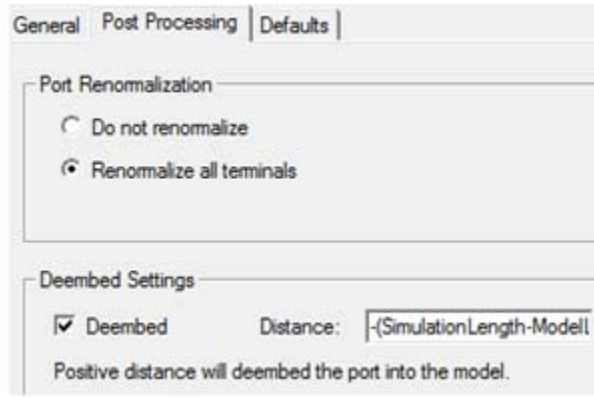


Figure 5: Post Processing Deembed Setting

The mesh plot is shown below.

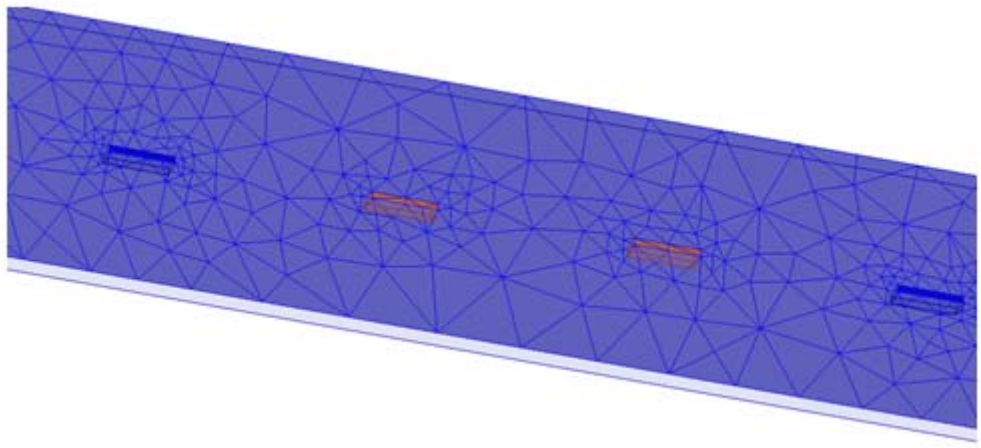


Figure 6: Stripline Mesh Plot

The results for the S-parameter plot are shown below.

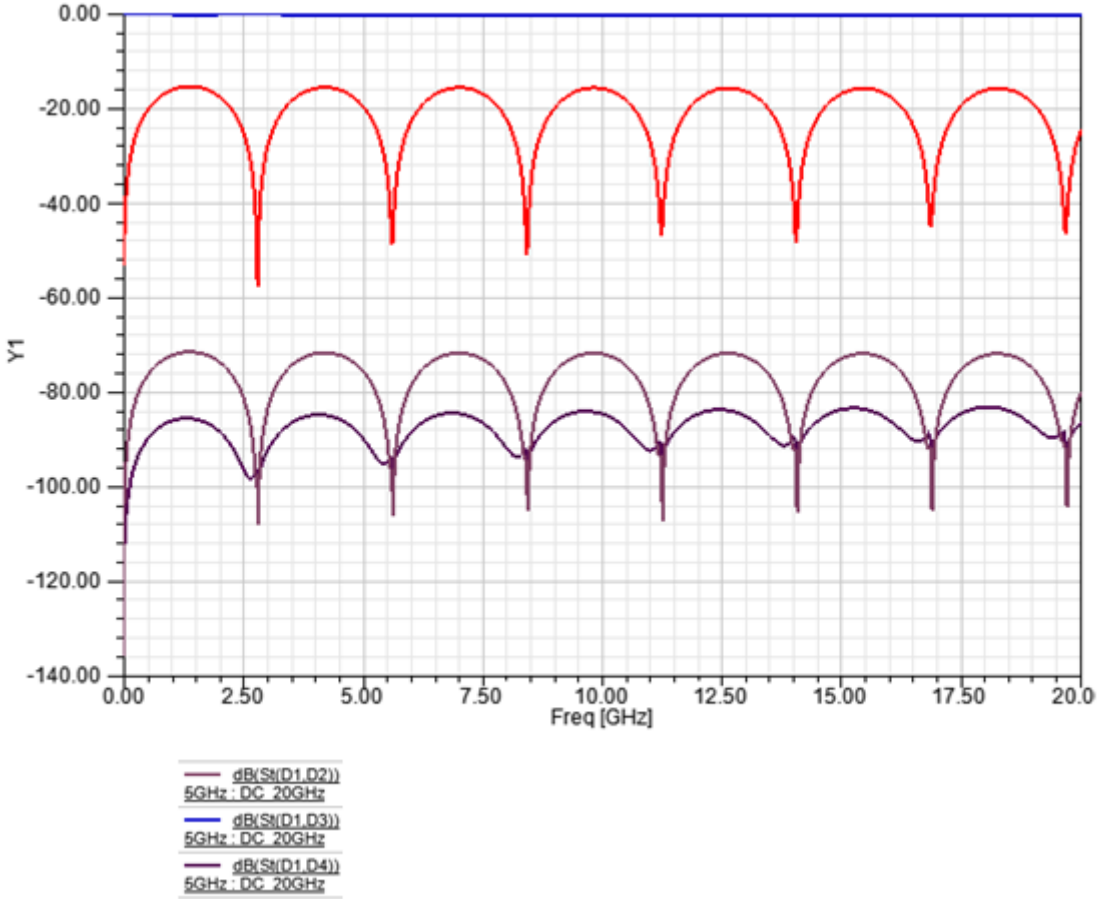
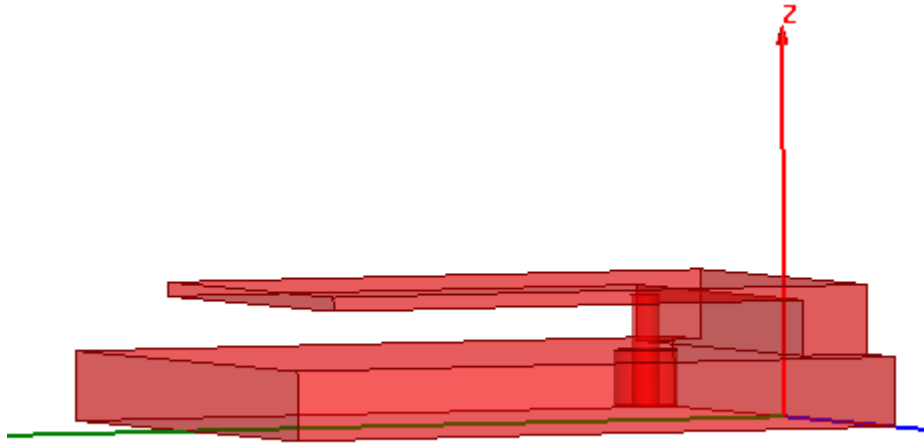


Figure 7: Stripline S-Parameter Plot

Tune a Coax Fed Patch Antenna

Description - A coax fed quarter wave patch antenna will be tuned using the Adjoint Derivatives. The antenna is shorted on one end. The design variable, feed_pos, controls the location of the coax feed along the x direction. Derivatives of the S parameter w.r.t. this variable will be computed.



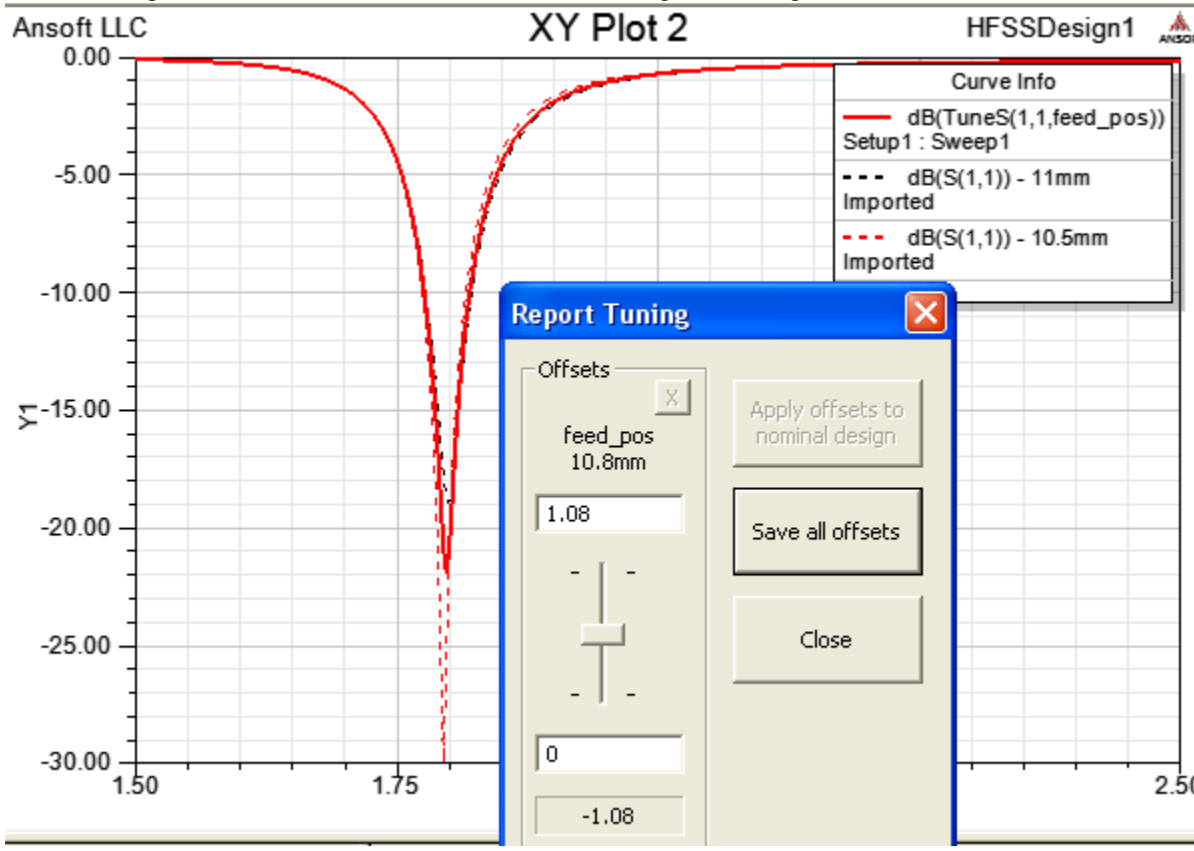
Model - The patch, shoulder, and ground plate are united into a single PEC object. The air filled coax has an internal port that is capped by the PEC ground plate. The variable feed_pos controls the value of X for the coax location in millimeters. The surrounding air box with radiation boundary is not shown.

Setup - Adapt at 1.8 GHz., and do an interpolating sweep from 1.5 to 2 GHz. The derivative of the S parameter data w.r.t. feed_pos about the default value of 10.8 mm is computed. This feature is enabled in the **Derivatives** tab of the **Setup1** window.

Post processing for tuning a Coax Fed Patch Antenna

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

To view the derivatives you can create a new plot by right clicking on Results and choosing **Create Modal Solution Data Report>Rectangular Plot**. In the **New Report Setup** window, select feed_pos in the derivative box and select the desired quantities to plot.



To tune the design, double-click on XY Plot 2 under Results to open the plot shown. In a separate project, the response for feed_pos = 10.5 and 11 mm was computed and the results from each was imported into this plot. By right clicking on Results and choosing **Tune Reports** you can open the **Report Tuning** dialog shown. As you adjust the slider, the plot is recomputed using the derivatives. Adjust the Offset to -0.3 and +0.2 to compare to the imported feed_pos = 10.5 and 11mm results.

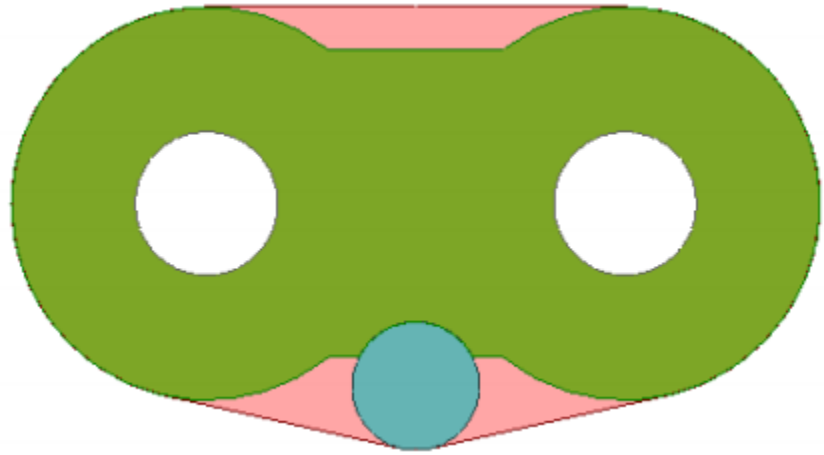
Related Topics

Derivative Tuning for Reports

Twinaxial Cable

Twinaxial cables are used for in-rack connections between supercomputers to carry the networking traffic. They are meant for the transmission of short-range signals. The following figure illustrates a twinaxial cable (28 wire gauge) design, which comprises two inner conductors (S1 and S2) made of

silver, and a shield or drain made of aluminum enclosed in an air-box. The central conductor is insulated with a dielectric layer.

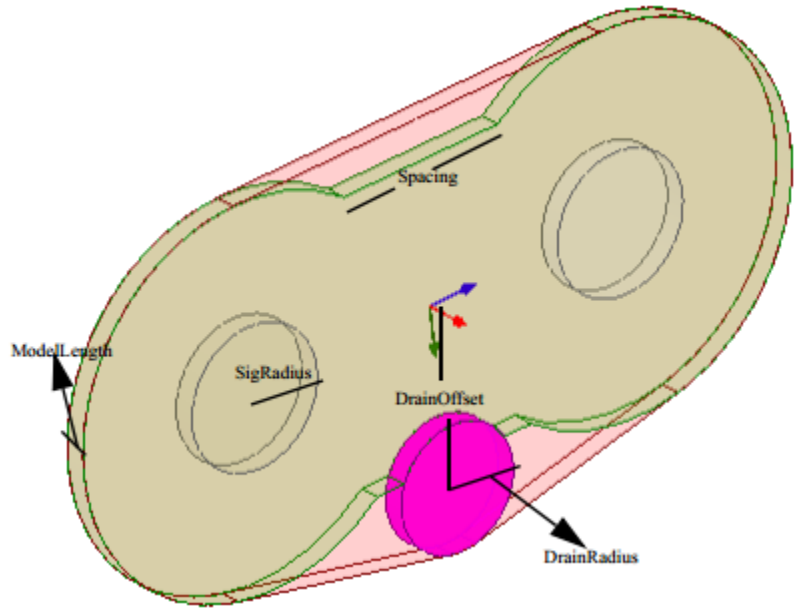


twinax cable design

The main intent is to draw a minimal length of the twinaxial cable and stretch it to any required length by using the post processing feature of *Deembedding*. Such a design saves simulation time and makes minimal use of computational resources and ensures efficient simulation, without explicitly modeling the actual length of the cable.

The design is parameterized as shown. The geometry is drawn using the parameters and boolean operations in such a way that all the individual objects that make up the geometry track with it. For example, when you change the values of the variables appearing in the **Properties** window, the

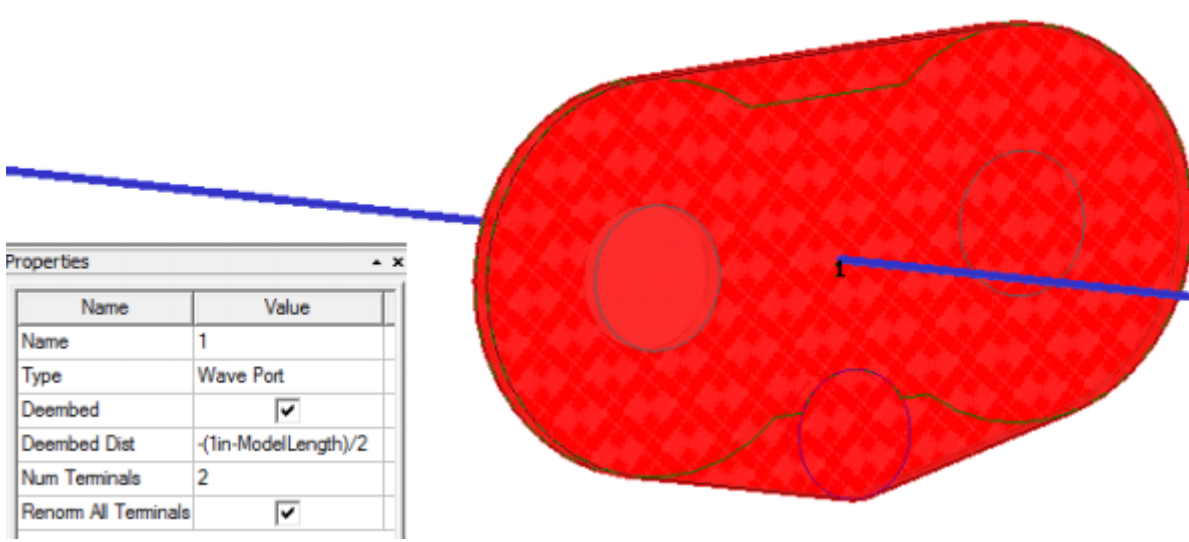
model resizes accordingly and the objects with changed parameters are track with the geometry appropriately.



some of the parameters used in the design

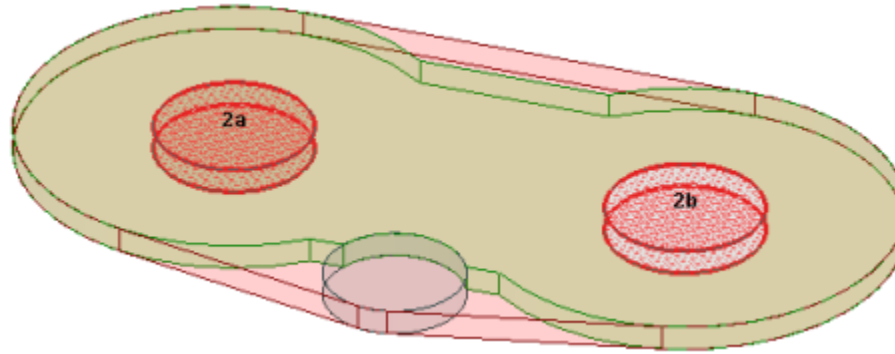
Excitations

Right-click **Excitations** on the project tree and select **List** to open the **Design List** dialog box where you can select the terminals and waveports that are assigned on this design. In the following figure, the two wave ports with their deembedding distances are highlighted.



wave ports and deembedding

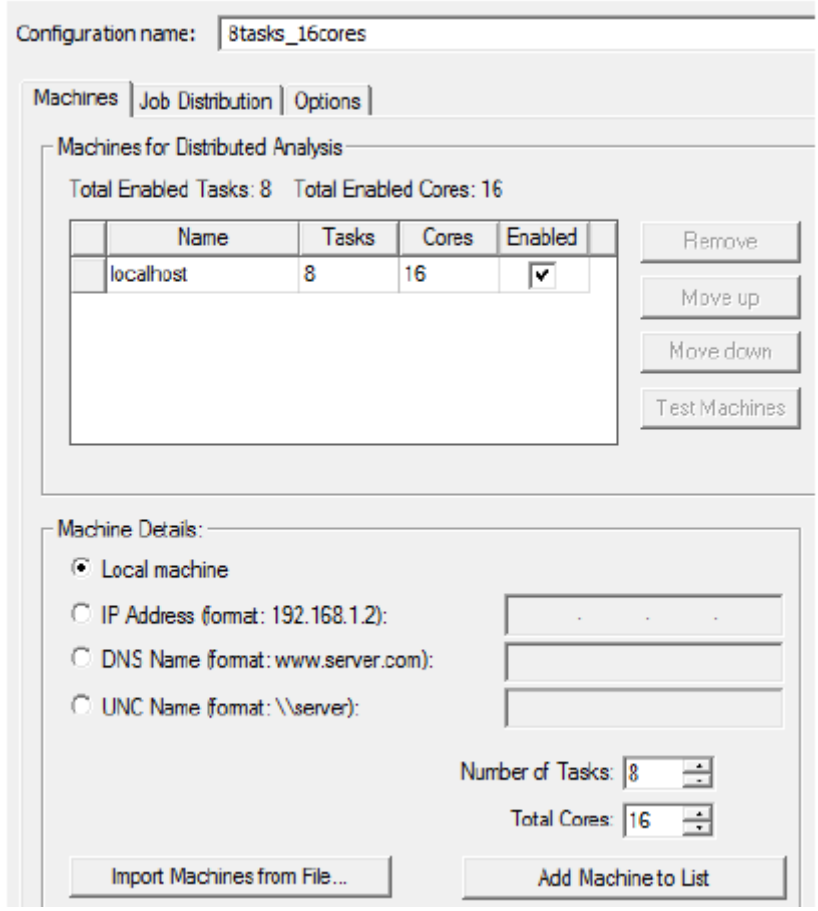
Negative values of deembedding indicates that the wave ports are deembedded away from the structure to stretch them to the required lengths. The transmission line characteristics are calculated along the shifted reference plane due to the deembedding. Deembedding prevents explicit drawing of the entire cable lengths. The following figure shows the 4 terminals assigned in this design.



Twinaxial Cable terminals

HPC and Solution Setups

Run the design at an adapt frequency of 20 GHz. Since a parametric sweep is used (look under **Optimetrics**), this design is a good choice to employ HPC. You can access HPC settings from **Tools > Options > HPC and Analysis Options**. Click the **Add** button to open the **Analysis and Configuration** window where you can set the number of cores and tasks for the HPC simulation.



HPC analysis and configuration options

For example in figure above, 16 cores are available on the machine in which the design was simulated. In a set up of 8 tasks executed with these 16 cores, the sweep is run with 8 frequency points being solved in parallel using two cores of matrix multiprocessing for each frequency point. When such an analysis is executed on a single machine, the simulation is very efficient if the machine has enough shared memory to accommodate 8 simultaneous solves. Otherwise the analysis could be

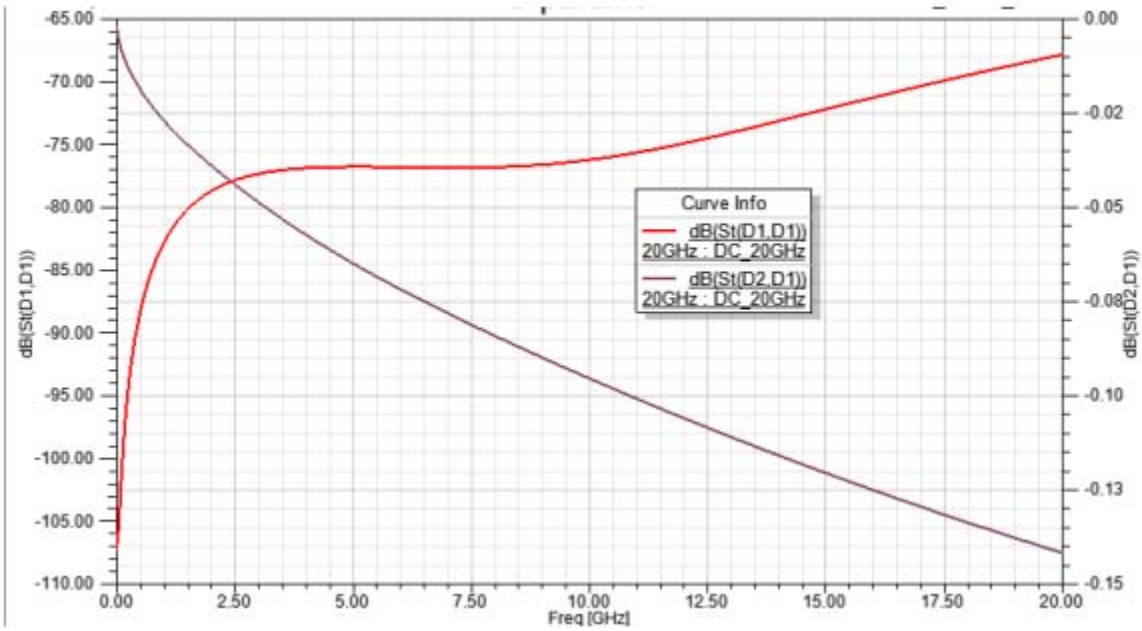
performed across multiple machines (that have HFSS installed in them) without requiring any additional HFSS license for each machine.

Note For more information about HPC, see the sections **Setting HPC and Analysis Options** and **Editing Distributed Machine Configurations** in the online help.

The advantage of deembedding is that it saves the trouble of explicitly modeling the long cable lengths. Such a design is efficient and can be solved using minimum computational resources. By using HPC, the solution time is further reduced.

Results

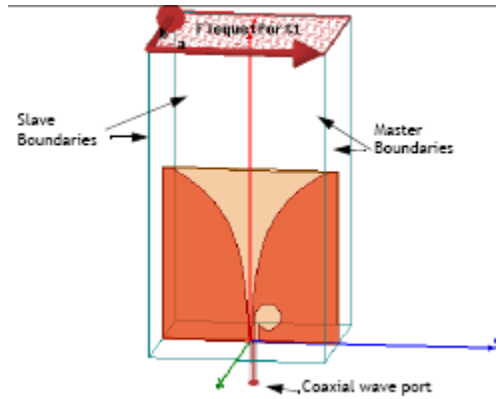
The following figure shows the S-parameter plot.



s-parameter plot

Unit Cell of a Phased Array

Description - A unit cell of an infinite phased array of vivaldi antennas is simulated using linked boundaries and a Floquet port.



Model - the antenna is fed by a coax line with a wave port. The upper face of the unit cell is terminated in a Floquet port. The sides are two pairs of Master and Slave boundaries. The substrate ε_r = 6 and is 1.27 mm thick. The conducting traces are 2D objects with PerE boundaries.

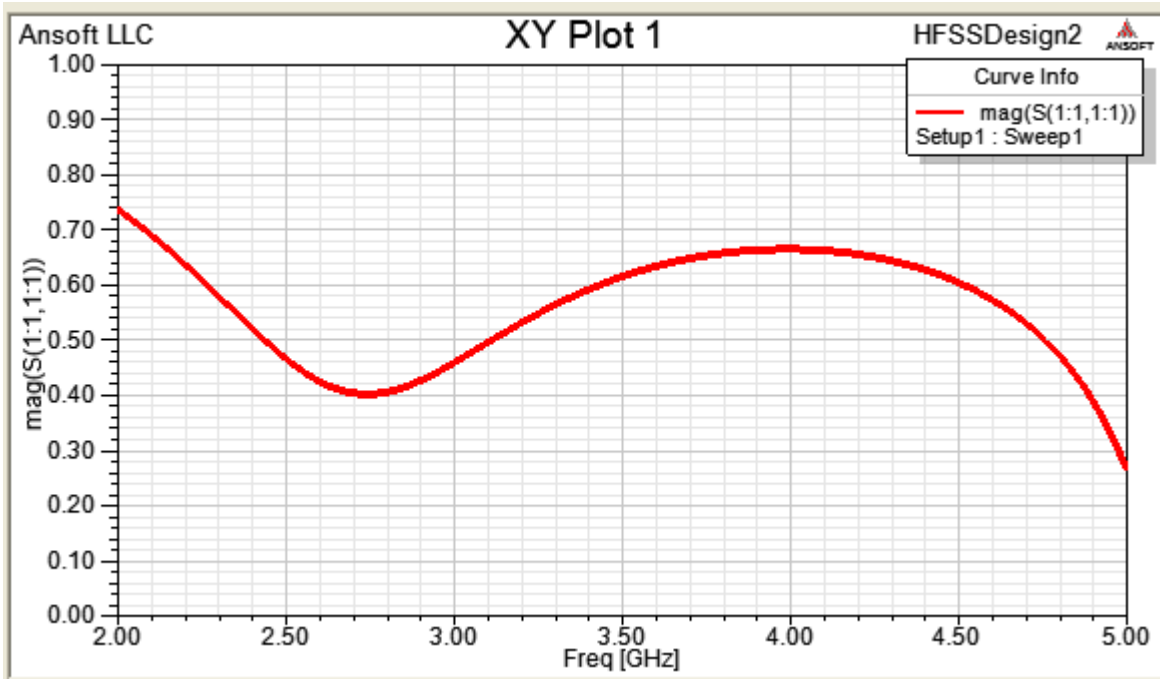
Setup - Adapt at 4.5 GHz with an interpolating sweep from 2 to 5 GHz.

Note To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed in the Properties window.

Post Processing

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

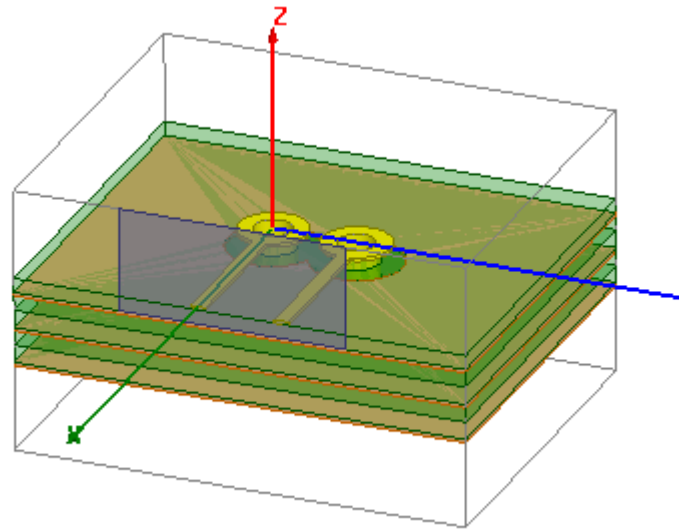
To view a plot of active S parameter seen at the feed, look in the **Project** tree and double-click on XY Plot1. To view the modes present on the Floquet port click on the desired mode under **Port Field Display>Floquet Port 1** in the **Project** tree, and a vector plot of the mode will be displayed.



This design was analyzed in "Analysis of Periodic Structures via a Time-Domain Finite-Element Formulation with a Floquet ABC," L.E.R. Peterson et al., IEEE Trans, AP, March 2006, pp 933-944. You will see the plot computed here agrees nicely with Fig. 9b in the reference.

Via Model

Description - a model of a differential via pair. It has a pair of microstrip lines that transition through the vias to a pair of striplines on a lower layer. This model was created using the Via Wizard.*



Model - the two microstrip lines are each assigned a terminal in the coupled microstrip port. Likewise for the two striplines at the opposite end. The conductors are copper and a radiation boundary is applied to the air box.

Setup - adapt at 4.38 GHz with an interpolating sweep that has an upper frequency of 4.38 GHz and uses DC extrapolation at the lower end. Mixed Order is set for Order of Basis Function.

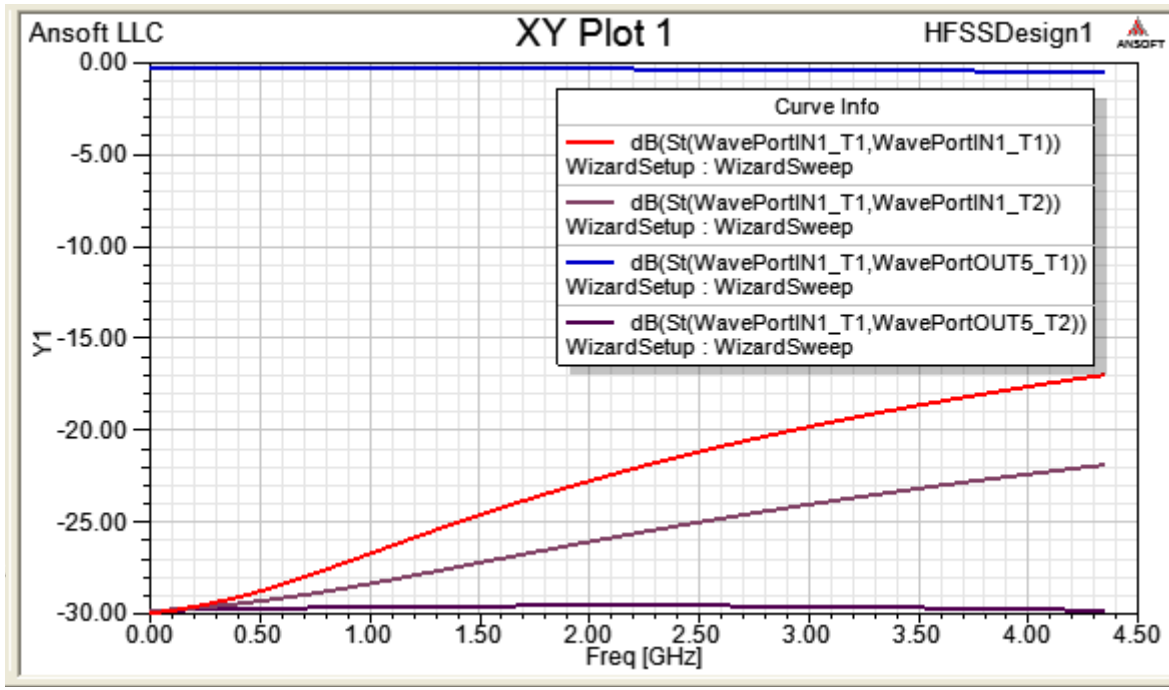
Note To view a port or boundary, select the desired item in the Project Tree. It is then highlighted in the Model window and the properties will be displayed in the Properties window. Selecting an object in the History tree will also display its properties.

* To download the Via Wizard, see www.ansoft.com/3dviadesign.

Via Model Postprocessing

After solving, you can view solution data by right-clicking on Setup1 and selecting **Profile** to display the **Solution** dialog. You also view the **Solution** tabs for **Convergence**, **Matrix Data**, and **Mesh Statistics**.

To view the S parameter plot shown below, double click on XY plot1 in the Project Tree under Results.



Example Toolkits

The Toolkit command in the HFSS menu provides access to a design type specific IronPython script, such as the Hearing Aid Compliance Test, MIMO calculation or a module specific task, such as for Boundary, Excitation, or Solve Setup.

- The Hearing Aid Compliance Test toolkit example involves multiple modules.
- The MIMO example includes a Toolkit script and a User Defined Solution Script.
- The Cable modeling toolkits serve automotive cable applications and oil and gas applications.

Hearing Aid Compliance Test

MIMO Example

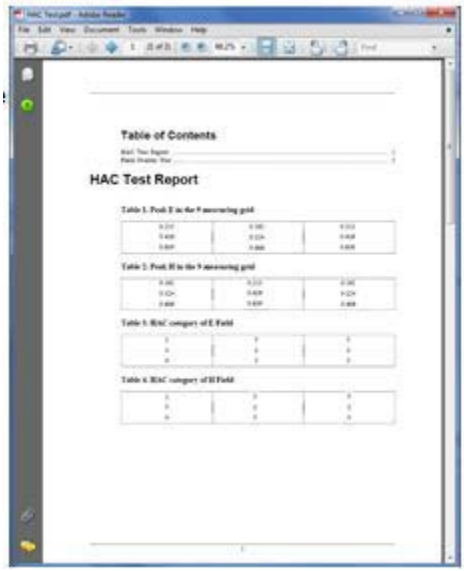
Cable Modeling Automotive

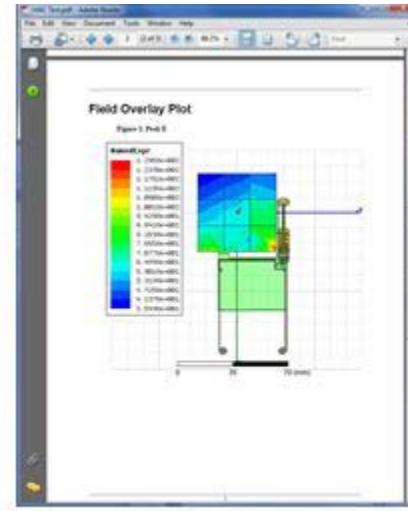
Cable Modeling Oil & Gas

Hearing Aid Compliance Test

Using a toolkit approach, you can test and report the hearing aid compatibility of your designs in HFSS. This involves an easy access to a python script that provides custom GUIs to setup the non-model measurement plane and sweep setup that satisfy the ANSI C63.19 standard. Various report-

ing mechanisms are created automatically. Reporting mechanisms include field overlay plot and UDD (User Defined Document).





The python script is installed with the software. The script assumes that you have already created the phone model and the appropriate relative coordinate system that defines the location and orientation of the ear piece. You must also assign appropriate boundaries/excitations and any other required design data, such as solution setup.

The python script provides default but customizable UI and canned automations. The UDD format can be adjusted/extended as needed. The default UI is a modal dialog.

This allows us to support changing standards, or multiple standards, without necessarily being tied to the product release cycle. Similarly, users can create new UDD to customize the test report. UDD are saved in project result directory. This type of reports can be created prior to simulation, but will only be populated when there is solution.

Once the measurement plane, sweep setup and various reports are created by the Toolkit script, they become part of the design and users will be able to make modifications. However, that might cause the HAC test to fail compliance with the standard.

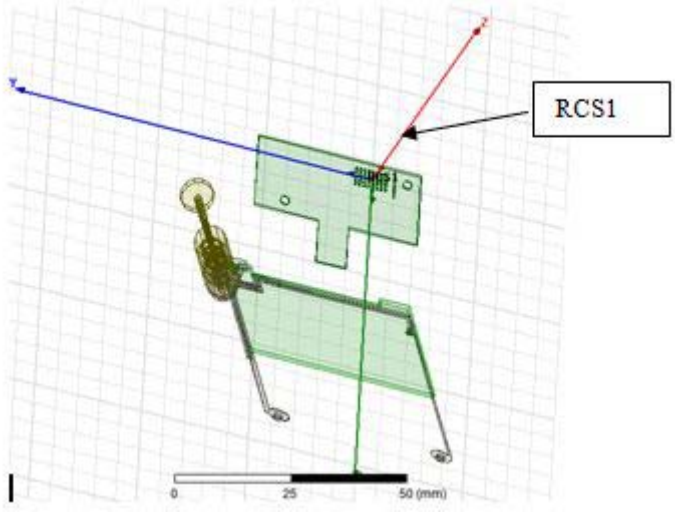
The measurement plane will be parameterized based on the variables used in the RCS definition of the ear piece. The measurement plane is a non-model object and the automated UI operations executed by the python script will not invalidate previously simulated results. This means that users can add the HAC analysis at any stage of their design process.

Related Topics

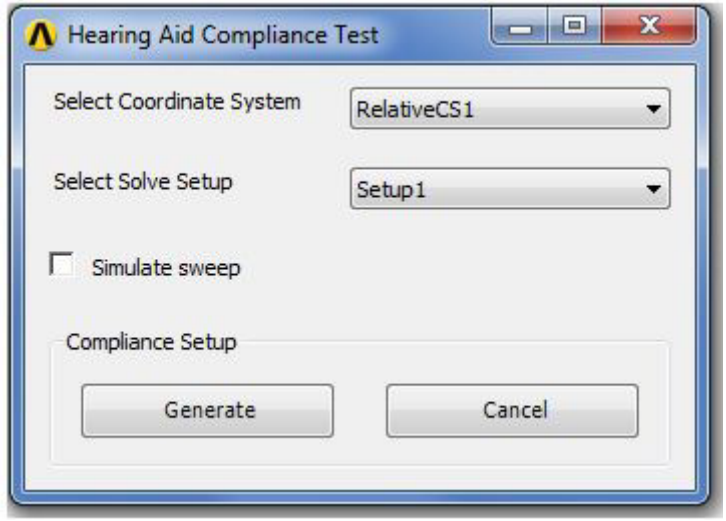
Using the Hearing Aid Compliance Test

Using the Hearing Aid Compliance Test

1. Insert phone model and create the relative coordinate system that defines the location and orientation of the ear piece. The name of the relative coordinate system for the earpiece is predefined.



2. Click **HFSS>Toolkits>HearingAidCompliance** to bring up the "Hearing Aid Compliance Test" dialog.

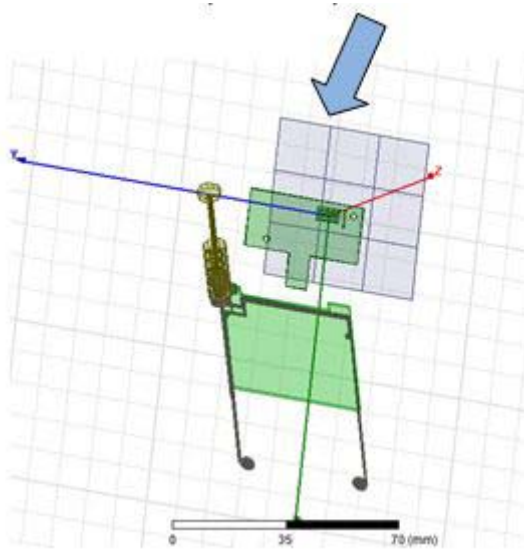


The first combo box is populated with user defined relative coordinate systems and the second is populated with the solve setup in the active design. A sweep will be created within the selected solve setup.

Users can choose to include simulating the created sweep as part of the automation.

3. Click Generate.

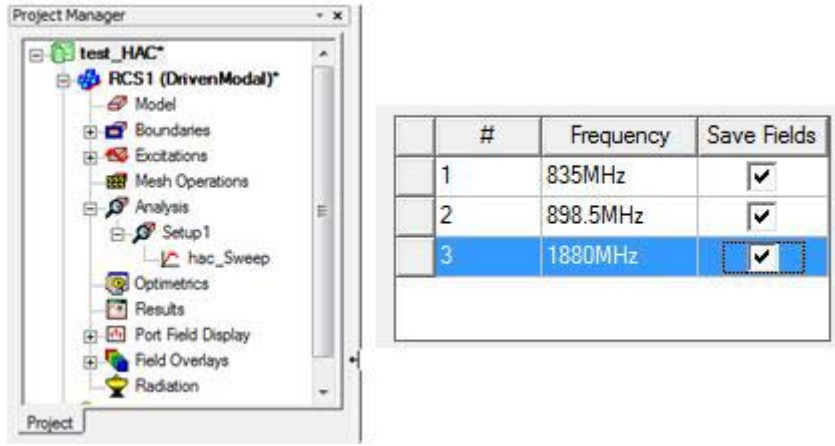
   A measurement plane composed of 9 non-model rectangles are inserted into the design.



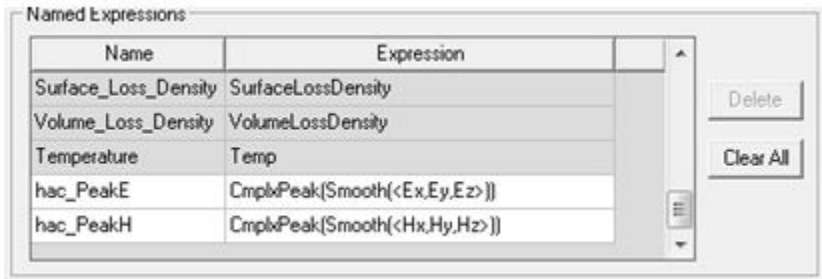
   These 9 rectangles will be named hac_grid1_1, hac _grid1_2, hac _grid1_3, hac _grid2_1, … hac _grid3_3. If existing objects in the design that conflict with any of these 9 names, the python script searches for the next set of names to use, in the format of hac1_grid1_1, hac1 _grid1_2, hac1 _grid1_3, hac1 _grid2_1, … hac1 _grid3_3, so on and so forth.

4. A discrete sweep is created for the selected solve setup. The frequencies required in the ANSI standard will be included in the sweep. You can select the testing frequencies. This sweep will be named "HAC_Sweep". When there is already a sweep of the same name, then the next

name to use is "HAC1_Sweep", "HAC2_Sweep", so on and so forth.



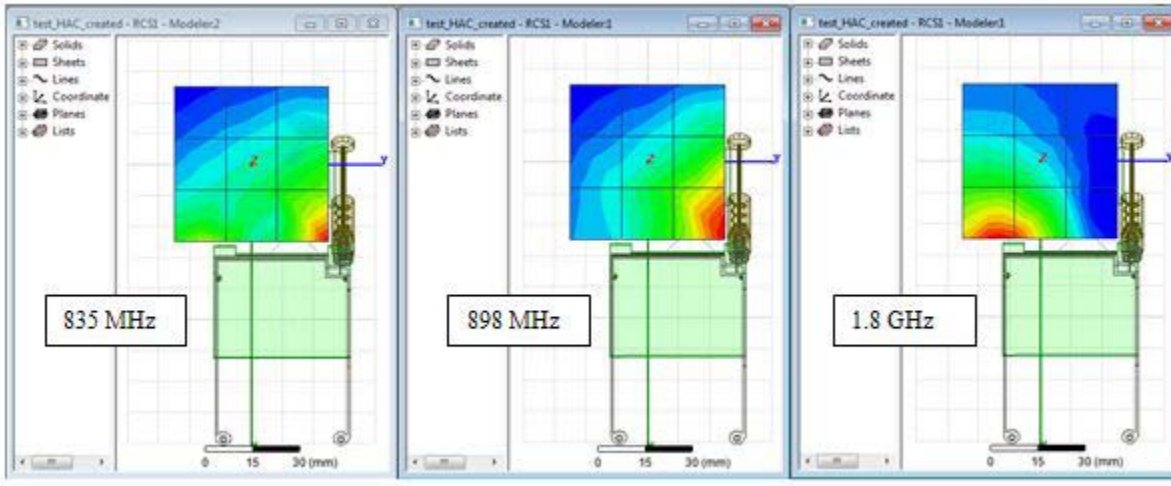
5. The following field calculator expressions are created.



These field calculator expressions will be available in reporter and Optimetrics and can be plotted/evaluated on any user defined polyline.

6. PeakE and PeakH Field overlay are plotted on the surface of the measurement plane, using the

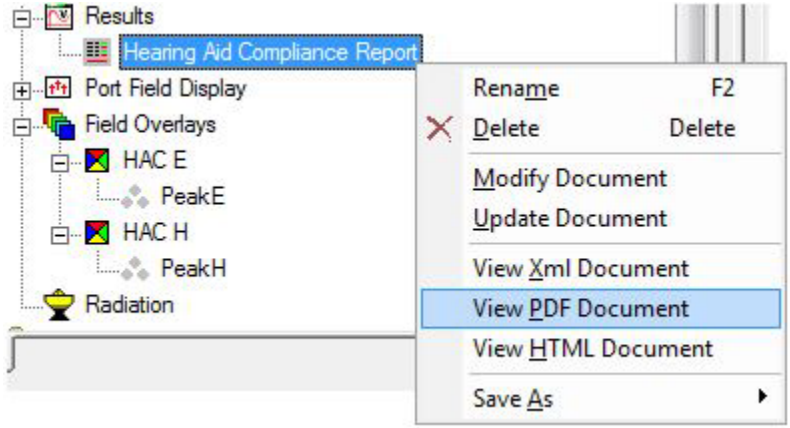
sweep solution and at each of the frequency.



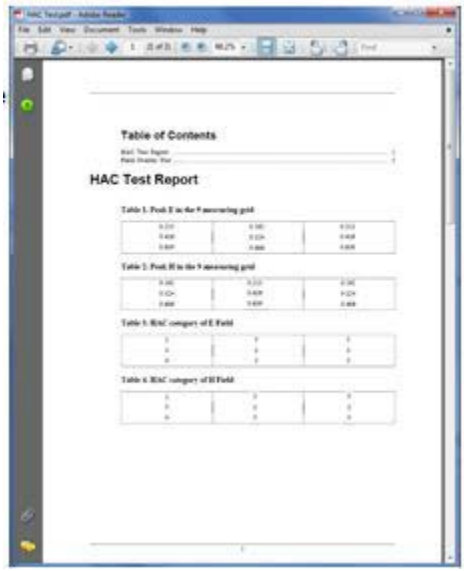
If you did not select the correct coordinate system to generate the measurement grid, and therefore, no solution to populate the plot, there will be an error message.

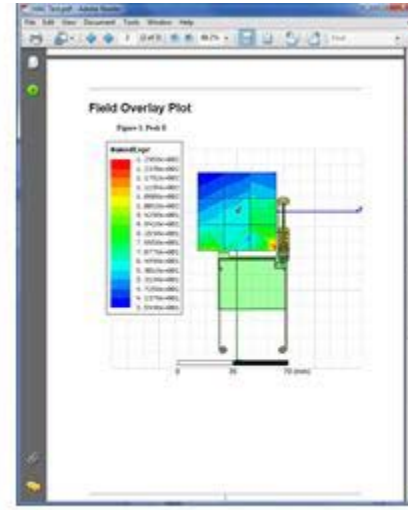
7. If you select **Simulate Sweep**, a User Defined Document will be created and inserted into the Results folder.

Right click and select "View PDF Document" to display the pdf.



The Hearing Aid Compliance Report opens:

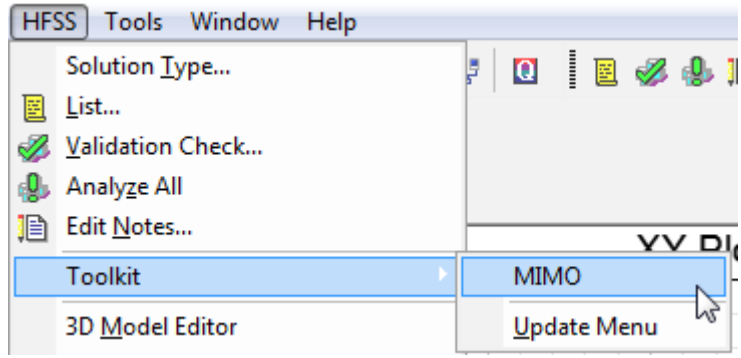




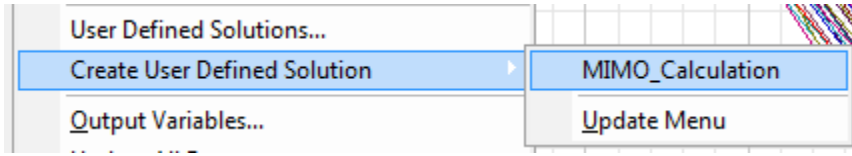
User Defined Solution for MIMO Calculations

HFSS includes the means to use python scripts for developing Toolkits for User Defined Solutions and User Defined Outputs for specific applications. The MIMO.py script in the syslib>Toolskits directory and the MIMO_Calculation.py script in the syslib>UserDefinedOutputs directory illustrate this technique as applied to an important design application. This toolkit does not work with Linux.

You will need to click **HFSS>Toolkit>Update Menu** to cause the **Toolkit** menu to display the MIMO selection.



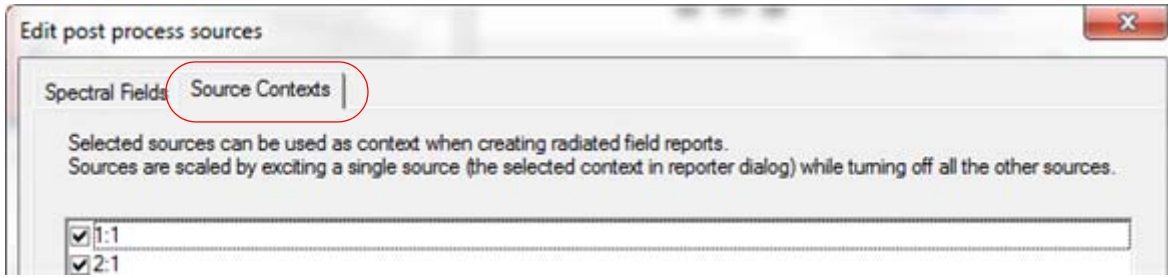
You also click **Results>Create User Defined Solution>Update Menu** to display the MIMO_Calculation selection.



The use of Multiple Input Multiple Output (MIMO) design in both transmitters and receivers has become an important technique applied towards improved communication. The MIMO calculations also illustrate the process that applies towards generating User Defined Reports. In this case, the Toolkit and User Defined Solutions help you obtain reports of MIMO Calculations of the Envelope Correlation Coefficient and Diversity Gain. You can also use these MIMO quantities in Optimetrics. You can use these scripts with any HFSS Modal or Terminal Network Analysis design with at least two inputs.

Once you have selected a design and have (if necessary) activated the scripts (Update Menu), the MIMO process includes the following steps.

1. In the **Edit Sources** dialog, use the **Source Contexts** tab to specify the sources to use as context when generating radiated fields.



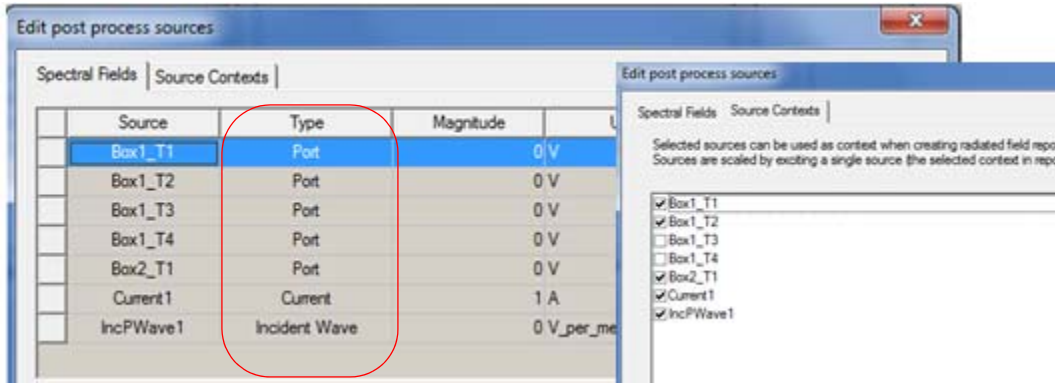
The **Source Contexts** tab is only available in HFSS Driven Modal and Terminal Network Analysis solution types.

It is not available in

- Composite Excitation sub-solution type
- Design with array setup
- Eigen and Transient solution type
- HFSS-IE design type

To run the MIMO calculations example, you need a driven Modal or Terminal Network Analysis design with at least two sources. The default for all sources is unchecked (disabled).

All types of sources can be enabled as source context.



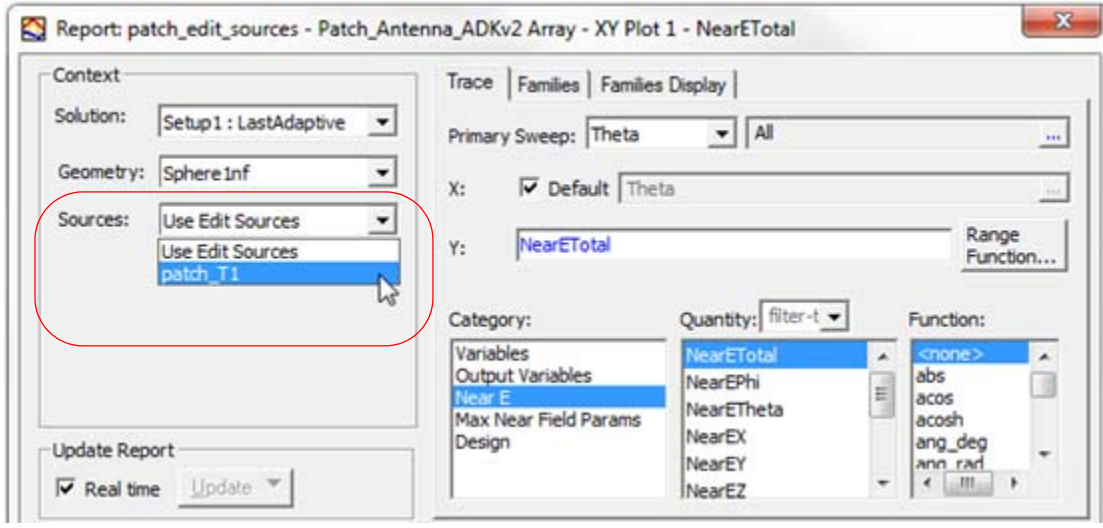
Each singly selected source is excited with factory default values. The phase will always be 0 deg and the magnitude will be 1 Watt for modal design and 1 Volt (Total voltage) for terminal design. These factory default values cannot be changed by users.

Design **Edit Source** settings such as "Include Port Post Processing Effects" will still be applicable for these singly excited sources. Thus changing any setups on the **Spectral Fields** tab of the **Edit Source** dialog will invalidate all traces that are defined with source context.

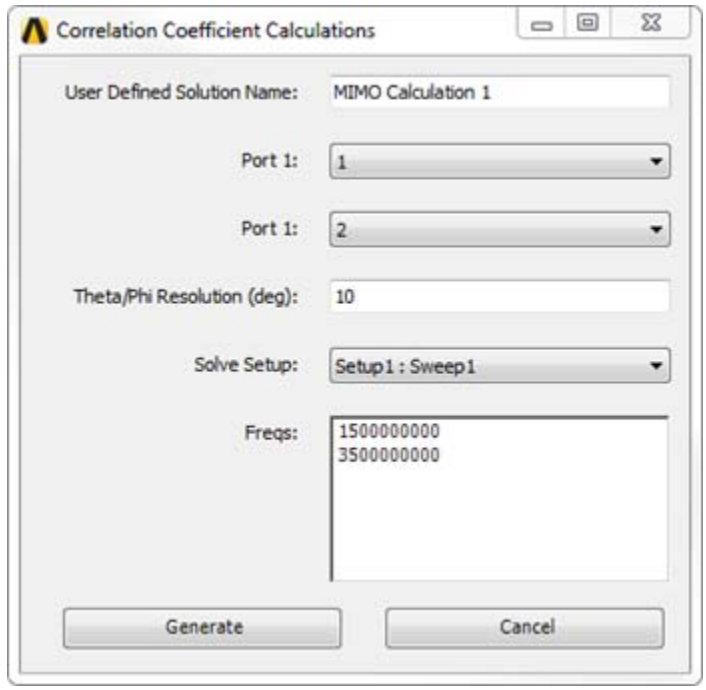
Enabling any source as context will not impact traces that are already defined

Disabling source as context will invalidate an existing trace that uses that source as context

2. You can scale sources by exciting a single source while using the Reporter. A Sources combo box appears in the Reporter when you have specified at least one source in the **Edit Sources** dialog.



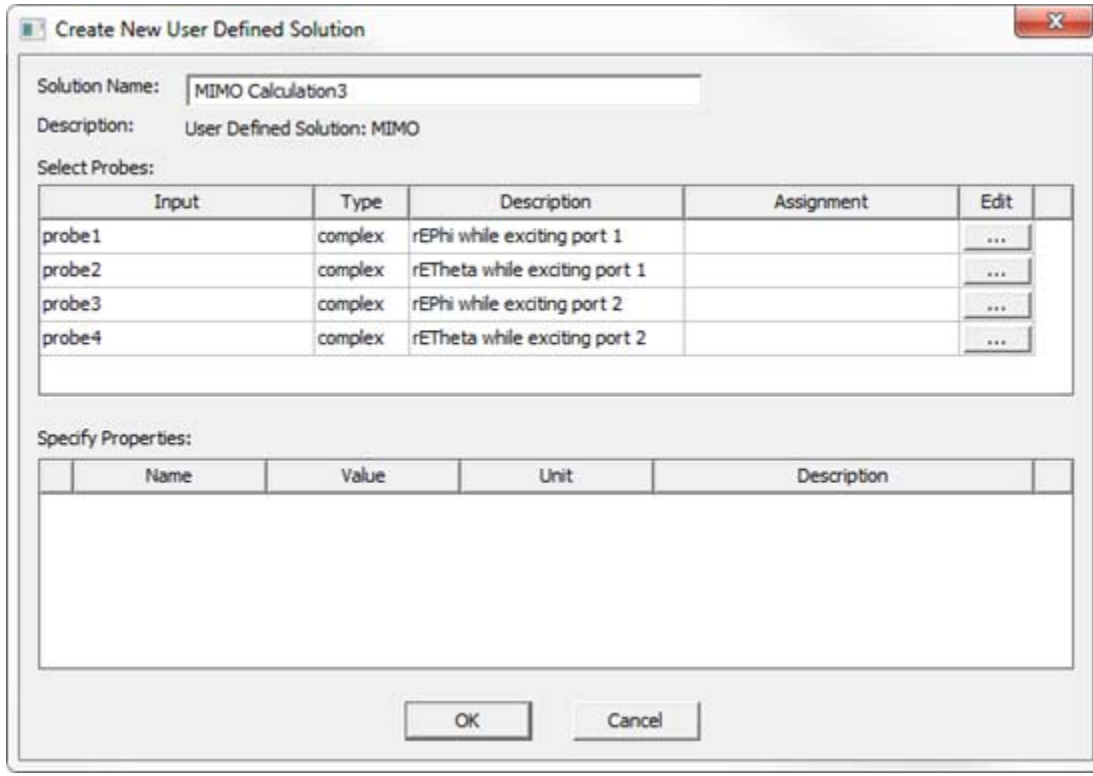
3. Select **HFSS>Toolkit>MIMO** to view the **Correlation Coefficient Calculations** dialog.



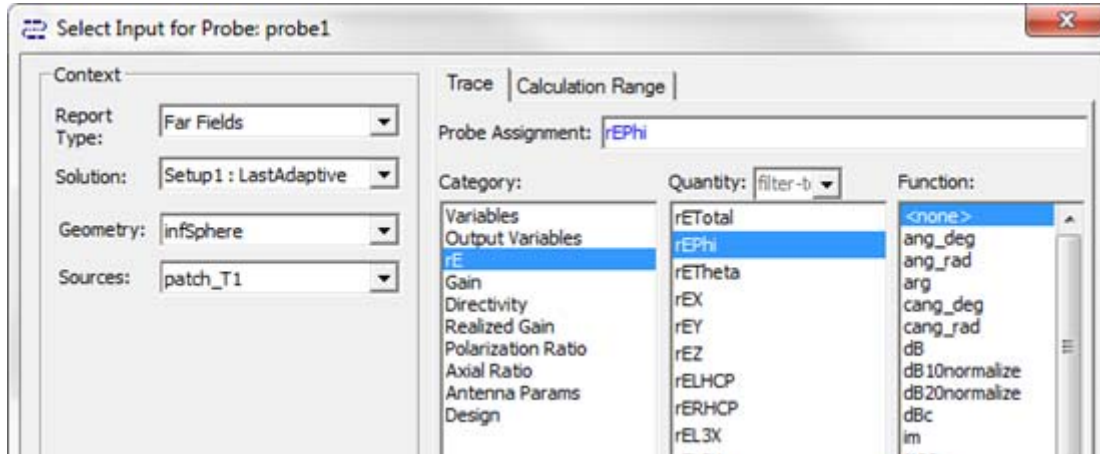
From here you can select Ports, specify the Theta/Phi resolution in degrees, and select the Solve Setup. Click Generate to apply the settings.

4. Select **Results>Create User Defined Solution** to select Input probes in the **Editing User**

Defined Solution dialogue for the MIMO calculation.

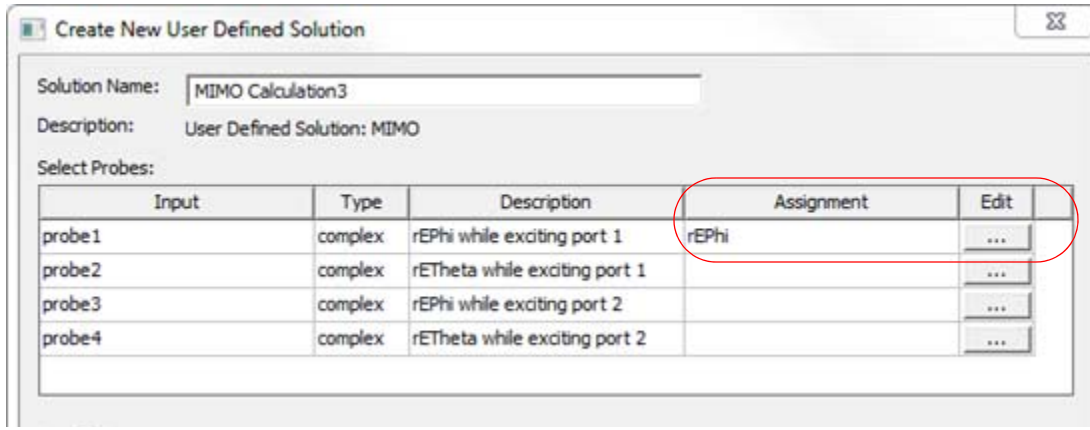


From here, you can also make assignments for the sources. Click the ellipses [...] buttons in the Edit column to open a **Select Input** dialog for the respective probe.



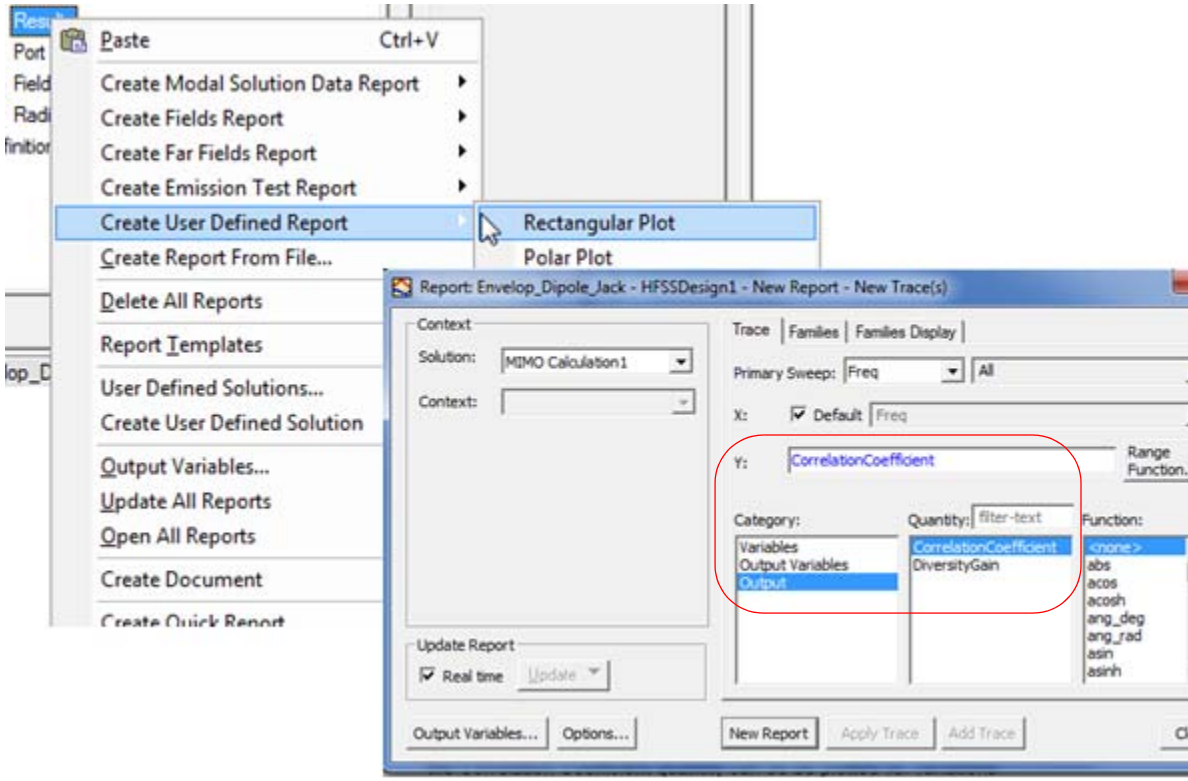
If you edit the Probe assignment here, and OK the dialog. You select Probe assignment category, quantity and function based on the Description in the User Defined Solution. The

Reporter Probe Assignment field restricts the legal values to the Description field. Red text denotes illegal assignments, and blue, legal. The assignment value for that probe appears in the **Create User Defined Solution** dialog.

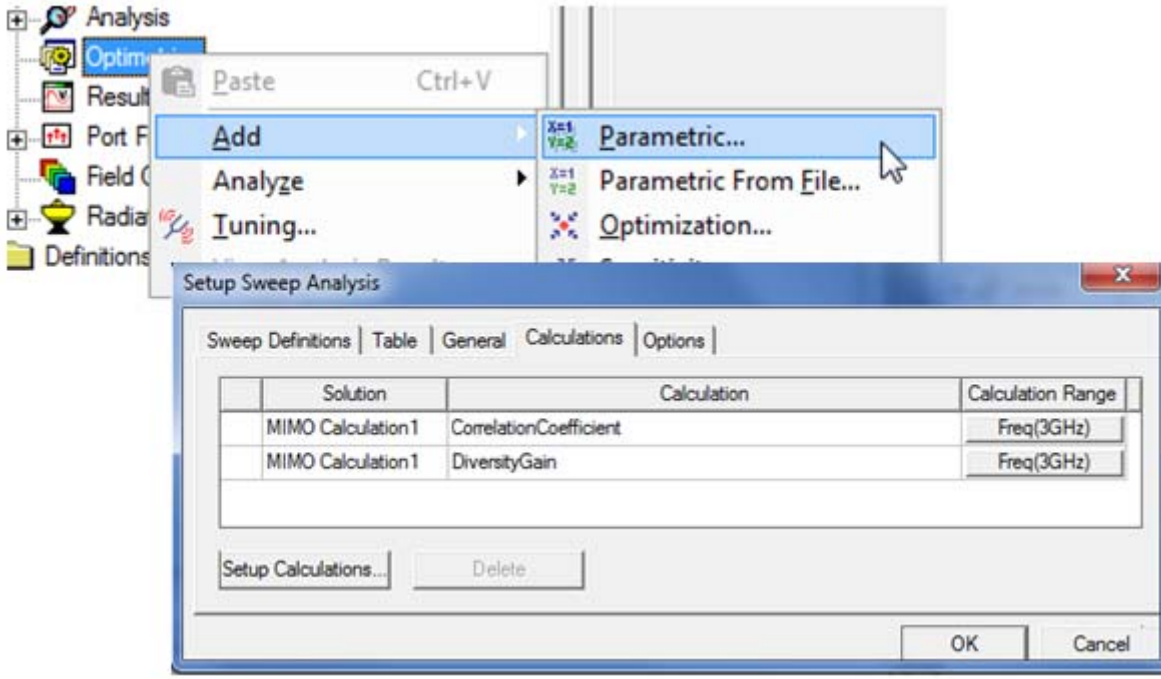


The created solution "MIMO Calculation1" has 2 output quantities, the "CorrelationCoefficient" (Envelope Correlation Coefficient) and "DiversityGain". These quantities can be used in both reporter and Optimetrics. For the Reporter, you can select Output as the Category to select

CorrelationCoefficent and Diversity Gain as Quantities.



For Optimetrics, the MIMO calculations appear in the Sweep Definitions.



Cable Modeling - Oil and Gas Toolkit

To work with tool kits in the oil and gas industries:

1. Click **HFSS>Toolkit>CableModeling>Oil-GasCableBundle**.

   The **Cable Modeling - Oil & Gas** dialog box appears.
2. Type the **Name** of the toolkit, and select the units.
3. If you have previously imported a customized set of parameters, you can click **Use Defaults** to use the default values.
4. To export the cross-section parameters into a .csv or .tab file, click **Export**. The **Export Cable Parameters** dialog box appears. Specify the **File name** and **type**, and click **Save**. The file is exported and can be seen in Excel.
   a. The parameter name is listed in the **ParameterName** field.
   b. The next column displays the **Value** of the parameters. If you want to create a "menu" type of parameter to give more than one choice of values, separate the values with semi-colons. The first string in the menu options must be the current value of the parameter.
   c. The next column shows the **Section** parameters related to different sections of cable.
   d. The next column shows the **PropType**.
5. To read in the excel file and populate parameters in the toolkit panel, click **Import**. The

Import Cable Parameters dialog box appears. Specify the **File name** and **type**, and click **Open**.

6. To add new variables, click **Variables**. This opens the **Edit Variables** dialog box. All variables that are already present in the project and design levels are listed. Click **Add** to add a new row to create new variables.
7. Click **Draw** to draw the 2D cross-section geometry. A validation check is run. This catches errors like any variable missing a value, or any parameter missing a value.
8. For 3D designs, you can create a 3D geometry. Click **Extrude**. The **Extrude to 3D** dialog box appears.
   - Click **Sweep Along Vector** to extrude along the z axis. Z length is the input parameter.
   - Click **Sweep Along Path** to extrude along a selected path.

Note 3D models of cables are hard to solve if the ratio of extruded length to cross-section diameter is high.

Cable Modeling - Automotive Toolkit

To work with tool kits in the automotive industries:

1. Click **HFSS>Toolkit>CableModeling>AutomotiveCableBundle**.

   The **Cable Modeling - Automotive** dialog box appears with the **Cable Parameters** listed.
2. Modify the **Cable Parameters**:
   a. Select the wire **Standard**. You can choose between the **ISO** or **AWG** standard types.
   b. Select the **Wire Type**.
   c. Optionally, modify the number of wires.
   d. Optionally, modify the **Conductor Diameter**.
   e. Select the **Conductor Material**.
   f. Select the **Insulation Type**.
   g. Optionally, modify the **Insulation Thickness**.
   h. Optionally, modify the **Insulation Material**.
   i. To add additional wire types, click the **+** button. A new row is added. You can now create a bundle consisting of several different types of wires.
3. Type the **Name** of the toolkit, and select the units.
4. If you have previously imported a customized set of parameters, you can click **Use Defaults** to use the default values.
5. To export the cross section parameters into a .csv or .tab file, click **Export**. The **Export Cable Parameters** dialog box appears. Specify the **File name** and **type**, and click **Save**. The file is exported and can be seen in Excel.
6. To read in the excel file and populate parameters in the toolkit panel, click **Import**. The **Import Cable Parameters** dialog box appears. Specify the **File name** and **type**, then and **Open**.

7. Specify the **Outer Jacket Parameters**:
   a. Specify the **Thickness** value for the inner diameter.
   b. Specify the **Material** for outer jacket
   c. After specifying wires to be included in the bundle, click **Compute** to calculate the minimal bundle diameter of the bundle.

Note The wire radii are adjusted downward very slightly (on the order of 0.1%) at the end of the packing process to eliminate any residual overlaps between the wires and to make the model easier to mesh.

Note The Circle Packing algorithm is used to compute the values. It tries to automatically pack the conductors in the bundle tightly with a minimum of empty space between them. It generates and tests a large number of random variations in the conductor positions and attempts to minimize the overall diameter of the bundle.

   d. Specify the **Seeding for wire arrangement** value. The value entered here is the seed value for the pseudo-random number generator used in the conductor packing process. Changing this value results in different arrangements of the conductors. This can be useful for performing statistical analysis of the cables.
8. To add new variables, click **Variables**. This opens the **Edit Variables** dialog box. All variables that are already present in the project and design levels are listed. Click **Add** to add a new row to create new variables.
9. Click **Draw** to draw the 2D cross-section geometry. A validation check is run. This catches errors like any variable missing a value, or any parameter missing a value.
10. For 3D designs, you can create a 3D geometry. Click **Extrude**. The **Extrude to 3D** dialog box appears.
    - Click **Sweep Along Vector** to extrude along the z axis. Z length is the input parameter.
    - Click **Sweep Along Path** to extrude along a selected path.

Note 3D models of cables are hard to solve if the ratio of extruded length to cross-section diameter is high.

2

Getting Help

ANSYS Technical Support

To contact ANSYS technical support staff in your geographical area, please log on to the ANSYS corporate website, http://www.ansys.com/Support. You can also contact your ANSYS EM account manager in order to obtain this information.

E-mail can work well for technical support. All ANSYS EM software files are ASCII text and can be sent conveniently by e-mail. When reporting difficulties, it is extremely helpful to include very specific information about what steps were taken or what stages the simulation reached. This allows more rapid and effective debugging.

Help Menu Commands

To access online help from the menu bar, do the following:

- Click **Help>Contents**
- Click **Help>Index**
- Click **Help>Search**

You can also access help for the scripting commands via the menu bar:

- Click **Help>Scripting Contents**
- Click **Help>Scripting Index**
- Click **Help>Search Scripting**

Context-Sensitive Help

To access online help from the HFSS user interface, do one of the following:

- To open a help topic about an HFSS menu command, press **Shift**+**F1** or click  and then click the command or toolbar icon.
- To open a help topic about an HFSS dialog box, open the dialog box, and then press **F1**.

PDF of Online Help for Printing

The pdf of online help provides a format and function better suited for printing than the chm. You can print ranges of pages encompassing topics and subtopics as needed.

Open the PDF:

.

Conventions Used in the Online Help

The following documentation conventions are used in the online help.

- Procedures are presented as numbered lists. A single bullet indicates that the procedure has only one step.
- Bold type is used for the following:

  - Keyboard entries that should be typed in their entirety exactly as shown. For example, "**copy file1**" means to type the word **copy**, to type a space, and then to type **file1**.

  - On-screen prompts and messages, names of options and text boxes, and menu commands. Menu commands are often separated by carats. For example, click **File>Open**.

  - Labeled keys on the computer keyboard. For example, "Press **Return**" means to press the key labeled **Return**.
- Italic type is used for the following:

  - Emphasis.

  - The titles of publications.

  - Keyboard entries when a name or a variable must be typed in place of the words in italics. For example, "**copy** *file name*" means to type the word **copy**, to type a space, and then to type a file name.
- The plus sign (+) is used between keyboard keys to indicate that you should press the keys at the same time. For example, "Press **Shift+F1**" means to press the **Shift** key and the **F1** key at the same time.
- Toolbar buttons serve as shortcuts for executing commands. Toolbar buttons are displayed after the command they execute. For example,

  "Click **Draw**> **Line**  " means that you can click the Draw Line toolbar button to execute the **Line** command.

Searching in Help

The online help system provides four ways to search for information and navigate quickly:

- A hierarchical table of contents - you can expand or collapse the hierarchy by clicking, and you can jump to selected entries by double-clicking.
- A searchable index - you can search for indexed terms by typing the text field, and jump to topic locations by double-clicking on them.
- A full text search - you can type text, and search the entire online help. Items are listed according to rank in discussing the search text.
- A favorites list - you can select topics that you use frequently to create a favorites list.

Using WebUpdate

To use WebUpdate:

1. Select **Help>Launch WebUpdate**.

   This displays the **WebUpdate** dialog, which lists the applications available for update.

2. Select the application of interest and click **Next**.

   This displays the application and whether it is currently up to date and whether an update is available

3. If an update is available, enable the application checkbox to select it.
   a. You can choose to enable the checkboxes to install the update automatically and to save the update to disk.

      If you choose to update, the **Next** button is enabled.

   b. Click **Next** to continue the update.
   c. The Webupdate shows the progress of the update.
4. Click **Close** when done

Obtaining Information About the Software and Version

To obtain information about

1. Click **Help>About ANSYS HFSS**.
   The **About ANSYS HFSS** dialog box appears, listing information about the product.
2. Click the **Installed Components** tab to view a list of software installed.
3. Click the **Client License Settings** tab to view information about the following:
   - Active Host Ids
   - Active Servers
   - Admin Directory
   - Customer Number
   - FLEXlm Version
   - License Files
4. To export the software information:
   a. Click **Export**.
      The **Save As** dialog box appears.
   b. Browse to the location where you want to save the information as a text file.
   c. Type a name for the file in the **File name** text box. The **Save as type** pull-down list is already specified as **Export (*.txt)**.
   d. Click **Save**.
5. Click **OK** to close the **About ANSYS HFSS** dialog box.

What's New in this Release

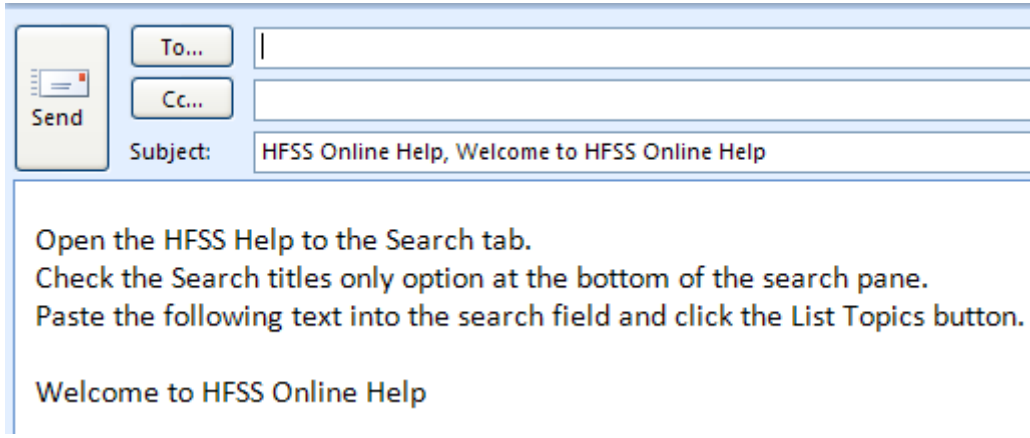
Click **Help>What's New in this Release** to open a pdf file describing the important features.

Emailing a Help Page Title

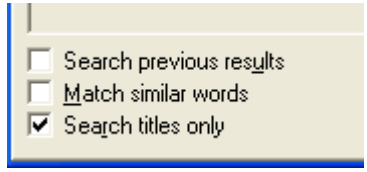
To email a help page title from the online help chm:

1. Go to a help page of interest in the online help.
2. Click the ANSYS logo on the lower left corner of the help.

3. The help launches an email.



4. Supply an email address and send the email.
5. Upon receiving the email, the recipient follows the directions by opening the online help, and selecting the Search pane.
6. Select Search titles only..



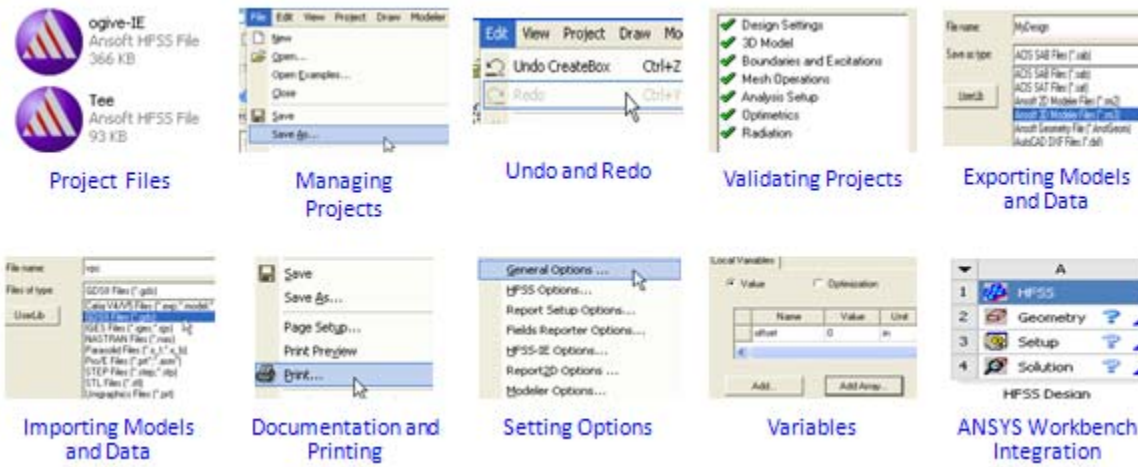
7. The recipient copies the section title from the email and pastes it into the search field.
8. The recipient clicks the List Topics button.

   This displays the topic. The recipient can click the topic to go directly to the help page.

3

Working with HFSS Projects

An HFSS or HFSS-IE project is a folder that includes one or more HFSS models, or *designs*. Each design ultimately includes a geometric model, its boundary conditions and material assignments, and field solution and post-processing information.



A new project called *Projectn* is automatically created when the software is launched. By option, a design named Design*n* is automatically created for a new project. You can also open a new project by clicking **File>New**. In general, use the **File** menu commands to manage projects. If you move or change the names of files without using these commands, the software may not be able to find information necessary to solve the model.

HFSS Files

When you create an HFSS project, it is given an *.hfss* file extension and stored in the directory you specify. Any files related to that project are also stored in that directory.

Some common HFSS file and folder types are listed below:

.hfss	HFSS project.
design_name.**hfssresults**	HFSS folder containing results data for a design. It resides in the project.hfssresults folder.
project_name.**hfssresults**	HFSS folder containing results data for a project.
project_name.**asol**	The .asol file contains the database of all solved variations and where the resulting data is stored in the *design*.hfssresults folder. This file is stored in the *project_name*.**hfssresults** folder.
.pjt	HFSS version 8.5 and earlier project.
.anfp	ANSYS PCB neutral file

Related Topics

Projects

Creating Projects

Saving Projects

Deleting Projects

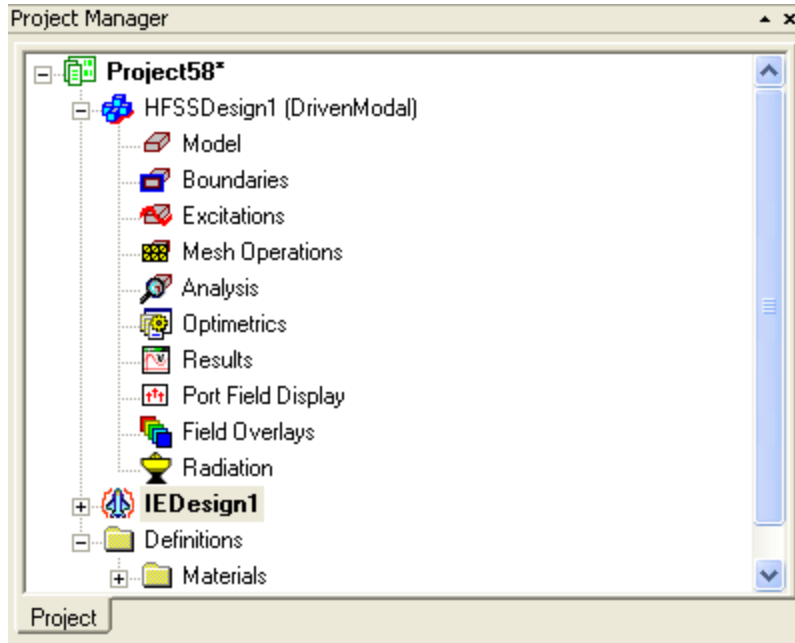
Managing Projects and Designs

Creating Projects

- Click **File**> **New**  .

  A new project is listed in the project tree. It is named Project*n* by default, where *n* is the order in which the project was added to the current project folder. A default design named Design*n* is added under the project.

  Project definitions, such boundary and material assignments, are stored under the project name in the project tree.



You specify the name of the project when you save it using the **File>Save** or **File>Save As** commands.

Related Topics

Projects

Creating Projects

Saving Projects

Deleting Projects

Managing Projects and Designs

Projects

Open a previously saved project using the **File>Open** command.

1. Click **File**> **Open**  .
2. Use the file browser to find the HFSS or HFSS-IE .hfss project file.

   By default, files that can be opened or translated by HFSS or HFSS-IE are displayed.
3. Select the file you want to open.
4. Click **OK**.

   The project information appears in the project tree.

   If you open another project without editing the automatically-created project, HFSS or HFSS-IE removes the automatically-created project.

You can also open a saved project by:

- Dragging an HFSS or HFSS-IE project file icon to the HFSS or HFSS-IE icon.
- Dragging an HFSS or HFSS-IE project file icon to the HFSS or HFSS-IE desktop.
- Double-clicking on an HFSS or HFSS-IE project file icon.

Related Topics

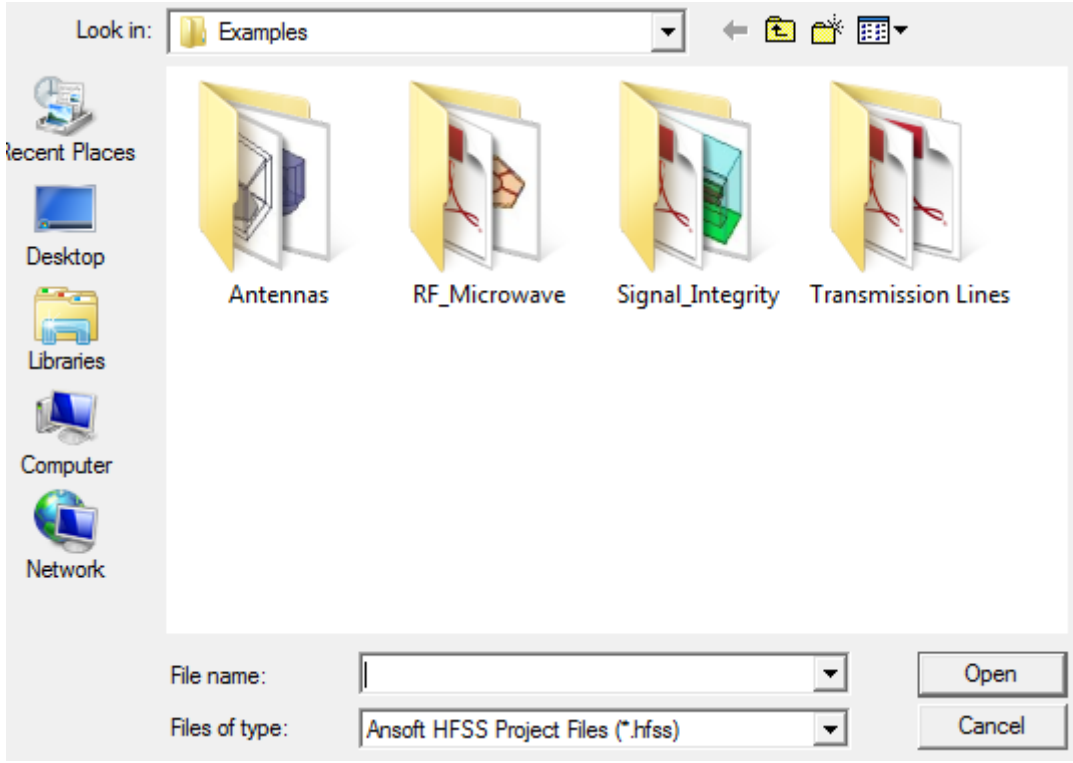
Opening Legacy HFSS Projects

Opening Example Projects

Updating Design Components

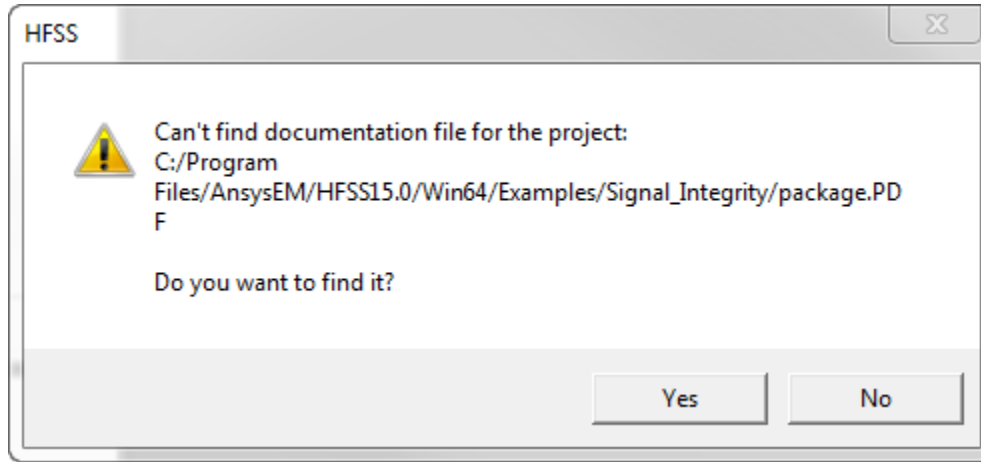
Opening Example Projects

You can directly access and open example projects included with product install by using **File>Open Examples**. This displays a browser open to the Examples folder in the product install. You can select from various display styles for the folders and projects.

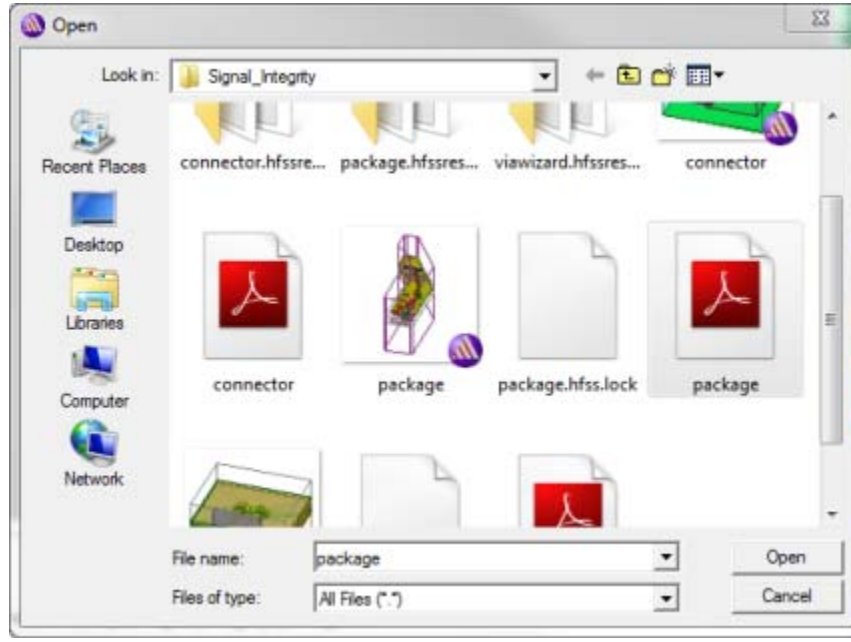


The online help contains additional descriptions of these projects.

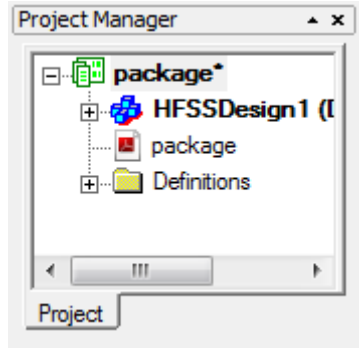
When you open an example project, you may see a dialog saying that the project can't find a documentation file for the project.



If you want to find the documentation file, click Yes to open a file browser You can look in the Project folder for the pdf file associated with the project..



When you add the documentation file to the project, you can see the icon in the Project tree.



Related Topics

Example Projects

Opening Recent Projects

To open a project you recently saved in HFSS or HFSS-IE:

- Click the name of the project file at the bottom of the **File** menu.

If you open another project without editing the automatically-created project, HFSS or HFSS-IE removes the automatically-created project.doc

Related Topics

Projects

Creating Projects

Saving Projects

Deleting Projects

Managing Projects and Designs

Opening Legacy HFSS Projects

HFSS 12 does not open projects created in Ansoft HFSS version 8.5 or earlier. HFSS 10 can be used. HFSS 10 files can be opened directly. However saving them in 12 means they cannot be used in 10.

1. Click **File**> **Open**  .
2. In the **Look in** pull-down list, click the location of the project. In the folder list, double-click folders to find the one that contains the project.
3. Double-click the project you want to open.

Legacy HFSS Project Translation

HFSS 12 translates all HFSS 10 data. It does not open projects created in Ansoft HFSS version 8.5 or earlier. HFSS 10 can be used to translate earlier projects. Virtually all of the project's pre-pro-

cessing data is translated. Note that solution results and Optimetrics setup data are unavailable; however, the nominal model created for Optimetrics is translated. Following are additional notes about the translation of various legacy project information.

Model Geometry

- The translated geometry's construction history is unavailable; therefore the original object properties you defined cannot be modified in the **Properties** window. However, you can modify the geometry using version 10's modeling features.
- For units unavailable in version 10, such as yards, the nearest available units are used; the model will be scaled slightly to fit the new units.
- View visualization settings apply to the saved design. If these have been changed from the default (15 deg), this affects the memory and CPU required to open the project.

Excitations and Boundaries

- Port impedance and calibration lines become integration lines in version 10. If the legacy project contained both impedance and calibration lines, impedance lines are translated and calibration lines are ignored. If the project contained both impedance and terminal lines, both are translated. The impedance lines will be ignored for Driven Terminal solutions and terminal lines will be ignored if the project is changed to a Driven Modal solution.
- Boundaries assigned to named interface selections or rectangle selections are not translated.
- For a boundary assigned to the intersection of two faces, HFSS 10 will create a new 2D sheet object from the intersecting area and assign the boundary to that object.

Materials

- Functions defined in legacy projects become project variables in version 10; therefore, functional material properties are translated.
- Perfect conductors become regular materials with conductivity values of 1E30.
- Object coordinate systems are created for objects assigned anisotropic materials in legacy projects. The coordinate system is defined at the same origin as the global coordinate system, with the same orientation defined when the anisotropic material was assigned to the object in the legacy project.
- Nonlinear materials from legacy projects that have magnetic saturation values greater than zero are treated as ferrite materials in version 10. Their properties are not modified.

Mesh Operations	• Mesh refinement operations performed on arbitrary boxes in legacy projects are ignored. • Area- and volume-based mesh operations are translated as length-based mesh operations in version 10 by taking their square roots and cube roots, respectively.
Optimetrics	• Setup information, including design variables, is not supported; however, the nominal model can be translated. • Parameterizing a translated model is limited because geometry construction history is unavailable.
Solution Types	• Driven solver projects that contained terminal lines are translated to Driven Terminal solution types in version 10.
Solution Setup	• Impedance-only and emissions-only solutions are not supported in version 10; therefore these selections in legacy projects are ignored. • The design's initial mesh is used for the version 10 solution. Current meshes are not translated. • Saving dominant-only or higher-order-only mode S-matrix entries are not supported in version 10; therefore these mode selections in legacy projects are ignored. • For frequency sweeps, the **Number of Steps** value specified in the legacy project is converted to the corresponding **Step Size** value in version 10. • The total number of requested adaptive passes in the legacy project becomes the **Maximum Number of Passes** value in version 10. For example, if you request 3 adaptive passes, solve them, and then request 2 adaptive passes, 5 will be the value specified for the **Maximum Number of Passes** in version 10.
Solutions	• Solution data is not translated; therefore, you must solve legacy HFSS projects again in version 10.

Closing Projects

To close the current HFSS or HFSS-IE project, select **HFSS or HFSS-IE>Close**. This closes the project without exiting HFSS or HFSS-IE. If there you have a simulation running, you see a message saying that if you continue, HFSS or HFSS-IE will abort before closing. If you OK, HFSS or HFSS-IE aborts the simulation and closes the project.

Related Topics

Projects

Creating Projects

Saving Projects

Deleting Projects

Managing Projects and Designs

Saving Projects

Use the **File>Save As** command to do the following:

- Save a new project.
- Save the active project with a different name or in a different location.
- Save the active project in another file format for use in another program.

Use the **File>Save**  command to save the active project.

HFSS has a "Save before solving" setting located in the **Tools>Options> HFSS** or HFSS-IE **Options** menu. By default this is on. However, for efficiency reasons, the project is only saved if it has been modified since its last save.

A prompt appears when you attempt to save a previously-versioned file. If you agree to the prompt, the file is upgraded to the HFSS version in which you are running the software. In this case the file may no longer be compatible with previous versions. If you do not agree to the prompt, the file is not saved, so the file retains the previous compatibility.

If you have a simulation running, you see a warning that if you continue, HFSS will abort the simulation. If you OK the warning, HFSS or HFSS-IE aborts the simulation and saves the project.

Related Topics

Saving a New Project

Saving the Active Project

Saving a Copy of a Project

Deleting Projects

Managing Projects (Copy, Paste, Delete)

Saving a New Project

1. Click **File**> **Save As**.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Use the correct file extension for the file type.
5. If the window has a **Switch to saved** option, do one of the following:
   - Leave the option selected to display the new file name, and then close the current file.
   - Cancel the **Switch to saved** selection to save the file under the new name without changing which file is displayed.
6. Click **OK**.

HFSS saves the project to the location you specified.

Warning Be sure to save geometric models periodically. Saving frequently helps prevent the loss of your work if a problem occurs.

Although HFSS has an "auto-save" feature, it may not automatically save frequently enough for your needs.

Related Topics

Saving the Active Project

Saving a Copy of a Project

Saving the Active Project

- Click **File**> **Save**  .

  HFSS saves the project over the existing one.

Warning Be sure to save geometric models periodically. Saving frequently helps prevent the loss of your work if a problem occurs.

Although HFSS has an "auto-save" feature, it may not automatically save frequently enough for your needs.

Related Topics

Saving a New Project

Saving a Copy of a Project

Saving a Copy of a Project

To save an existing, active project with a new name, a different file extension, or to a new location:

1. Click **File** >**Save As**.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Select the desired file extension for the file type.
5. If the window has a **Switch to saved** field, do one of the following:
   - Leave the field selected to display the new file name, and then close the current file.
   - Cancel the **Switch to saved** selection to save the file under the new name without changing which file is displayed.
6. Click **OK**.

   HFSS saves the project with the new name or file extension to the location you specified.

Related Topics

Saving a New Project

Saving the Active Project

Renaming a Project

In general, use the **File** menu commands to manage projects. If you move or change the names of files without using these commands, the software may not be able to find information necessary to solve the model. To rename an existing, active project:

1. Select the project in the Project tree.
2. Right-click to display the short-cut menu.
3. Select **Rename**.

   This activates the text field for the project name.
4. Type the new project name and press enter.

   The new project name appears in the directory and the project remains in the original location.

Related Topics

Saving a New Project

Saving a Copy of a Project

Saving Project Data Automatically

HFSS stores recent actions you performed on the active project in an auto-save file in case a sudden workstation crash or other unexpected problem occurs. The auto-save file is stored in the same directory as the project file and is named Project*n*.hfss.auto by default, where n is the order in which the project was added to the current session. HFSS automatically saves all data for the project to the auto-save file, except solution data. By default, HFSS automatically saves project data after every 10 edits. An "edit" is any action you performed which changes data in the project or the design, including actions associated with project management, model creation, and solution analysis.

With auto-save activated, after a problem occurs, you may be able to choose to re-open the original project file (Project*n*.hfss), in an effort to recover the solution data, or open the auto-save file. If the original file is not available, attempting to open the file provides a message that the autosave is being used. If neither file is available, an error message is displayed.

To modify the auto-save settings:

1. Click **Tools** > **Options**>**General Options**.

   The **Options** dialog box appears.
2. Under the **Project Options** tab, verify that **Do Autosave** is selected.

   This option is selected by default.
3. In the **Autosave interval** box, enter the number of *edits* that you want to occur between automatic saves. By default, this option is set at 10.

Note Auto-save *always* increments forward; therefore, even when you undo a command, HFSS or HFSS-IE counts it as an edit.

4. Click **OK** to apply the specified auto-save settings.

Once the specified number of edits is carried out, a "model-only" save will occur. This means that HFSS or HFSS-IE does not save solutions data or clear any undo/redo history.

When HFSS or HFSS-IE auto-saves, an ".auto" extension is appended to the original project file name. For example, "Project1.hfss" will automatically be saved as "Project1.hfss.auto".

Warning When you close or rename a project, HFSS or HFSS-IE deletes the auto-save file. HFSS or HFSS-IE assumes that you have saved any desired changes at this point.

Related Topics

Recovering Project Data in an Auto-Save File

Save Before Solve Option

Save Before Solve Option

The **Tools> HFSS or HFSS-IE Options** command displays a dialog with a checkbox for an automatic **Save Before Solve** option. The main purpose is to force a full save before running the solve.

In the case where you start a solve while another solve is running, and the Save Before Solve option is set, HFSS or HFSS-IE asks if you want solve without saving first. This lets you do multiple solves, and if you have not edited the project in between solves, crash recovery will work. In any case, you can start a new solve while running another without having to abort the running solve.

Related Topics

Recovering Project Data in an Auto-Save File

Recovering Project Data in an Auto-Save File

Following a sudden workstation crash or other unexpected problem, you can recover the project data in its auto-save file.

Warning When you recover a project's auto-save file you *cannot* recover any solutions data; recovering an auto-save file means you will lose any solutions data that existed in the original project file.

To recover project data in an auto-save file:

1. If HFSS or HFSS-IE has crashed, launch HFSS from your desktop.
2. Click **File>Open**, and then select the original Project*n*.hfss project file for which you want to recover its Project*n*.hfss.auto auto-save file.

   The **Crash Recovery** window appears, which gives you the option to open the original project file or the auto-save file.
3. Select **Open project using autosave file** to recover project data in the auto-save file, and then click **OK**. HFSS or HFSS-IE replaces the original project file with the data in the auto-save file.

   HFSS or HFSS-IE immediately overwrites the original project file data with the auto-save file data, removing the results directory (solutions data) from the original project file as it over-

writes to the auto-save file.

Warning If you choose to recover the auto-save file, you cannot recover the original project file that has been overwritten; recovering data in an auto-save file is *not* reversible.

Related Topics

Saving Project Data Automatically

Save Before Solve Option

Deleting Projects

To delete a project:

1. Select the project in the project tree.
2. Click either **Edit>Delete**, or right click to display the short-cut menu and select **Delete**.

   A dialog displays the message: "The project selected and all its files will be deleted from the permanent storage medium. Click OK to proceed."
3. Click **OK** to delete the files or **Cancel** to retain them.

Related Topics

Projects

Creating Projects

Saving Projects

Managing Projects and Designs

Archiving Projects

Use the **File>Archive...** command to place a project and any other files related to the project that you want to include in a <product>z file or zip format archive. You can make notes about the contents of the archive and specify whether to include results and solutions files. The Archive command attempts to automatically detect the necessary files for linked projects and automatically include them in the archive. You can also add additional files to the archive, including results files, external files and projects. For example, if a project linked to the main project also has linked or associated files, you can add them.

Archive File Types

Internally, project archive files are .zip files, and are compatible with any program that can read .zip files (e.g. WinZip, 7Zip). The naming convention is that project archive files will have an extension that is unique for each product. The extension is generated by adding a 'z' to the project file extension (e.g. .hfssz, .adsnz). This extension will be displayed as the default when saving and restoring archive files. Note that we also display .zip as a possible filter in the file selection dialogs.

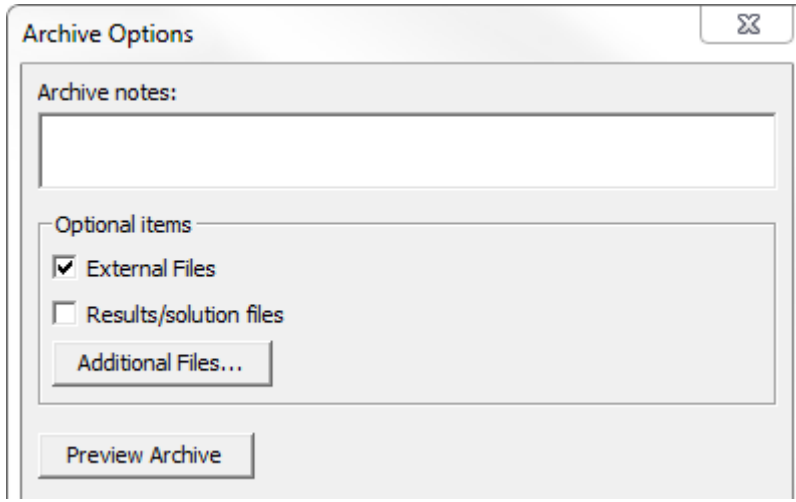
Archive Preview

The Archive command includes a preview feature that lets you review the contents of a planned archive.

To Archive the current project:

1. Click **File>Archive...**

   The **Archive Options** dialog opens.



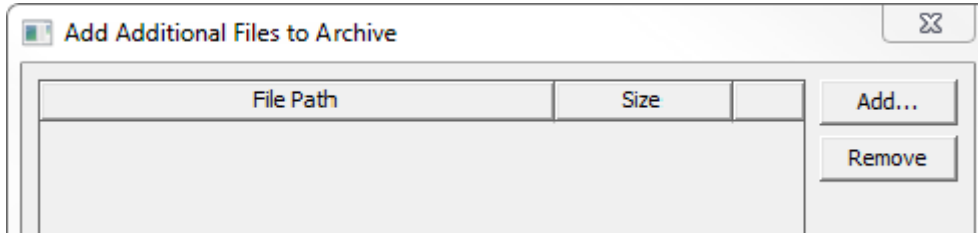
   Archive Notes: you can specify notes that will be visible when previewing the archive. These notes can be viewed from the preview dialog without actually restoring the archive.

   External Files: selecting this checkbox will cause all external files to be included in the archive. The External Files checkbox refers to any existing files associated with the project,

such as linked files, or files added through the **Project>Insert Documentation File** command or **Project>Data Set command**.

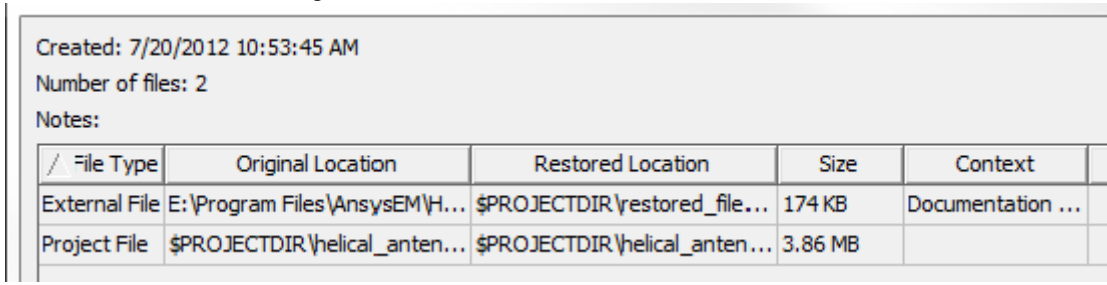
Results Files: selecting this checkbox will cause the entire results directory to be included in the archive. This may greatly increase the size of the archive file.

Clicking **Additional Files...** opens the **Add Additional Files to Archive** dialog.

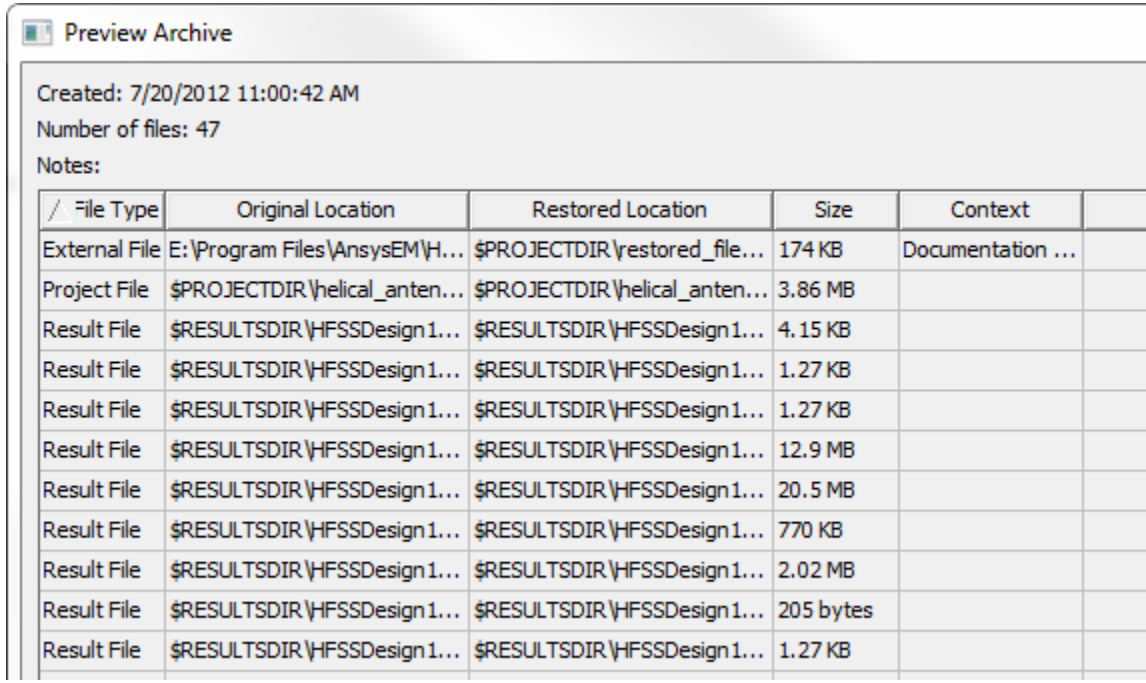


From here, you can click **Add...** to open a file browser to locate any additional files you want to include in the archive. You can select and the **Remove** any files listed. You can **OK** or **Cancel** any proposed changes.

2. Select any optional items, and make any desired Archive notes in the text field
3. When you have made your selections for Optional items, you can select Preview Archive to look at the archive contexts, and the locations for the archive and where restoring from the archive would place them.



Checking the Results option would archive those files to the Preview would show the results files.


Preview Archive

Created: 7/20/2012 11:00:42 AM
Number of files: 47
Notes:

File Type	Original Location	Restored Location	Size	Context
External File	E:\Program Files\AnsysEM\H...	$PROJECTDIR\restored_file...	174 KB	Documentation ...
Project File	$PROJECTDIR\helical_anten...	$PROJECTDIR\helical_anten...	3.86 MB	
Result File	$RESULTSDIR\HFSSDesign1...	$RESULTSDIR\HFSSDesign1...	4.15 KB	
Result File	$RESULTSDIR\HFSSDesign1...	$RESULTSDIR\HFSSDesign1...	1.27 KB	
Result File	$RESULTSDIR\HFSSDesign1...	$RESULTSDIR\HFSSDesign1...	1.27 KB	
Result File	$RESULTSDIR\HFSSDesign1...	$RESULTSDIR\HFSSDesign1...	12.9 MB	
Result File	$RESULTSDIR\HFSSDesign1...	$RESULTSDIR\HFSSDesign1...	20.5 MB	
Result File	$RESULTSDIR\HFSSDesign1...	$RESULTSDIR\HFSSDesign1...	770 KB	
Result File	$RESULTSDIR\HFSSDesign1...	$RESULTSDIR\HFSSDesign1...	2.02 MB	
Result File	$RESULTSDIR\HFSSDesign1...	$RESULTSDIR\HFSSDesign1...	205 bytes	
Result File	$RESULTSDIR\HFSSDesign1...	$RESULTSDIR\HFSSDesign1...	1.27 KB	

To read longer locations, you can drag the column header to expand them. For longer lists, you can use the scroll bar to view more of the list.

Previewing an archive before creating the archive can be helpful in order to see exactly what files will be included in an archive, as well as how those files are being relocated. Another purpose of previewing an archive is to view warnings and consider if any additional files need to be added to the archive.

The preview dialog also displays the archive notes, creation date, and number of included files.

4. When you are ready to create the archive, close the preview, and specify the format you want to use, Ansoft <product> Project archive (<product>z) or zip, and specify the archive location and name. Click OK to create the archive.

File Relocating

In a project to be archived, external files can be located anywhere on the user's system. One of the goals is for the restored project to be relatively self contained, and to NOT allow the restoring of an archived project to haphazardly write files anywhere on the restoring user's system.

To achieve this, it is sometimes necessary to change the location of files in the archived project such that the external files are now located in the project directory. At archive time, any external files not located in the project directory are relocated to the restored_files subdirectory of the project directory in the archived project. Any external files located in the user library or system library will be relocated to the personal library directory. Note that the project file that is written into the

archive will be updated to refer to the files at the new locations, and the original project file will remain unaltered.

Related Topics

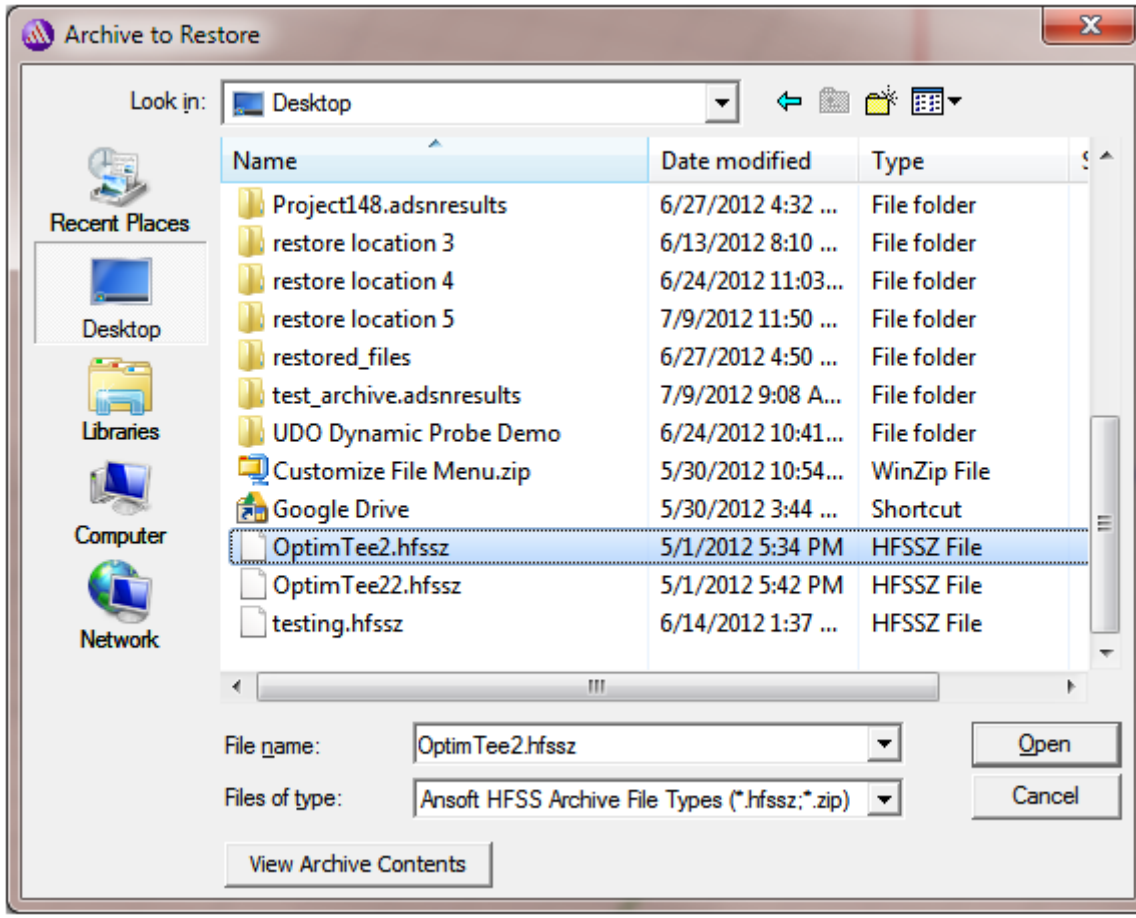
Restore Archive Command

Restore Archive Command

To restore an existing archive created with **Project>Archive...**, use the **Project>Restore Archive** command.

1. Click **Project>Restore Archive**

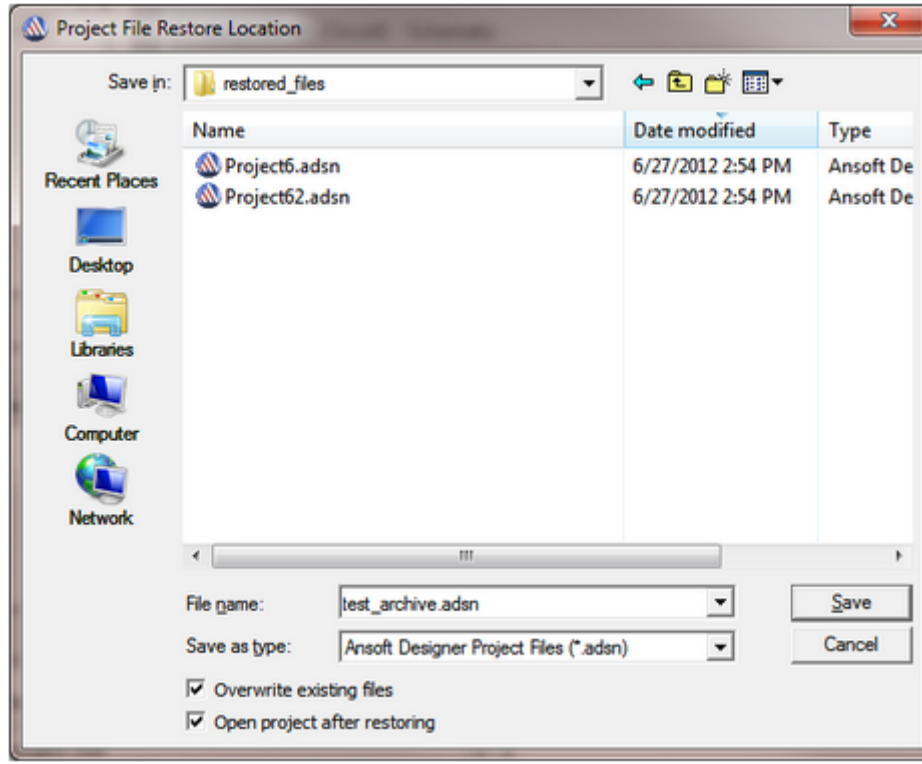
   This displays an **Archive to Restore** browser window that lets you navigate your file system for archive files of <product>z or *.zip.



2. If you select a valid archive file, you can click View Archive Contents to preview the contents.

   The preview dialog will show the same warnings that were generated at archive time, and these warnings may be useful to identify additional steps that are needed to update any files to refer to files which had to be manually added to the archive.

3. Click Open to change to a **Project File Restore Location** browser.



You can edit the file name, and check options to Overwrite existing files and to Open project after restoring.

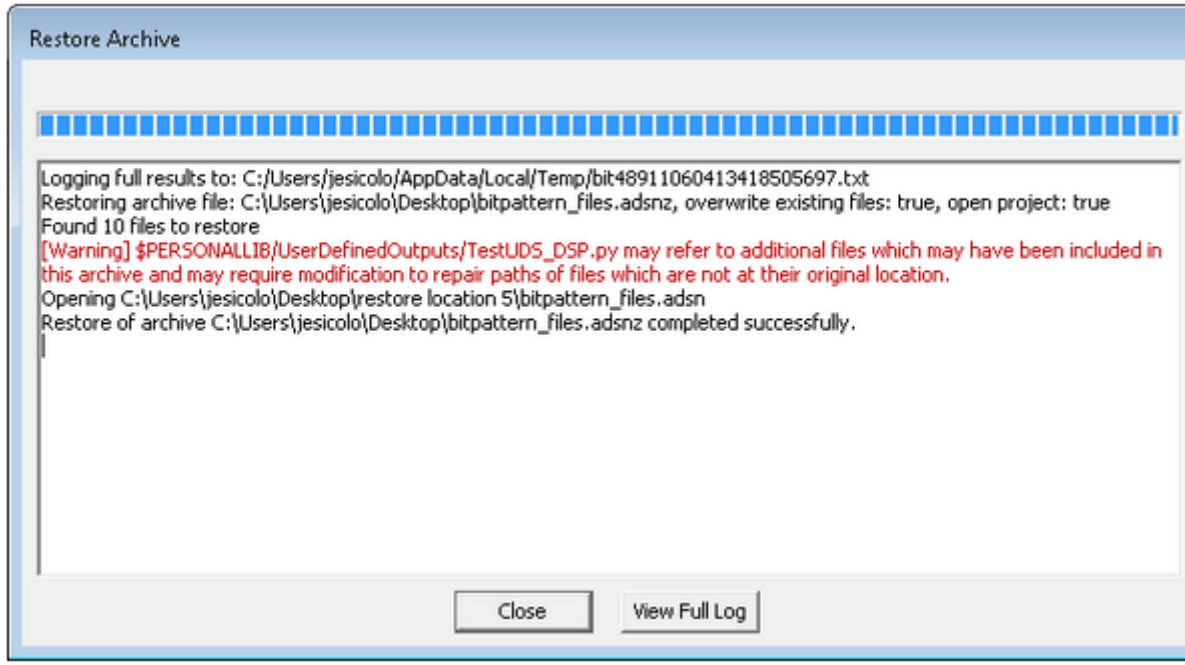
Overwrite existing files: If this button is checked, the restored files will automatically overwrite existing files during the restore process. If this button is unchecked, existing files will not be modified.

Open project after restoring: If this button is checked, the project will be opened in this instance of the application after all files have been restored.

4. Click Save to restore the archived file.

While restoring an archive, a dialog is displayed showing the restore results. The progress bar shows the relative progress, and the text window displays only important information and

warnings.



A full log file is also generated which contains detailed information about the restore process. The first line in the text window displays the location of the full log file. After the restore has been completed, the user can click "View Full Log" to view this log file, or they can use a text editor and open the file at the specified location.

Related Topics

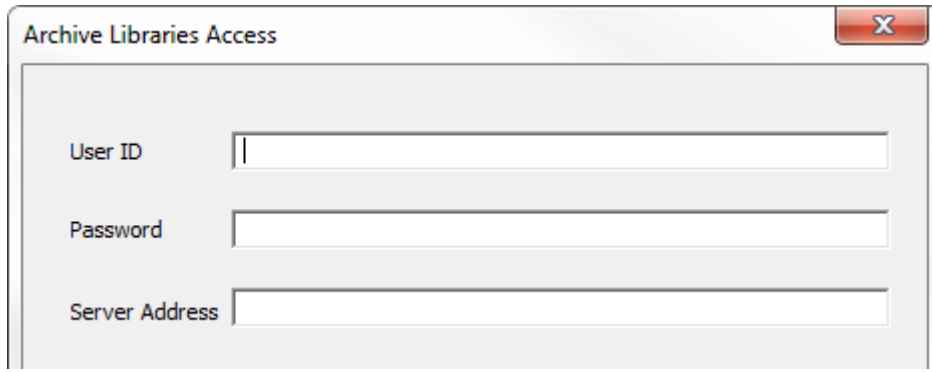
Archiving Projects

Download From Server

If your installation has setup a server location with project archives or other files, you can use **File>Download From Server** to access them.

For Archive files, click **File>Download from Server>Project Archives...**

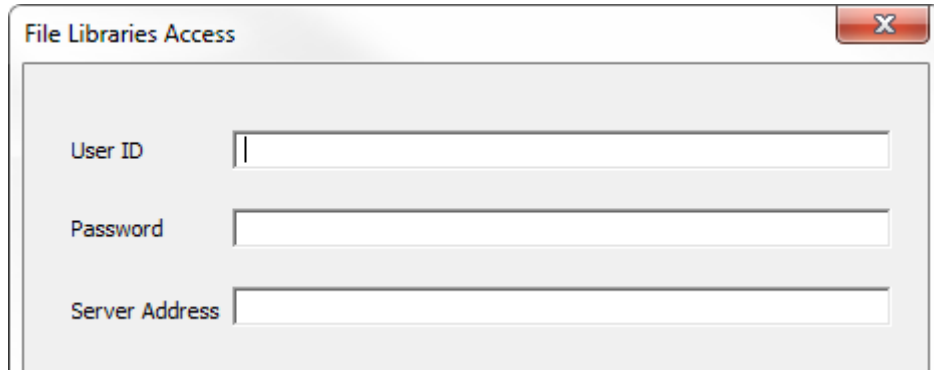
This opens the **Archive Libraries Access** dialog.



Here you specify a User ID, Password, and Server address. You can then browse for and select the files needed.

For File Libraries Access, clock **File>Download From Server>Other Files...**

This opens the **File Library Access** dialog.



Here you specify a User ID, Password, and Server address. You can then browse for and select the files needed.

Undoing Commands

Use the **Undo** command on the **Edit** menu to cancel, or undo, the last action you performed on the active project or design. This is useful for undoing unintended commands related to project management, model creation, and post-processing.

1. In the **Project Manager** window, do one of the following:
   - To undo the last action you performed on the *active project*, such as inserting a design or adding project variables, click the project icon.
   - To undo the last action you performed on the *active design*, such as drawing an object or deleting a field overlay plot, click the design icon.

Note You cannot undo an analysis that you've performed on a model, that is, the **HFSS>Analyze** command.

2. Click **Edit> Undo**, or click the **Undo** button  on the toolbars.

   Your last action is now undone.

Note When you save a project, HFSS always clears the entire undo/redo history for the project and its designs.

Related Topics

Redoing Commands

Redoing Commands

Use the **Redo** command on the **Edit** menu to reapply, or redo, the last action that was canceled, or undone. You can redo a canceled action related to project management, model creation, and post-processing.

1. In the **Project Manager** window, do one of the following:
   - To redo the last action you canceled on the *active project*, such as inserting a design or adding project variables, click the project icon.
   - To redo the last action you canceled on the *active design*, such as drawing an object or deleting a field overlay plot, click the design icon.
2. Click **Edit>Redo**, or click the **Redo** button  on the toolbars.

   Your last canceled action is now reapplied.

Note When you save a project, HFSS always clears the entire undo/redo history for the project and its designs.

Related Topics

Undoing Commands

Updating Design Components

To update components defined in the current design:

1. Click **Tools>Update Definitions**.
   The **Update Definitions** dialog box appears.
2. Select one of the following two radio buttons.
   - Show Items with newer definitions
   - Show All Items
3. From the **Show Types** list in the **List Options** section, select the types of definitions you want to show in the **Item List** list.
4. Select the item you want to update from the **Item List** list.
5. Click the **Select All** or **Unselect All** buttons to select or clear all items listed.
6. Click **Update**.
   A message appears telling you the update was successful. Click **OK** to close the message.

When you are finished updating definitions, click **Close**.

Managing Projects and Designs

You can copy and paste projects or designs, rename them, and delete them.

Related Topics

Saving Projects

Copy and Paste a Project or Design

To **copy** a project or design:

1. Select a project or design in the project tree in Project Manage Window to enable the menu command **Edit>Copy**.
2. Click **Edit>Copy**. The project or design is copied for pasting.

To **paste** a project or design:

1. Select a project or design in the project tree in Project Manage Window to enable the menu command **Edit>Paste**.
2. Click **Edit>Paste**. The project or design is pasted under the selected project, an icon is added to the project tree.

Rename a Project or Design

1. Right click the design icon to display the shortcut menu, as shown.
2. Click **Rename** in the shortcut menu.
3. Define the new name for the design by typing it directly into the Project Window.
4. Press **Enter** to complete the rename.

Delete a Project or Design

1. Select a project or design in the project tree in Project Manage Window to enable the menu command **Edit>Delete**.
2. Click **Edit>Delete**, or the **Delete** key.
3. Confirm the warning box to complete the delete operation, the icon is removed from the project tree.

Validating Projects

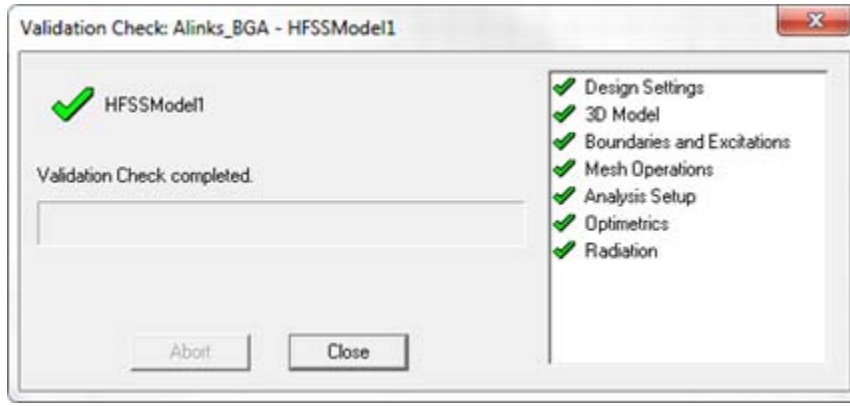
Before you run an analysis on a model, it is very important that you first perform a validation check on the project. When you perform a validation check on a project, HFSS runs a check on all the setup details of the active project to verify that all the necessary steps have been completed and their parameters are reasonable.

To perform a validation check on the active project:

1. Click **HFSS**>**Validation Check**  .

   HFSS checks the project setup, and then the **Validation Check** window appears.

2. View the results of the validation check in the **Validation Check** window.



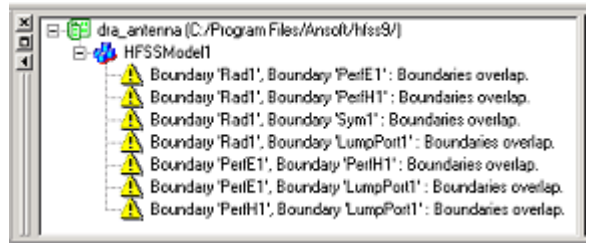
The following icons can appear next to an item:

 Indicates the step is complete.

 Indicates the step is incomplete.

 Indicates the step may require your attention.

3. View any messages in the **Message Manager** window.



4. If the validation check indicates that a step in your project is incomplete or incorrect, carefully

review the setup details for that particular step and revise them as necessary.

5. Click **HFSS**>**Validation Check** to run a validation check after you have revised any setup details for an incomplete or incorrect project step.
6. Click **Close**.

Related Topics

Modeler Validation Settings

Modeler Validation Settings

You can adjust the degree to which the software checks a model for faults that could jeopardize mesh accuracy. There are three levels of model validation that a user can specify for a given design: **Warning Only**, **Basic**, and **Strict**. Note that this setting affects only the "3D Model" stage of a design validation.

- The **Warning Only** entity check setting allows all models to pass 3D Model validation regardless of any faults that are found (acis_entity check errors). These faults are posted in the message window as warnings.
- The **Basic** entity check setting allows most models to pass 3D Model validation. This excuses non-manifold errors and most acis_entity_check errors. Some faults are flagged as model errors (basic entity check errors), thereby prohibiting a design from proceeding to the meshing stage of an analysis. You must either correct such errors before attempting to analyze the design under the Basic setting, or you must change the Model Validation level to Warning Only.
- The **Strict** entity check setting enforces a tighter tolerance for model faults than the "Warning Only" and "Basic" settings. All model faults that are found during 3D Model validation are posted to the message window. These errors must be corrected before attempting to analyze the design under the Strict setting, or you must change the Model Validation level to Basic or Warning Only.

To set the Model Validation level:

1. Select **Modeler->Validation Settings**.

   This displays the **Validation Settings** dialog that lets you set the validation as basic, strict, or warning only.
2. Choose the desired level of validation from the **Entity Check Level** drop down menu.

   You can also click the **Save as Default** button to make the current selection the default. You can select the **Restore Default** button.
3. Click OK to accept the selection and close the dialog.

Related Topics

Model Analysis

Analyze Objects

Interobject Misalignment

Analyze Surface Mesh

Heal

Healing State On: Validation Check

Show Analysis dialog

Align Faces

Remove Faces

Remove Edges

Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Exporting Files

You can export the following types of files from HFSS or HFSS-IE:

- Ansoft 2D modeler files
- 3D model files
- Graphics files
- Data tables

Related Topics

Exporting Matrix Data

Exporting Equivalent Circuit Data

Exporting 2D Geometry Files

When you export a file in a 2D geometry format (the Ansoft 2D Modeler (**.sm2**) format or the AutoCAD DXF (**.dxf**) format), the geometry located within the xy plane is exported.

Note If you want to export a plane that does not coincide with the global xy plane, you must create a relative coordinate system to redefine the location of the origin.

See Creating a Relative Coordinate System for more information.

To export a file to a **.sm2** or **.dxf** format:

1. Click **Modeler**>**Export** to save the file in an Ansoft 2D Modeler format.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Select **Ansoft 2D Geometry Files (*.sm2)** or **AutoCAD DXF Files (*.dxf)** from the **Save as type** pull-down list.
5. Click **Save**. The file is exported to the specified location with the appropriate file format.

Related Topics

Exporting 3D Model Files

Exporting Graphics Files

Exporting 3D Model Files

You can export HFSS or HFSS-IE 3D models to 3D model file formats:

To export a file to a 3D model format:

1. Click **Modeler**>**Export** to save the file in a 3D model format.

   The **Export File** dialog box appears.
2. Use the file browser to find the directory where you want to save the file.
3. Enter the name of the file in the **File name** box.
4. Select the desired 3D model file format from the **Save as type** pull-down list.

5. Click **Save**.

   The file is exported to the specified location as a 3D model file

Extension	Contents
.sat	ACIS geometry solid model files.
.sm2	Ansoft 2D Geometry File
.sm3	Ansoft 3D Modeler files in ACIS version 2.0 or greater.
.anstGeom	Ansoft Geometry File supporting UDMs and CAD integration with the Workbench.
.dxf, .dwg	AutoCAD Drawing Interchange Format files.
.model, .exp	.model - CATIA 4.1.9 to 4.2.4 .Catia V5 R6-R23 Export Files
.gds	GDSII files
.iges, .igs	Industry standard Initial Graphics Exchange Specification (IGES) files 5.3. AN additional license is required.
.x_b, .x_t	Parasolid Binary Files. Parasolid Text Files
.sab	Standard ACIS binary.
.step, .stp	Industry standard AP203 STEP files. An additional license is required.

6. If you selected **.sat** or .sat **.sm3**, the **Select Version** dialog box appears. Do the following:
   - Click an ACIS version in which to export the model from the **ACIS SM3 or SAT Version** pull-down list, and then click **OK**.
7. Click **Save**. Unless you selected GDSII, the file is exported to the specified location as a 3D model file. If you selected GDSII, the **GDSII Export** dialog appears.
   - If the model has been defined with layers, those layers are listed by layer number in the table, with columns for Layer Name, Layer Number, Elevation in units. There is a check-box to specify whether to include the layer in the exported file.
   - If you have defined a layer map file for the model, the **Layermap** button opens a browser for you to open that file before export. The *.layermap file is a text file that maps the

GDSII layer numbers to layer names in the stackup. The *.layermap file can have the same format as the .tech file used in GDSII import, but it only needs the layer name and number in the file. In a *.layermap file, other information is ignored.

- In the Polygon Vertices area, check a radio button to select either **No Limit to the number of vertices** or **Limit the number of vertices** to a specified value.
- For Arc tolerance, specify a value or accept the default.

8. Click the **OK** button in the GDSII Export dialog to complete the export. The file is exported to the specified location.

Related Topics

Exporting 2D Model Files

Exporting Graphics Files

Importing 3D Model Files

Importing GDSII Format Files

Export Results to Thermal Link for ANSYS Mechanical

Exporting Graphics Files

You can export the following graphics formats:

Extension	Contents
.bmp	Bitmap files.
.gif	Graphics Interchange Format files.
.jpeg	Joint Photographics Experts Group files.
.tiff	Tagged Image File Format files.
.wrl	Virtual Reality Modeling Language (VRML) files.

To export a file to a graphics format:

1. Click **Modeler**>**Export** to save the file in a graphics format.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Select the desired graphics file format from the **Save as type** pull-down list.
5. Click **Save**. The file is exported to the specified location as a graphics file.

Related Topics

Exporting 2D Model Files

Exporting 3D Model Files

Exporting Reports as Graphics

Exporting Data Table Files

You must have an existing plot open to see the **Report2D** menu.

1. Click **Report2D**>**Export to File**.
   - Alternatively, right-click on the data table, and then click **Export to File** on the shortcut menu.

   The **Export plot data to file** dialog box appears.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Select one of the following file formats from the **Save as type** pull-down list:

Extension	Contents
.txt	Post processor format file
.csv	Comma-delimited data file
.tab	Tab-separated file
.dat	Ansoft plot data file

5. Click **Save**. The file is exported to the specified location as a data table file.

Related Topics

Exporting Matrix Data

Exporting Equivalent Circuit Data

Exporting Ansoft Report Data Format Files

Importing Files

You can import the following types of files to HFSS or HFSS-IE:

- 2D model files
- 3D model files
- Solution data files
- Data table files
- HFSS or HFSS-IE Plot Data
- Plot Data (comma delimited files)

The import dialog contains a check box for the **Heal** command which is enabled by default.

Related Topics

Exporting Files

Importing 2D Model Files

You can read 2D model files directly into the active **Modeler** window:

Note If you import a file into an active **Modeler** window that contains an existing model, the file is added to the existing model; it will not replace it.

To import a 2D model file:

1. Click **Modeler**>**Import**.
   The **Import File** dialog box appears.
2. Select a file type from the **Files of type** pull-down list. For 2D model files, this would be either **GDSII Files (*.gds)** or **Ansoft 2D Geometry Files (*.sm2)**.
3. Use the file browser to find and select the file you want to import.
4. Click **Open**.
   The file is imported into the active **Modeler** window.

Extension	Contents
.gds	GDSII is a standard file format for 2D graphical design layout data.
.sm2	Ansoft 2D Modeler files.

Note When importing .sm2 files, they will import into the current XY or XZ plane depending upon how they were originally created. If you want to import them in a specific orientation other than the current XY or XZ plane, you must first create a relative coordinate system with the planes in the desired orientation.

See Creating a Relative Coordinate System for more information.

Related Topics

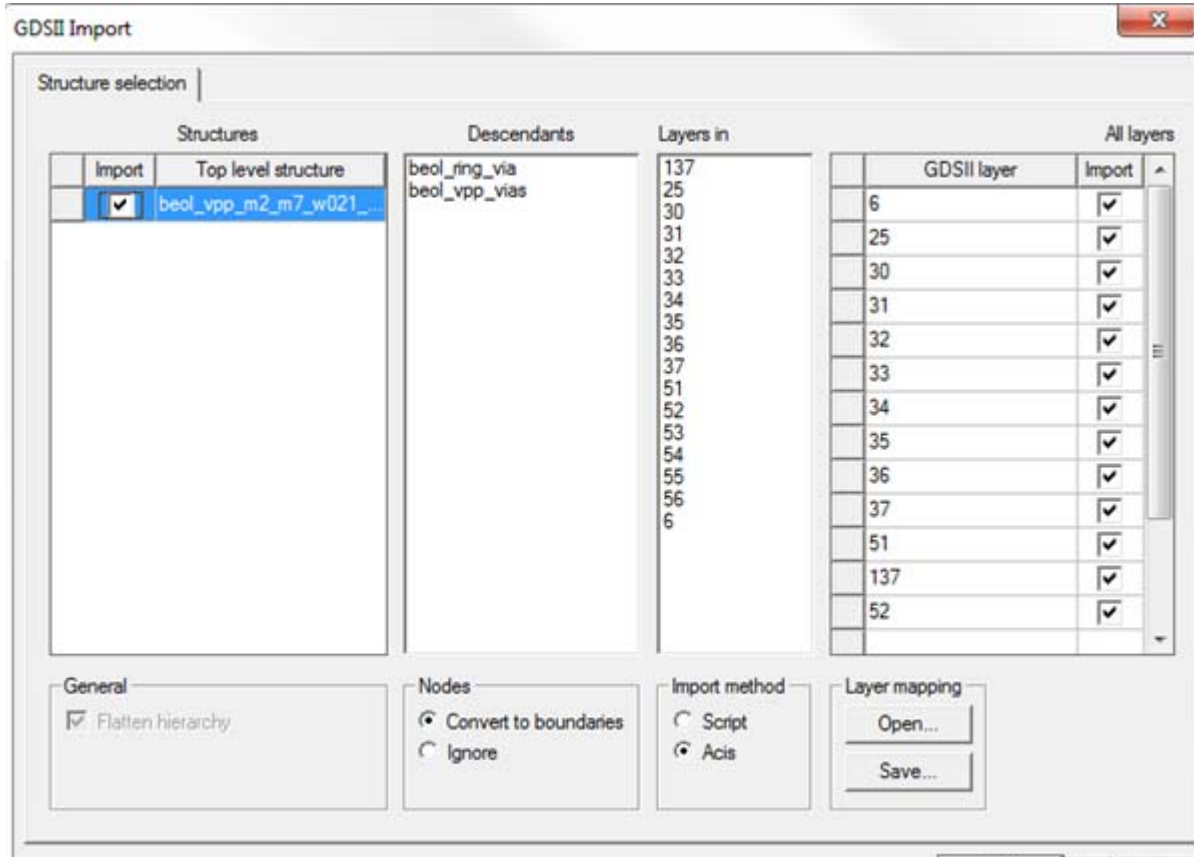
Importing 3D Model Files

Importing GDSII Format Files

Importing GDSII Format Files

See the introductory topic Importing 2D Model Files for the initial steps in the process of importing 2D data.

The process for importing GDSII format files into the HFSS or HFSS-IE uses a single dialog box:



GDSII Structures Panel

The GDSII file may contain several top level structures.

1. Click on a structure name in the **GDSII Structures** panel to highlight it.
2. Clicking on the **Select** checkbox in the **GDSII Structures** panel both highlights the structure and selects that top-level structure to be imported.

When multiple structures are imported, HFSS or HFSS-IE creates multiple designs under the current project, one for each of the GDSII structures.

Descendants Panel

The GDSII file is hierarchical and may contain many sub-layouts. The **Descendants** panel shows the sub-layouts in the selected top-level designs.

Layers for *structurename* Panel

The **Layers for *structurename*** panel shows the layers for the (most recently) highlighted top level structure [*structurename*]. GDSII layers are identified by layer numbers.

All Layers Panel

The **All Layers** panel lists all the layers from all the structures in the file.

Use the **Import** check boxes in the **All GDSII Layers** panel to select the layers to import. You can drag and drop the layers in the list to change the vertical stackup of layers.

General Field

The **Flatten hierarchy** checkbox is automatically selected. HFSS always flattens any hierarchical geometry in the GDSII.

Nodes Field

GDSII supports nodes and boundaries as separate data types. Normally, boundaries represent polygons. HFSS can either convert objects that use the nodes data type to boundary types, or can ignore them. Use the radio buttons to select **Convert to boundaries** or **Ignore**. The default is to convert data type nodes to data type boundary.

Import Method Field

Use the radio button to select the import method as Script of Acis

Layer Mapping Field

If desired, you can create a mapping of the GDSII layer numbers to layer names in the design stackup. To create and use the mapping.

1. Use a text editor to create a text file that maps the GDSII layer numbers to layer names in the stackup. The layer mapping file must have a **.tech** suffix.

   The .tech format:

   - / is the comment character
   - Units may be specified with a line UNITS <string> before the lines of layer information. <string> is any of the allowed desktop length units. The default units is nm.
   - Each layer is specified by a line that contains <*import layer*> <*product layer*> <*layer color*> <*layer elevation*> <*layer thickness*>

where:

<*import layer*> - the name of the DXF layer

<*destination layer*> - the name to map the DXF layer to. This specifies the layout layer in Designer. In HFSS, it is used as part of the name for objects imported from this DXF layer.

<*layer color*> - A string from the choices listed here, for example. blue2

<*layer elevation*> - double

<*layer height*> - double

2. For example:

```
UNITS um
/
/ ------------------------------------------------------------
/ import#  destination     Color    Elevation  Thickness
/ ------------------------------------------------------------
S12         signal12         blue      1100       530
TR          trace            red       6620       530
S3          signal3          yellow    8150       2000
```

3. Click the **Open** button in the **Layer mapping** panel to locate and open an existing layer mapping file.
4. Click **OK**.

   The file is imported into the active **Modeler** window.

   You can use the **Save** button to save a layer mapping file with the current settings.

Tech File Colors with RGB Values

Color string choices with corresponding RBG values

snow RBG = { 255, 250, 250},

ghost white RBG = { 248, 248, 255},

GhostWhite RBG = { 248, 248, 255},

white smoke RBG = { 245, 245, 245},

WhiteSmoke RBG = { 245, 245, 245},

gainsboro RBG = { 220, 220, 220},

floral white RBG = { 255, 250, 240},

FloralWhite RBG = { 255, 250, 240},

old lace RBG = { 253, 245, 230},

OldLace RBG = { 253, 245, 230},

linen RBG = { 250, 240, 230},

antique white RBG = { 250, 235, 215},

AntiqueWhite RBG = { 250, 235, 215},
papaya whip RBG = { 255, 239, 213},
PapayaWhip RBG = { 255, 239, 213},
blanched almond RBG = { 255, 235, 205},
BlanchedAlmond RBG = { 255, 235, 205},
bisque RBG = { 255, 228, 196},
peach puff RBG = { 255, 218, 185},
PeachPuff RBG = { 255, 218, 185},
navajo white RBG = { 255, 222, 173},
NavajoWhite RBG = { 255, 222, 173},
moccasin RBG = { 255, 228, 181},
cornsilk RBG = { 255, 248, 220},
ivory RBG = { 255, 255, 240},
lemon chiffon RBG = { 255, 250, 205},
LemonChiffon RBG = { 255, 250, 205},
seashell RBG = { 255, 245, 238},
honeydew RBG = { 240, 255, 240},
mint cream RBG = { 245, 255, 250},
MintCream RBG = { 245, 255, 250},
azure RBG = { 240, 255, 255},
alice blue RBG = { 240, 248, 255},
AliceBlue RBG = { 240, 248, 255},
lavender RBG = { 230, 230, 250},
lavender blush RBG = { 255, 240, 245},
LavenderBlush RBG = { 255, 240, 245},
misty rose RBG = { 255, 228, 225},
MistyRose RBG = { 255, 228, 225},
white RBG = { 255, 255, 255},
black RBG = { 0, 0, 0},
dark slate gray RBG = { 47, 79, 79},
DarkSlateGray RBG = { 47, 79, 79},
dark slate grey RBG = { 47, 79, 79},
DarkSlateGrey RBG = { 47, 79, 79},
dim gray RBG = { 105, 105, 105},
DimGray RBG = { 105, 105, 105},

dim grey RBG = { 105, 105, 105},
DimGrey RBG = { 105, 105, 105},
slate gray RBG = { 112, 128, 144},
SlateGray RBG = { 112, 128, 144},
slate grey RBG = { 112, 128, 144},
SlateGrey RBG = { 112, 128, 144},
light slate gray RBG = { 119, 136, 153},
LightSlateGray RBG = { 119, 136, 153},
light slate grey RBG = { 119, 136, 153},
LightSlateGrey RBG = { 119, 136, 153},
gray RBG = { 190, 190, 190},
grey RBG = { 190, 190, 190},
light grey RBG = { 211, 211, 211},
LightGrey RBG = { 211, 211, 211},
light gray RBG = { 211, 211, 211},
LightGray RBG = { 211, 211, 211},
midnight blue RBG = { 25, 25, 112},
MidnightBlue RBG = { 25, 25, 112},
navy RBG = { 0, 0, 128},
navy blue RBG = { 0, 0, 128},
NavyBlue RBG = { 0, 0, 128},
cornflower blue RBG = { 100, 149, 237},
CornflowerBlue RBG = { 100, 149, 237},
dark slate blue RBG = { 72, 61, 139},
DarkSlateBlue RBG = { 72, 61, 139},
slate blue RBG = { 106, 90, 205},
SlateBlue RBG = { 106, 90, 205},
medium slate blue RBG = { 123, 104, 238},
MediumSlateBlue RBG = { 123, 104, 238},
light slate blue RBG = { 132, 112, 255},
LightSlateBlue RBG = { 132, 112, 255},
medium blue RBG = { 0, 0, 205},
MediumBlue RBG = { 0, 0, 205},
royal blue RBG = { 65, 105, 225},
RoyalBlue RBG = { 65, 105, 225},

blue RBG = { 0, 0, 255},
dodger blue RBG = { 30, 144, 255},
DodgerBlue RBG = { 30, 144, 255},
deep sky blue RBG = { 0, 191, 255},
DeepSkyBlue RBG = { 0, 191, 255},
sky blue RBG = { 135, 206, 235},
SkyBlue RBG = { 135, 206, 235},
light sky blue RBG = { 135, 206, 250},
LightSkyBlue RBG = { 135, 206, 250},
steel blue RBG = { 70, 130, 180},
SteelBlue RBG = { 70, 130, 180},
light steel blue RBG = { 176, 196, 222},
LightSteelBlue RBG = { 176, 196, 222},
light blue RBG = { 173, 216, 230},
LightBlue RBG = { 173, 216, 230},
powder blue RBG = { 176, 224, 230},
PowderBlue RBG = { 176, 224, 230},
pale turquoise RBG = { 175, 238, 238},
PaleTurquoise RBG = { 175, 238, 238},
dark turquoise RBG = { 0, 206, 209},
DarkTurquoise RBG = { 0, 206, 209},
medium turquoise RBG = { 72, 209, 204},
MediumTurquoise RBG = { 72, 209, 204},
turquoise RBG = { 64, 224, 208},
cyan RBG = { 0, 255, 255},
light cyan RBG = { 224, 255, 255},
LightCyan RBG = { 224, 255, 255},
cadet blue RBG = { 95, 158, 160},
CadetBlue RBG = { 95, 158, 160},
medium aquamarine RBG = { 102, 205, 170},
MediumAquamarine RBG = { 102, 205, 170},
aquamarine RBG = { 127, 255, 212},
dark green RBG = { 0, 100, 0},
DarkGreen RBG = { 0, 100, 0},
dark olive green RBG = { 85, 107, 47},

DarkOliveGreen RBG = { 85, 107, 47},
dark sea green RBG = { 143, 188, 143},
DarkSeaGreen RBG = { 143, 188, 143},
sea green RBG = { 46, 139, 87},
SeaGreen RBG = { 46, 139, 87},
medium sea green RBG = { 60, 179, 113},
MediumSeaGreen RBG = { 60, 179, 113},
light sea green RBG = { 32, 178, 170},
LightSeaGreen RBG = { 32, 178, 170},
pale green RBG = { 152, 251, 152},
PaleGreen RBG = { 152, 251, 152},
spring green RBG = { 0, 255, 127},
SpringGreen RBG = { 0, 255, 127},
lawn green RBG = { 124, 252, 0},
LawnGreen RBG = { 124, 252, 0},
green RBG = { 0, 255, 0},
chartreuse RBG = { 127, 255, 0},
medium spring green RBG = { 0, 250, 154},
MediumSpringGreen RBG = { 0, 250, 154},
green yellow RBG = { 173, 255, 47},
GreenYellow RBG = { 173, 255, 47},
lime green RBG = { 50, 205, 50},
LimeGreen RBG = { 50, 205, 50},
yellow green RBG = { 154, 205, 50},
YellowGreen RBG = { 154, 205, 50},
forest green RBG = { 34, 139, 34},
ForestGreen RBG = { 34, 139, 34},
olive drab RBG = { 107, 142, 35},
OliveDrab RBG = { 107, 142, 35},
dark khaki RBG = { 189, 183, 107},
DarkKhaki RBG = { 189, 183, 107},
khaki RBG = { 240, 230, 140},
pale goldenrod RBG = { 238, 232, 170},
PaleGoldenrod RBG = { 238, 232, 170},
light goldenrod yellow RBG = { 250, 250, 210},

LightGoldenrodYellow RBG = { 250, 250, 210},
light yellow RBG = { 255, 255, 224},
LightYellow RBG = { 255, 255, 224},
yellow RBG = { 255, 255, 0},
gold RBG = { 255, 215, 0},
light goldenrod RBG = { 238, 221, 130},
LightGoldenrod RBG = { 238, 221, 130},
goldenrod RBG = { 218, 165, 32},
dark goldenrod RBG = { 184, 134, 11},
DarkGoldenrod RBG = { 184, 134, 11},
rosy brown RBG = { 188, 143, 143},
RosyBrown RBG = { 188, 143, 143},
indian red RBG = { 205, 92, 92},
IndianRed RBG = { 205, 92, 92},
saddle brown RBG = { 139, 69, 19},
SaddleBrown RBG = { 139, 69, 19},
sienna RBG = { 160, 82, 45},
peru RBG = { 205, 133, 63},
burlywood RBG = { 222, 184, 135},
beige RBG = { 245, 245, 220},
wheat RBG = { 245, 222, 179},
sandy brown RBG = { 244, 164, 96},
SandyBrown RBG = { 244, 164, 96},
tan RBG = { 210, 180, 140},
chocolate RBG = { 210, 105, 30},
firebrick RBG = { 178, 34, 34},
brown RBG = { 165, 42, 42},
dark salmon RBG = { 233, 150, 122},
DarkSalmon RBG = { 233, 150, 122},
salmon RBG = { 250, 128, 114},
light salmon RBG = { 255, 160, 122},
LightSalmon RBG = { 255, 160, 122},
orange RBG = { 255, 165, 0},
dark orange RBG = { 255, 140, 0},
DarkOrange RBG = { 255, 140, 0},

coral RBG = { 255, 127, 80},
light coral RBG = { 240, 128, 128},
LightCoral RBG = { 240, 128, 128},
tomato RBG = { 255, 99, 71},
orange red RBG = { 255, 69, 0},
OrangeRed RBG = { 255, 69, 0},
red RBG = { 255, 0, 0},
hot pink RBG = { 255, 105, 180},
HotPink RBG = { 255, 105, 180},
deep pink RBG = { 255, 20, 147},
DeepPink RBG = { 255, 20, 147},
pink RBG = { 255, 192, 203},
light pink RBG = { 255, 182, 193},
LightPink RBG = { 255, 182, 193},
pale violet red RBG = { 219, 112, 147},
PaleVioletRed RBG = { 219, 112, 147},
maroon RBG = { 176, 48, 96},
medium violet red RBG = { 199, 21, 133},
MediumVioletRed RBG = { 199, 21, 133},
violet red RBG = { 208, 32, 144},
VioletRed RBG = { 208, 32, 144},
magenta RBG = { 255, 0, 255},
violet RBG = { 238, 130, 238},
plum RBG = { 221, 160, 221},
orchid RBG = { 218, 112, 214},
medium orchid RBG = { 186, 85, 211},
MediumOrchid RBG = { 186, 85, 211},
dark orchid RBG = { 153, 50, 204},
DarkOrchid RBG = { 153, 50, 204},
dark violet RBG = { 148, 0, 211},
DarkViolet RBG = { 148, 0, 211},
blue violet RBG = { 138, 43, 226},
BlueViolet RBG = { 138, 43, 226},
purple RBG = { 160, 32, 240},
medium purple RBG = { 147, 112, 219},

MediumPurple RBG = { 147, 112, 219},
thistle RBG = { 216, 191, 216},
snow1 RBG = { 255, 250, 250},
snow2 RBG = { 238, 233, 233},
snow3 RBG = { 205, 201, 201},
snow4 RBG = { 139, 137, 137},
seashell1 RBG = { 255, 245, 238},
seashell2 RBG = { 238, 229, 222},
seashell3 RBG = { 205, 197, 191},
seashell4 RBG = { 139, 134, 130},
AntiqueWhite1 RBG = { 255, 239, 219},
AntiqueWhite2 RBG = { 238, 223, 204},
AntiqueWhite3 RBG = { 205, 192, 176},
AntiqueWhite4 RBG = { 139, 131, 120},
bisque1 RBG = { 255, 228, 196},
bisque2 RBG = { 238, 213, 183},
bisque3 RBG = { 205, 183, 158},
bisque4 RBG = { 139, 125, 107},
PeachPuff1 RBG = { 255, 218, 185},
PeachPuff2 RBG = { 238, 203, 173},
PeachPuff3 RBG = { 205, 175, 149},
PeachPuff4 RBG = { 139, 119, 101},
NavajoWhite1 RBG = { 255, 222, 173},
NavajoWhite2 RBG = { 238, 207, 161},
NavajoWhite3 RBG = { 205, 179, 139},
NavajoWhite4 RBG = { 139, 121, 94},
LemonChiffon1 RBG = { 255, 250, 205},
LemonChiffon2 RBG = { 238, 233, 191},
LemonChiffon3 RBG = { 205, 201, 165},
LemonChiffon4 RBG = { 139, 137, 112},
cornsilk1 RBG = { 255, 248, 220},
cornsilk2 RBG = { 238, 232, 205},
cornsilk3 RBG = { 205, 200, 177},
cornsilk4 RBG = { 139, 136, 120},
ivory1 RBG = { 255, 255, 240},

ivory2 RBG = { 238, 238, 224},
ivory3 RBG = { 205, 205, 193},
ivory4 RBG = { 139, 139, 131},
honeydew1 RBG = { 240, 255, 240},
honeydew2 RBG = { 224, 238, 224},
honeydew3 RBG = { 193, 205, 193},
honeydew4 RBG = { 131, 139, 131},
LavenderBlush1 RBG = { 255, 240, 245},
LavenderBlush2 RBG = { 238, 224, 229},
LavenderBlush3 RBG = { 205, 193, 197},
LavenderBlush4 RBG = { 139, 131, 134},
MistyRose1 RBG = { 255, 228, 225},
MistyRose2 RBG = { 238, 213, 210},
MistyRose3 RBG = { 205, 183, 181},
MistyRose4 RBG = { 139, 125, 123},
azure1 RBG = { 240, 255, 255},
azure2 RBG = { 224, 238, 238},
azure3 RBG = { 193, 205, 205},
azure4 RBG = { 131, 139, 139},
SlateBlue1 RBG = { 131, 111, 255},
SlateBlue2 RBG = { 122, 103, 238},
SlateBlue3 RBG = { 105, 89, 205},
SlateBlue4 RBG = { 71, 60, 139},
RoyalBlue1 RBG = { 72, 118, 255},
RoyalBlue2 RBG = { 67, 110, 238},
RoyalBlue3 RBG = { 58, 95, 205},
RoyalBlue4 RBG = { 39, 64, 139},
blue1 RBG = { 0, 0, 255},
blue2 RBG = { 0, 0, 238},
blue3 RBG = { 0, 0, 205},
blue4 RBG = { 0, 0, 139},
DodgerBlue1 RBG = { 30, 144, 255},
DodgerBlue2 RBG = { 28, 134, 238},
DodgerBlue3 RBG = { 24, 116, 205},
DodgerBlue4 RBG = { 16, 78, 139},

SteelBlue1 RBG = { 99, 184, 255},
SteelBlue2 RBG = { 92, 172, 238},
SteelBlue3 RBG = { 79, 148, 205},
SteelBlue4 RBG = { 54, 100, 139},
DeepSkyBlue1 RBG = { 0, 191, 255},
DeepSkyBlue2 RBG = { 0, 178, 238},
DeepSkyBlue3 RBG = { 0, 154, 205},
DeepSkyBlue4 RBG = { 0, 104, 139},
SkyBlue1 RBG = { 135, 206, 255},
SkyBlue2 RBG = { 126, 192, 238},
SkyBlue3 RBG = { 108, 166, 205},
SkyBlue4 RBG = { 74, 112, 139},
LightSkyBlue1 RBG = { 176, 226, 255},
LightSkyBlue2 RBG = { 164, 211, 238},
LightSkyBlue3 RBG = { 141, 182, 205},
LightSkyBlue4 RBG = { 96, 123, 139},
SlateGray1 RBG = { 198, 226, 255},
SlateGray2 RBG = { 185, 211, 238},
SlateGray3 RBG = { 159, 182, 205},
SlateGray4 RBG = { 108, 123, 139},
LightSteelBlue1 RBG = { 202, 225, 255},
LightSteelBlue2 RBG = { 188, 210, 238},
LightSteelBlue3 RBG = { 162, 181, 205},
LightSteelBlue4 RBG = { 110, 123, 139},
LightBlue1 RBG = { 191, 239, 255},
LightBlue2 RBG = { 178, 223, 238},
LightBlue3 RBG = { 154, 192, 205},
LightBlue4 RBG = { 104, 131, 139},
LightCyan1 RBG = { 224, 255, 255},
LightCyan2 RBG = { 209, 238, 238},
LightCyan3 RBG = { 180, 205, 205},
LightCyan4 RBG = { 122, 139, 139},
PaleTurquoise1 RBG = { 187, 255, 255},
PaleTurquoise2 RBG = { 174, 238, 238},
PaleTurquoise3 RBG = { 150, 205, 205},

PaleTurquoise4 RBG = { 102, 139, 139},
CadetBlue1 RBG = { 152, 245, 255},
CadetBlue2 RBG = { 142, 229, 238},
CadetBlue3 RBG = { 122, 197, 205},
CadetBlue4 RBG = { 83, 134, 139},
turquoise1 RBG = { 0, 245, 255},
turquoise2 RBG = { 0, 229, 238},
turquoise3 RBG = { 0, 197, 205},
turquoise4 RBG = { 0, 134, 139},
cyan1 RBG = { 0, 255, 255},
cyan2 RBG = { 0, 238, 238},
cyan3 RBG = { 0, 205, 205},
cyan4 RBG = { 0, 139, 139},
DarkSlateGray1 RBG = { 151, 255, 255},
DarkSlateGray2 RBG = { 141, 238, 238},
DarkSlateGray3 RBG = { 121, 205, 205},
DarkSlateGray4 RBG = { 82, 139, 139},
aquamarine1 RBG = { 127, 255, 212},
aquamarine2 RBG = { 118, 238, 198},
aquamarine3 RBG = { 102, 205, 170},
aquamarine4 RBG = { 69, 139, 116},
DarkSeaGreen1 RBG = { 193, 255, 193},
DarkSeaGreen2 RBG = { 180, 238, 180},
DarkSeaGreen3 RBG = { 155, 205, 155},
DarkSeaGreen4 RBG = { 105, 139, 105},
SeaGreen1 RBG = { 84, 255, 159},
SeaGreen2 RBG = { 78, 238, 148},
SeaGreen3 RBG = { 67, 205, 128},
SeaGreen4 RBG = { 46, 139, 87},
PaleGreen1 RBG = { 154, 255, 154},
PaleGreen2 RBG = { 144, 238, 144},
PaleGreen3 RBG = { 124, 205, 124},
PaleGreen4 RBG = { 84, 139, 84},
SpringGreen1 RBG = { 0, 255, 127},
SpringGreen2 RBG = { 0, 238, 118},

SpringGreen3 RBG = { 0, 205, 102},
SpringGreen4 RBG = { 0, 139, 69},
green1 RBG = { 0, 255, 0},
green2 RBG = { 0, 238, 0},
green3 RBG = { 0, 205, 0},
green4 RBG = { 0, 139, 0},
chartreuse1 RBG = { 127, 255, 0},
chartreuse2 RBG = { 118, 238, 0},
chartreuse3 RBG = { 102, 205, 0},
chartreuse4 RBG = { 69, 139, 0},
OliveDrab1 RBG = { 192, 255, 62},
OliveDrab2 RBG = { 179, 238, 58},
OliveDrab3 RBG = { 154, 205, 50},
OliveDrab4 RBG = { 105, 139, 34},
DarkOliveGreen1 RBG = { 202, 255, 112},
DarkOliveGreen2 RBG = { 188, 238, 104},
DarkOliveGreen3 RBG = { 162, 205, 90},
DarkOliveGreen4 RBG = { 110, 139, 61},
khaki1 RBG = { 255, 246, 143},
khaki2 RBG = { 238, 230, 133},
khaki3 RBG = { 205, 198, 115},
khaki4 RBG = { 139, 134, 78},
LightGoldenrod1 RBG = { 255, 236, 139},
LightGoldenrod2 RBG = { 238, 220, 130},
LightGoldenrod3 RBG = { 205, 190, 112},
LightGoldenrod4 RBG = { 139, 129, 76},
LightYellow1 RBG = { 255, 255, 224},
LightYellow2 RBG = { 238, 238, 209},
LightYellow3 RBG = { 205, 205, 180},
LightYellow4 RBG = { 139, 139, 122},
yellow1 RBG = { 255, 255, 0},
yellow2 RBG = { 238, 238, 0},
yellow3 RBG = { 205, 205, 0},
yellow4 RBG = { 139, 139, 0},
gold1 RBG = { 255, 215, 0},

gold2 RBG = { 238, 201, 0},
gold3 RBG = { 205, 173, 0},
gold4 RBG = { 139, 117, 0},
goldenrod1 RBG = { 255, 193, 37},
goldenrod2 RBG = { 238, 180, 34},
goldenrod3 RBG = { 205, 155, 29},
goldenrod4 RBG = { 139, 105, 20},
DarkGoldenrod1 RBG = { 255, 185, 15},
DarkGoldenrod2 RBG = { 238, 173, 14},
DarkGoldenrod3 RBG = { 205, 149, 12},
DarkGoldenrod4 RBG = { 139, 101, 8},
RosyBrown1 RBG = { 255, 193, 193},
RosyBrown2 RBG = { 238, 180, 180},
RosyBrown3 RBG = { 205, 155, 155},
RosyBrown4 RBG = { 139, 105, 105},
IndianRed1 RBG = { 255, 106, 106},
IndianRed2 RBG = { 238, 99, 99},
IndianRed3 RBG = { 205, 85, 85},
IndianRed4 RBG = { 139, 58, 58},
sienna1 RBG = { 255, 130, 71},
sienna2 RBG = { 238, 121, 66},
sienna3 RBG = { 205, 104, 57},
sienna4 RBG = { 139, 71, 38},
burlywood1 RBG = { 255, 211, 155},
burlywood2 RBG = { 238, 197, 145},
burlywood3 RBG = { 205, 170, 125},
burlywood4 RBG = { 139, 115, 85},
wheat1 RBG = { 255, 231, 186},
wheat2 RBG = { 238, 216, 174},
wheat3 RBG = { 205, 186, 150},
wheat4 RBG = { 139, 126, 102},
tan1 RBG = { 255, 165, 79},
tan2 RBG = { 238, 154, 73},
tan3 RBG = { 205, 133, 63},
tan4 RBG = { 139, 90, 43},

chocolate1 RBG = { 255, 127, 36},
chocolate2 RBG = { 238, 118, 33},
chocolate3 RBG = { 205, 102, 29},
chocolate4 RBG = { 139, 69, 19},
firebrick1 RBG = { 255, 48, 48},
firebrick2 RBG = { 238, 44, 44},
firebrick3 RBG = { 205, 38, 38},
firebrick4 RBG = { 139, 26, 26},
brown1 RBG = { 255, 64, 64},
brown2 RBG = { 238, 59, 59},
brown3 RBG = { 205, 51, 51},
brown4 RBG = { 139, 35, 35},
salmon1 RBG = { 255, 140, 105},
salmon2 RBG = { 238, 130, 98},
salmon3 RBG = { 205, 112, 84},
salmon4 RBG = { 139, 76, 57},
LightSalmon1 RBG = { 255, 160, 122},
LightSalmon2 RBG = { 238, 149, 114},
LightSalmon3 RBG = { 205, 129, 98},
LightSalmon4 RBG = { 139, 87, 66},
orange1 RBG = { 255, 165, 0},
orange2 RBG = { 238, 154, 0},
orange3 RBG = { 205, 133, 0},
orange4 RBG = { 139, 90, 0},
DarkOrange1 RBG = { 255, 127, 0},
DarkOrange2 RBG = { 238, 118, 0},
DarkOrange3 RBG = { 205, 102, 0},
DarkOrange4 RBG = { 139, 69, 0},
coral1 RBG = { 255, 114, 86},
coral2 RBG = { 238, 106, 80},
coral3 RBG = { 205, 91, 69},
coral4 RBG = { 139, 62, 47},
tomato1 RBG = { 255, 99, 71},
tomato2 RBG = { 238, 92, 66},
tomato3 RBG = { 205, 79, 57},

tomato4 RBG = { 139, 54, 38},
OrangeRed1 RBG = { 255, 69, 0},
OrangeRed2 RBG = { 238, 64, 0},
OrangeRed3 RBG = { 205, 55, 0},
OrangeRed4 RBG = { 139, 37, 0},
red1 RBG = { 255, 0, 0},
red2 RBG = { 238, 0, 0},
red3 RBG = { 205, 0, 0},
red4 RBG = { 139, 0, 0},
DeepPink1 RBG = { 255, 20, 147},
DeepPink2 RBG = { 238, 18, 137},
DeepPink3 RBG = { 205, 16, 118},
DeepPink4 RBG = { 139, 10, 80},
HotPink1 RBG = { 255, 110, 180},
HotPink2 RBG = { 238, 106, 167},
HotPink3 RBG = { 205, 96, 144},
HotPink4 RBG = { 139, 58, 98},
pink1 RBG = { 255, 181, 197},
pink2 RBG = { 238, 169, 184},
pink3 RBG = { 205, 145, 158},
pink4 RBG = { 139, 99, 108},
LightPink1 RBG = { 255, 174, 185},
LightPink2 RBG = { 238, 162, 173},
LightPink3 RBG = { 205, 140, 149},
LightPink4 RBG = { 139, 95, 101},
PaleVioletRed1 RBG = { 255, 130, 171},
PaleVioletRed2 RBG = { 238, 121, 159},
PaleVioletRed3 RBG = { 205, 104, 137},
PaleVioletRed4 RBG = { 139, 71, 93},
maroon1 RBG = { 255, 52, 179},
maroon2 RBG = { 238, 48, 167},
maroon3 RBG = { 205, 41, 144},
maroon4 RBG = { 139, 28, 98},
VioletRed1 RBG = { 255, 62, 150},
VioletRed2 RBG = { 238, 58, 140},

VioletRed3 RBG = { 205, 50, 120},
VioletRed4 RBG = { 139, 34, 82},
magenta1 RBG = { 255, 0, 255},
magenta2 RBG = { 238, 0, 238},
magenta3 RBG = { 205, 0, 205},
magenta4 RBG = { 139, 0, 139},
orchid1 RBG = { 255, 131, 250},
orchid2 RBG = { 238, 122, 233},
orchid3 RBG = { 205, 105, 201},
orchid4 RBG = { 139, 71, 137},
plum1 RBG = { 255, 187, 255},
plum2 RBG = { 238, 174, 238},
plum3 RBG = { 205, 150, 205},
plum4 RBG = { 139, 102, 139},
MediumOrchid1 RBG = { 224, 102, 255},
MediumOrchid2 RBG = { 209, 95, 238},
MediumOrchid3 RBG = { 180, 82, 205},
MediumOrchid4 RBG = { 122, 55, 139},
DarkOrchid1 RBG = { 191, 62, 255},
DarkOrchid2 RBG = { 178, 58, 238},
DarkOrchid3 RBG = { 154, 50, 205},
DarkOrchid4 RBG = { 104, 34, 139},
purple1 RBG = { 155, 48, 255},
purple2 RBG = { 145, 44, 238},
purple3 RBG = { 125, 38, 205},
purple4 RBG = { 85, 26, 139},
MediumPurple1 RBG = { 171, 130, 255},
MediumPurple2 RBG = { 159, 121, 238},
MediumPurple3 RBG = { 137, 104, 205},
MediumPurple4 RBG = { 93, 71, 139},
thistle1 RBG = { 255, 225, 255},
thistle2 RBG = { 238, 210, 238},
thistle3 RBG = { 205, 181, 205},
thistle4 RBG = { 139, 123, 139},
gray0 RBG = { 0, 0, 0},

grey0 RBG = { 0, 0, 0},
gray1 RBG = { 3, 3, 3},
grey1 RBG = { 3, 3, 3},
gray2 RBG = { 5, 5, 5},
grey2 RBG = { 5, 5, 5},
gray3 RBG = { 8, 8, 8},
grey3 RBG = { 8, 8, 8},
gray4 RBG = { 10, 10, 10},
grey4 RBG = { 10, 10, 10},
gray5 RBG = { 13, 13, 13},
grey5 RBG = { 13, 13, 13},
gray6 RBG = { 15, 15, 15},
grey6 RBG = { 15, 15, 15},
gray7 RBG = { 18, 18, 18},
grey7 RBG = { 18, 18, 18},
gray8 RBG = { 20, 20, 20},
grey8 RBG = { 20, 20, 20},
gray9 RBG = { 23, 23, 23},
grey9 RBG = { 23, 23, 23},
gray10 RBG = { 26, 26, 26},
grey10 RBG = { 26, 26, 26},
gray11 RBG = { 28, 28, 28},
grey11 RBG = { 28, 28, 28},
gray12 RBG = { 31, 31, 31},
grey12 RBG = { 31, 31, 31},
gray13 RBG = { 33, 33, 33},
grey13 RBG = { 33, 33, 33},
gray14 RBG = { 36, 36, 36},
grey14 RBG = { 36, 36, 36},
gray15 RBG = { 38, 38, 38},
grey15 RBG = { 38, 38, 38},
gray16 RBG = { 41, 41, 41},
grey16 RBG = { 41, 41, 41},
gray17 RBG = { 43, 43, 43},
grey17 RBG = { 43, 43, 43},

gray18 RBG = { 46, 46, 46},
grey18 RBG = { 46, 46, 46},
gray19 RBG = { 48, 48, 48},
grey19 RBG = { 48, 48, 48},
gray20 RBG = { 51, 51, 51},
grey20 RBG = { 51, 51, 51},
gray21 RBG = { 54, 54, 54},
grey21 RBG = { 54, 54, 54},
gray22 RBG = { 56, 56, 56},
grey22 RBG = { 56, 56, 56},
gray23 RBG = { 59, 59, 59},
grey23 RBG = { 59, 59, 59},
gray24 RBG = { 61, 61, 61},
grey24 RBG = { 61, 61, 61},
gray25 RBG = { 64, 64, 64},
grey25 RBG = { 64, 64, 64},
gray26 RBG = { 66, 66, 66},
grey26 RBG = { 66, 66, 66},
gray27 RBG = { 69, 69, 69},
grey27 RBG = { 69, 69, 69},
gray28 RBG = { 71, 71, 71},
grey28 RBG = { 71, 71, 71},
gray29 RBG = { 74, 74, 74},
grey29 RBG = { 74, 74, 74},
gray30 RBG = { 77, 77, 77},
grey30 RBG = { 77, 77, 77},
gray31 RBG = { 79, 79, 79},
grey31 RBG = { 79, 79, 79},
gray32 RBG = { 82, 82, 82},
grey32 RBG = { 82, 82, 82},
gray33 RBG = { 84, 84, 84},
grey33 RBG = { 84, 84, 84},
gray34 RBG = { 87, 87, 87},
grey34 RBG = { 87, 87, 87},
gray35 RBG = { 89, 89, 89},

grey35 RBG = { 89, 89, 89},
gray36 RBG = { 92, 92, 92},
grey36 RBG = { 92, 92, 92},
gray37 RBG = { 94, 94, 94},
grey37 RBG = { 94, 94, 94},
gray38 RBG = { 97, 97, 97},
grey38 RBG = { 97, 97, 97},
gray39 RBG = { 99, 99, 99},
grey39 RBG = { 99, 99, 99},
gray40 RBG = { 102, 102, 102},
grey40 RBG = { 102, 102, 102},
gray41 RBG = { 105, 105, 105},
grey41 RBG = { 105, 105, 105},
gray42 RBG = { 107, 107, 107},
grey42 RBG = { 107, 107, 107},
gray43 RBG = { 110, 110, 110},
grey43 RBG = { 110, 110, 110},
gray44 RBG = { 112, 112, 112},
grey44 RBG = { 112, 112, 112},
gray45 RBG = { 115, 115, 115},
grey45 RBG = { 115, 115, 115},
gray46 RBG = { 117, 117, 117},
grey46 RBG = { 117, 117, 117},
gray47 RBG = { 120, 120, 120},
grey47 RBG = { 120, 120, 120},
gray48 RBG = { 122, 122, 122},
grey48 RBG = { 122, 122, 122},
gray49 RBG = { 125, 125, 125},
grey49 RBG = { 125, 125, 125},
gray50 RBG = { 127, 127, 127},
grey50 RBG = { 127, 127, 127},
gray51 RBG = { 130, 130, 130},
grey51 RBG = { 130, 130, 130},
gray52 RBG = { 133, 133, 133},
grey52 RBG = { 133, 133, 133},

gray53 RBG = { 135, 135, 135},
grey53 RBG = { 135, 135, 135},
gray54 RBG = { 138, 138, 138},
grey54 RBG = { 138, 138, 138},
gray55 RBG = { 140, 140, 140},
grey55 RBG = { 140, 140, 140},
gray56 RBG = { 143, 143, 143},
grey56 RBG = { 143, 143, 143},
gray57 RBG = { 145, 145, 145},
grey57 RBG = { 145, 145, 145},
gray58 RBG = { 148, 148, 148},
grey58 RBG = { 148, 148, 148},
gray59 RBG = { 150, 150, 150},
grey59 RBG = { 150, 150, 150},
gray60 RBG = { 153, 153, 153},
grey60 RBG = { 153, 153, 153},
gray61 RBG = { 156, 156, 156},
grey61 RBG = { 156, 156, 156},
gray62 RBG = { 158, 158, 158},
grey62 RBG = { 158, 158, 158},
gray63 RBG = { 161, 161, 161},
grey63 RBG = { 161, 161, 161},
gray64 RBG = { 163, 163, 163},
grey64 RBG = { 163, 163, 163},
gray65 RBG = { 166, 166, 166},
grey65 RBG = { 166, 166, 166},
gray66 RBG = { 168, 168, 168},
grey66 RBG = { 168, 168, 168},
gray67 RBG = { 171, 171, 171},
grey67 RBG = { 171, 171, 171},
gray68 RBG = { 173, 173, 173},
grey68 RBG = { 173, 173, 173},
gray69 RBG = { 176, 176, 176},
grey69 RBG = { 176, 176, 176},
gray70 RBG = { 179, 179, 179},

grey70 RBG = { 179, 179, 179},
gray71 RBG = { 181, 181, 181},
grey71 RBG = { 181, 181, 181},
gray72 RBG = { 184, 184, 184},
grey72 RBG = { 184, 184, 184},
gray73 RBG = { 186, 186, 186},
grey73 RBG = { 186, 186, 186},
gray74 RBG = { 189, 189, 189},
grey74 RBG = { 189, 189, 189},
gray75 RBG = { 191, 191, 191},
grey75 RBG = { 191, 191, 191},
gray76 RBG = { 194, 194, 194},
grey76 RBG = { 194, 194, 194},
gray77 RBG = { 196, 196, 196},
grey77 RBG = { 196, 196, 196},
gray78 RBG = { 199, 199, 199},
grey78 RBG = { 199, 199, 199},
gray79 RBG = { 201, 201, 201},
grey79 RBG = { 201, 201, 201},
gray80 RBG = { 204, 204, 204},
grey80 RBG = { 204, 204, 204},
gray81 RBG = { 207, 207, 207},
grey81 RBG = { 207, 207, 207},
gray82 RBG = { 209, 209, 209},
grey82 RBG = { 209, 209, 209},
gray83 RBG = { 212, 212, 212},
grey83 RBG = { 212, 212, 212},
gray84 RBG = { 214, 214, 214},
grey84 RBG = { 214, 214, 214},
gray85 RBG = { 217, 217, 217},
grey85 RBG = { 217, 217, 217},
gray86 RBG = { 219, 219, 219},
grey86 RBG = { 219, 219, 219},
gray87 RBG = { 222, 222, 222},
grey87 RBG = { 222, 222, 222},

gray88 RBG = { 224, 224, 224},
grey88 RBG = { 224, 224, 224},
gray89 RBG = { 227, 227, 227},
grey89 RBG = { 227, 227, 227},
gray90 RBG = { 229, 229, 229},
grey90 RBG = { 229, 229, 229},
gray91 RBG = { 232, 232, 232},
grey91 RBG = { 232, 232, 232},
gray92 RBG = { 235, 235, 235},
grey92 RBG = { 235, 235, 235},
gray93 RBG = { 237, 237, 237},
grey93 RBG = { 237, 237, 237},
gray94 RBG = { 240, 240, 240},
grey94 RBG = { 240, 240, 240},
gray95 RBG = { 242, 242, 242},
grey95 RBG = { 242, 242, 242},
gray96 RBG = { 245, 245, 245},
grey96 RBG = { 245, 245, 245},
gray97 RBG = { 247, 247, 247},
grey97 RBG = { 247, 247, 247},
gray98 RBG = { 250, 250, 250},
grey98 RBG = { 250, 250, 250},
gray99 RBG = { 252, 252, 252},
grey99 RBG = { 252, 252, 252},
gray100 RBG = { 255, 255, 255},
grey100 RBG = { 255, 255, 255},
dark grey RBG = { 169, 169, 169},
DarkGrey RBG = { 169, 169, 169},
dark gray RBG = { 169, 169, 169},
DarkGray RBG = { 169, 169, 169},
dark blue RBG = { 0, 0, 139},
DarkBlue RBG = { 0, 0, 139},
dark cyan RBG = { 0, 139, 139},
DarkCyan RBG = { 0, 139, 139},
dark magenta RBG = { 139, 0, 139},

DarkMagenta RBG = { 139, 0, 139},
dark red RBG = { 139, 0, 0},
DarkRed RBG = { 139, 0, 0},
light green RBG = { 144, 238, 144},
LightGreen RBG = { 144, 238, 144}

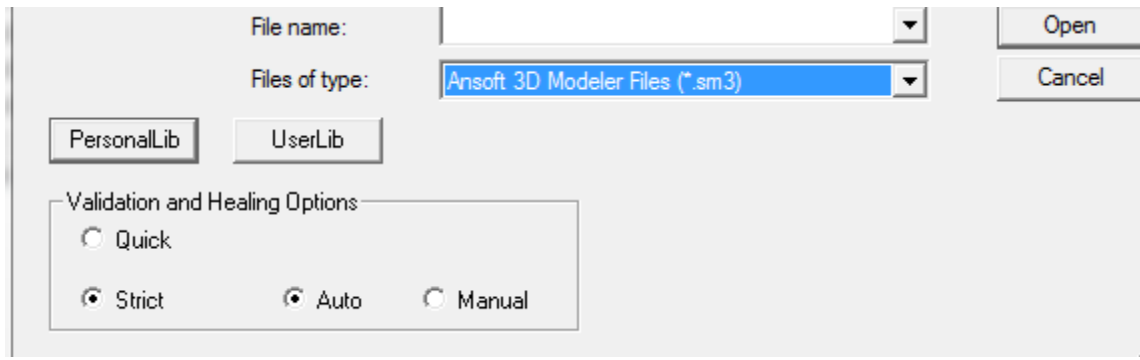
Importing 3D Model Files

You can read 3D model files directly into the active **3D Modeler** window:

Note If you import a file into an active **3D Modeler** window that contains an existing model, the file will be added to the existing model; it will not replace it.

To import a 3D model file:

1. Click **Modeler**>**Import**.
   The **Import File** dialog box appears.
2. Select the file type you want from the **Files of type** pull-down list.
3. Select any import options available for the selected file type.
   - Some file types permit you to specify Validation and Healing options. In these cases, you can choose between Quick for faster import, or for Strict healing, which you can specify as Auto or Manual. See the table below and **Healing an Imported Object**.



   - For ProE files, you can choose check to enable the **Import Free Surfaces** option. This imports such surfaces as well as parts.
   - For Nastran and STL files, you can set the Model Resolution Length in Model Units, or accept the auto setting.
   - For STEP and IEGS files, you can specify a Stitch tolerance and units. The default value (auto) comes from the **Healing** dialog **Options** tab with Manual Healing selected.
4. Use the file browser to find the file you want to import.
5. Select the 3D model file you want to import or enter the name of the file in the **File Name** box.
6. Click **Open**.

The file is imported into the active **Modeler** window.

If you selected Validation and Healing options **Strict** with the **Manual** option selected for the import, then the **Healing Options** dialog box opens, allowing you to set parameters for the heal operation.

For tips on dealing with very complex models, see Technical Notes: Handling Complicated Models.

Note While objects created in HFSS or HFSS-IE can always be classed in the history tree as either a solid, sheet, or wire some imported objects may have mixture of these. HFSS or HFSS-IE places such objects in an **Unclassified** folder in the history tree.

Extension	Contents
.sat	ACIS 23.2 geometry solid model files.[1]
.sm3	Ansoft 3D modeler files ACIS 1.0 to ACIS 23.2[1]
.sm2	Ansoft 2D modeler files
.sld	Ansoft legacy 3D model files
.anstGeom	Ansoft Geometry File supporting UDMs and CAD integration with the Workbench.
.dxf, .dwg **.tech**	AutoCAD Drawing Interchange Format files from version 2.5 through 2009. The .tech file is an ASCII file that contains layer names, units, color, elevation, thickness, and material information in a tab delimited format. See Importing DXF and DWG Format Files.
.ipt, iam	Autodesk Inventor V6-V2013
.model, **.CATPart** **.CATProduct**	Catia R4/R5 models.[1] .model - CATIA 4.1.9 to 4.2.4 .CATPart, .CATProduct - CATIA R5-R23 (V5- V-6 R2013), (Windows only)
.gds	GDSII files.
.iges, .igs	Industry standard Initial Graphics Exchange Specification (IGES) files versions up to 5.3.[1]

Extension	Contents
.nas	NASTRAN format files.[2]
.x_t, .x_b	Parasolid Files 10-25.0.155.[1]
.prt*, .asm*	Pro/E model files. For ProE models, the import dialog lets enable Import Free Surfaces, which will include such surfaces as well as parts. .Pro/E 16 to Wildfire 5.0, Creo 2.0
.sab	Standard ACIS binary.
.sldprt, .sldasm	Solidworks 98-2013
.step, .stp	Industry standard AP203 STEP files and AP214 (geometry only).[1]
.stl	Stereolithography format files.[2]
.prt	Unigraphics file 11 to NX8.5[1]

1. Automatic or Manual Healing available if desired. See Healing an Imported Object.

2. Defeaturing based on Model Resolution Length. Select Auto, None, or enter a numeric value directly in the entry box.

Related Topics

Importing 2D Model Files

Importing DXF and DWG Format Files.

Exporting 3D Model Files

Technical Notes: Handling Complicated Models

Importing DXF and DWG Format Files

You can import AutoCAD versions 2.5 through 2011. The entities are imported as 2 dimensional (not 3D). The types of entities imported are:

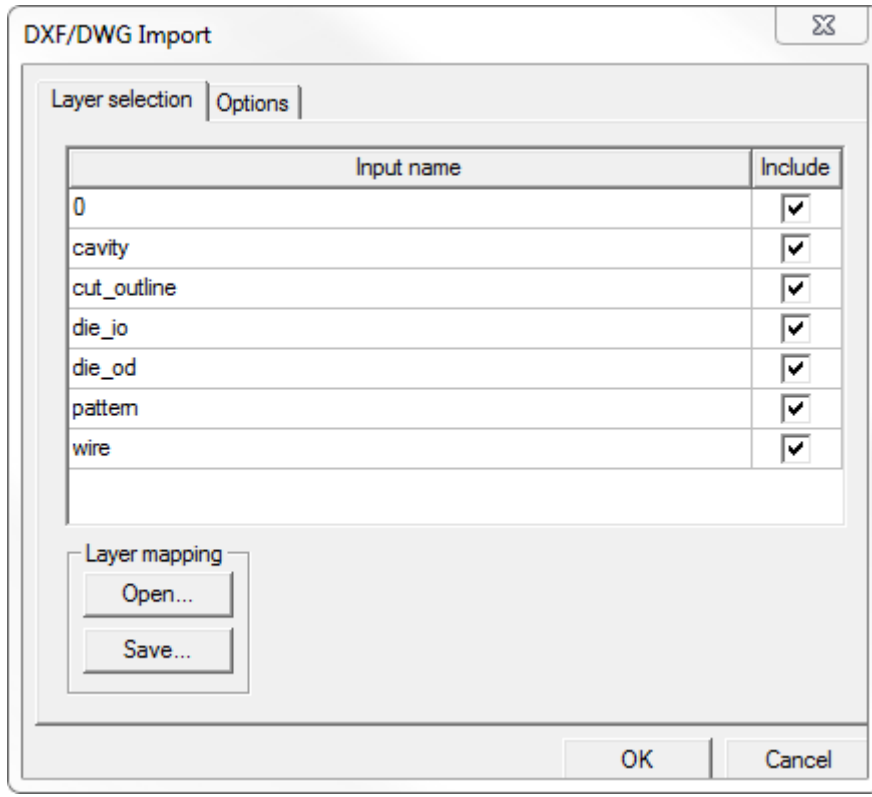
- 2D Polyline, Polyline, and Line
- Arc
- Circle
- Ellipse
- Solid
- Block

To import a **.dxf** or **.dwg** model file (which may use an associated .tech file):

1. Click **Modeler**>**Import**.

   The **Import File** dialog box appears.
2. Select **AutoCAD Files (*.dxf;*.dwg)** from the **Files of type** pull-down list.
3. Use the file browser to find the file you want to import.
4. Select the **.dxf**/**.dwg** model file you want to import.
5. Click **Open**.

Initially, the **DWG/DXF Import** dialog opens with the **Layer Selection** tab is displayed:



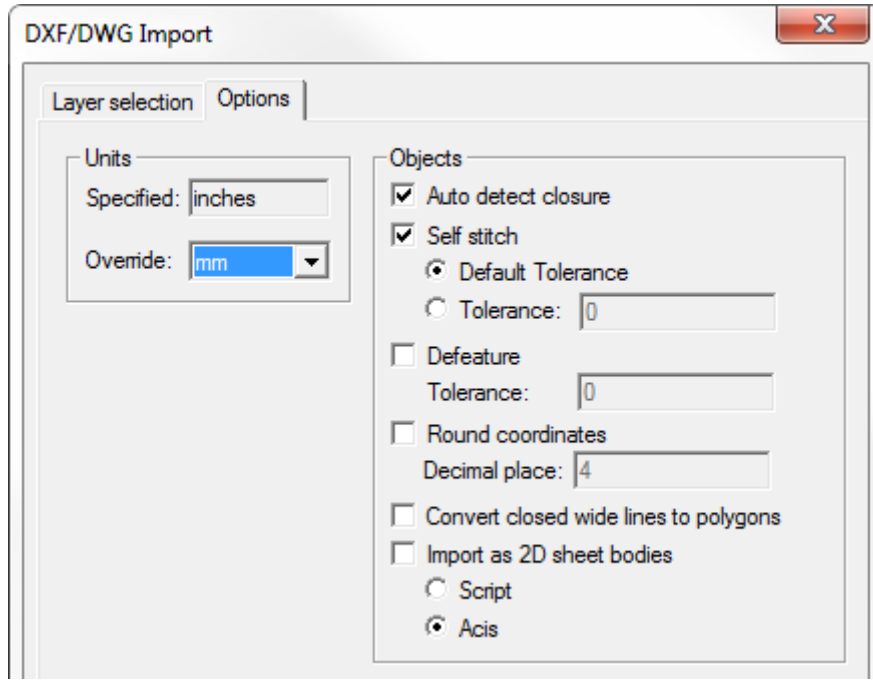
The **Input Layer Name** field shows the name of the layer in the DXF/DWG file (not editable)

6. Use the **Include** check boxes to specify which layers to import from the selected file.
7. You can use the **Open** button for a browser window to locate a tech file. The tech file is a plain text file that includes units, layer names, color, elevation, and thickness information.

```
units um
//Layer_Name    Color    Elevation  Thickness
BOTTOMLAYER     purple   0          200
```

```
MIDLAYER        green     500        200
TOPLAYER        blue      1000       200
```

8. Click the **Options** tab:



9. Use the **Override** pulldown to select the layout units for the imported file (default is **mm**).
10. Use the Objects check boxes to fine-tune the import:
    - **Auto-detect closure** causes polylines to be checked to see whether or not they are closed. If a polyline is closed, the modeler creates a polygon in the design.
    - **Self-stitch** causes multiple straight line segments to be joined to form polylines. If the resulting polyline is closed, a polygon is created in the modeler.

      Default Tolerance/Tolerance specifies whether you can specify a self stitch Tolerance value. If particular features in a model are outside of a normal tolerance allowance, you can set a Tolerance for importing that specific model.
    - **De-feature tolerance** removes certain small features in the imported geometry to reduce complexity. The features that are removed include: multiple points placed within the specified distance; thin or narrow regions ("thins" and "spikes"); and extraneous points along straight line segments.
    - **Round coordinates** to **Decimal place** rounds all imported data to the specified number of decimal points.
    - **Convert closed wide lines to polygons** imports wide polylines as polygons. You have more flexibility to change the shape of such an object when it is imported as a polygon.

- **Import as 2D sheet bodies** causes imported objects to be organized in terms of 2D sheets.

11. For Import method, select Script or Acis.
12. When you have completed selections on all tabs, click **OK** on any tab.

    The file is imported into the active **Layout** window.

Importing Solution Data

1. Click **HFSS>Results>Import Solutions**.

   The **Imported Data** dialog box appears.
2. Click **Import Solution**.

   The **S Parameter Import** dialog box appears.
3. In the **File Name** text box, type the name of the solution file you want to import or click **Browse** and use the file browser to locate the file.
4. Selecting the file with the file browser loads it, but note that the file has not been imported yet.
5. Optionally, type a new name in the **Source Name** box or accept the default name.
6. Click the solutions you want to import in the **Available Solutions** list, and then click **Import**.

   You return to the **Imported Data** dialog box.
7. Click the solution data you want to import, and then click **OK**.

Related Topics

Viewing Matrix Data

Importing Data Tables

You can import data table files that contain data in the following formats:

- Tab-separated. HFSS will recognize complex data if the values are separated by a comma (e.g. *real, imaginary*).
- Comma-separated. HFSS will recognize complex data if the values are separated by a space (e.g. *real imaginary*).

1. Click **HFSS** >**Results**>**Import Solutions**.
   - Alternatively, right-click **Results** in the project tree and then click **Import Solutions** on the shortcut menu.

   The **Imported Data** dialog box appears.
2. Click **Import Table**.

   The **Table Import** dialog box appears.
3. In the **File Name** text box, type the name of the data table file you want to import or click **Browse** and use the file browser to locate the file.
4. If the data in the table is complex, select the format — real/ imaginary, or magnitude/ phase — in which to import the data.

If the data is simple, this option will be ignored.

5. Click **Load File**. Note that the file has not been imported yet.
6. Optionally, type a new name in the **Source Name** box that indicates the origin or the data table, or accept the default name.
7. Optionally, type a new name in the **Table Name** box that describes the data in the table, or accept the default name
8. In the **All Columns** list, the headings of each column in the data file are listed. Optionally, specify a new name for a column heading by doing the following:
   a. In the **All Columns** list, click the heading you want to change. The heading appears in the **Column Name** box.
   b. Type a new name in the **Column Name** box, and then click **Set Column Name**.

      The heading is changed to the new name in every place it appears in the **Imported Data** dialog box.
9. In the **Independent Data Columns** list, the first heading in the data table file is listed by default. In the **Dependent Data Columns** list, the second and subsequent headings in the data table file are listed by default. Optionally, click a heading name and then click an arrow button to move it from one column to another.
10. If the data in the **Dependent Data Columns** list contains matrix data, select **Matrix Data**. If it contains field data, select **Field Data**.
11. Click **Import**.

    You return to the **Imported Data** dialog box.
12. Click the data you want to import in the **Current Imports** list, and then click **OK**.

    The solution data is now available for post processing.

Related Topics

Adding Datasets

Importing HFSS Plot Data

Import Solutions can also import HFSS plot data.

1. On the HFSS menu click **Results**>**Import Solutions**.

   The **Imported Data** dialog box appears.
2. Click the Import Plot Data button

   The **Read Plot Data** file dialog opens.
3. Use the file browser to select the plot data file (*.dat) to open.

   You can choose to specify the file as Read Only.
4. Click the Open button to import the file.

   The imported files are listed in the **Imported Data** dialog.

Related Topics

Importing Plot Data.

Importing Plot Data

The **Report2D> Import Data** command lets you import plot data from comma delimited files (.csv) tab delimited files (.tab) or Ansoft Plot Data files (*.dat). You need to have a report open for the **Report2D** menu to appear.

1. On the HFSS menu click **Report2D>Import Data**.

   This displays a file browser window.
2. Use the Look In feature, or the icons to navigate to the file location.
3. Specify the file name in the file name field, or select the file from those listed in the current directory.
4. The file format field contains a drop-down menu listing the formats you can import. These include comma delimited files (.csv) tab delimited files (.tab) or Ansoft Plot Data files (*.dat).
5. Click **Open** to import the file into the currently open Report.

   The imported traces appear in the Project tree under the current report.

Related Topics

Importing HFSS Plot Data

Inserting a Documentation File

You may want to add a documentation file to the project tree.

1. Click **Project>Insert Documentation** File.

   This opens a file browser dialog that lets you navigate your file system.

2. Selecting the file and click OK.

   This places the documentation file in the project tree.

Related Topics

Printing (Image of Active window)

Saving Project Notes

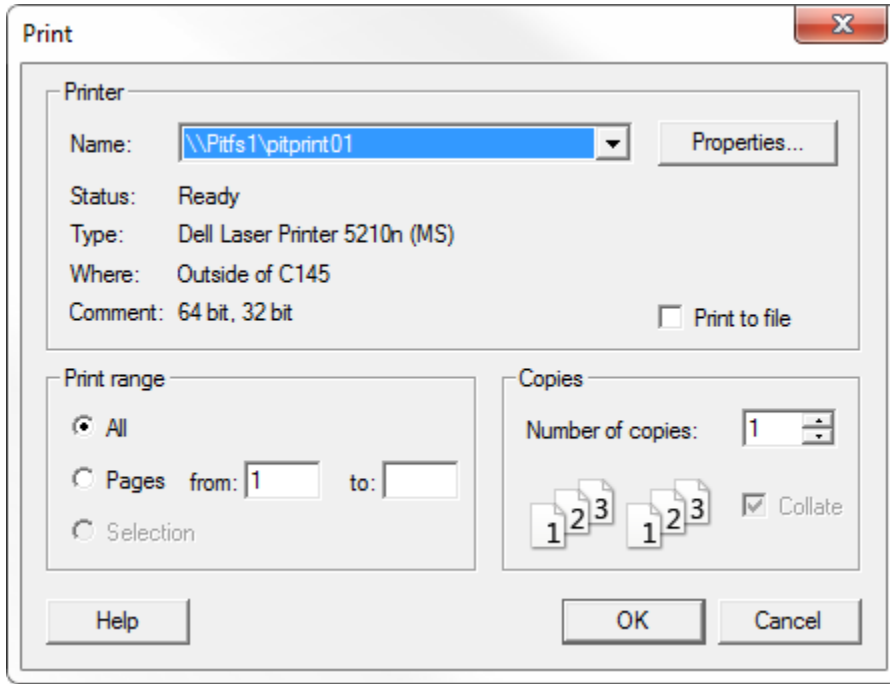
Printing

The printing commands enable you to send an image of the active window to the printer.

To print the project:

1. Click **File**>**Print**  .

   A dialog box similar to the following one appears:



2. You can change the printer (if other printer names are listed on the drop down), set the print range, number of copies, or use the check box to Print to file.
3. Do one of the following:
   - Click **OK** to print the project.
   - Click **Cancel** to dismiss the window without printing.
   - Click **Setup** to define printer settings.

You can also access the printer properties by clicking **Printer** in the **Page Setup** dialog box.

Related Topics

Getting Help

Previewing the Printout

Changing the Page Setup

Previewing the Printout

To preview how the page will look when printed:

1. Click **File>Print Preview**.
   The preview window appears.
2. To print the project after seeing the preview, click the **Print** button.
   The **Print** dialog box appears.
3. To navigate through the preview, click the **Next Page**, **Prev Page**, and/or **Two Page** buttons.
4. To zoom in or out on the preview, click the **Zoom In** or **Zoom Out** button.

To close the preview and return to your project, click **Close**.

Related Topics

Printing

Changing the Page Setup

To set or change the page setup:

1. Click **File>Page Setup**.
   The **Page Setup** dialog box appears.
2. Under **Paper**, select a **Size** and **Source** for the paper.
3. Under **Orientation**, select either **Portrait** or **Landscape**.
4. Under **Margins**, change the values as desired in the **Left**, **Right**, **Top**, and **Bottom** text boxes.
5. Click **OK**.

You can also access the printer properties by clicking **Printer** in the **Page Setup** dialog box.

Saving Project Notes

You can save notes about a project, such as its creation date and a description of the device being modeled. This is useful for keeping a running log on the project.

To add notes to a project:

1. Click **HFSS or HFSS-IE>Edit Notes**  .

   The **Design Notes** window appears.
2. Click in the window and type your notes.
3. Click **OK** to save the notes with the current project.

To edit existing project notes:

- Double-click the **Notes** icon in the project tree.

  The **Design Notes** window appears, in which you can edit the project's notes.

To delete the existing notes for a design:

1. Select the icon **Notes** in the project tree, click **Edit>Delete**.
2. Right-click the icon **Notes** in the project tree, click **Delete** from the shortcut menu.

The Notes icon is removed from the project tree.

Note Notes are used to document aspects of designs only. For project level documentation, you can insert a documentation file into a project with the Project>Insert Documentation Files command.

Related Topics

Printing (Image of Active window)

Inserting a Documentation File

Setting Options in HFSS

You can set the following options from the Desktop:

- General options, such as project options, units settings, and remote analysis options.
- HPC and Analysis Options, such as active configuration per design type, queuing, distributed memory vendor, HPC licensing, and whether to enable GPU for transient solves.
- HFSS-specific options, such as default solution mode, processor and RAM settings.
- Fields Reporter options, such as field overlay and phase animation settings.
- HFSS-IE Options, such as whether to duplicate boundaries, save and delete options, or processor and RAM settings.
- Report2D options, such as fonts, labels, line styles, and colors.
- Report Setup Options, including advanced mode editing, the number of significant digits to display, and drag and drop behavior.
- Modeler options, such as cloning options, display colors and render settings, snap modes and mouse sensitivity.

The Tools>Options>Export Options Files command writes xml files containing the Options settings at all levels to the specified directory. The **Tools>Options>Export Options** feature is intended to make it easier for different users to use ANSYS Electromagnetics tools installed on shared directories or network drives. The Example Uses for Export Options Features section outlines some use cases enabled by this feature.

Related Topics

Setting Options via Configuration Files

Example Uses for Export Options Features

User Options and the Update Registry Tool

Batchoptions Command Line Examples

Setting General Options

To set general options in HFSS:

1. Click **Tools>Options>General Options**.
   The **General Options** window appears, displaying six available tabs:
   - Project Options
   - Default Units
   - Remote Analysis Options
   - WebUpdate Options
   - Miscellaneous Options
   - Desktop Performance
2. Click each tab, and make the desired selections.
3. Click **OK**.

General Options: Project Options Tab

These options are set on the **Project Options** tab of the **General Options** dialog box.

1. To auto-save your project, do the following in the Autosave section:
   a. Select the **Do Autosave** check box.
   b. Enter the number of edits after which to save in the **Autosave interval** text box. The default is **10**.
2. For each directory, Project, Syslib, UserLib, and PersonalLib, enter a path in the **Directory** text box, or click the **...** button to find and select the desired directory. If you modify SysLib, UserLib or PersonalLib paths through **Tools>General** options, the User Defined Primitives menu should reflect those changes on next startup or on **Draw>User Defined Primitive>Update Menu**.
3. If you want to enter a directory path in the **Temp Directory** text box, you must check Override to enable the path text field and the **...** button for finding and selecting the desired directory.
4. To reset the library directories to the default, click **Reset Library Directory**.
5. Select or clear the **Expand Project Tree on Insert** check box.
6. For **When creating a new project**, select a radio button to either **Insert a design of type HFSS** or **Don't insert a design**.
7. You can choose to specify a warning to be given when available disk space is less than a given amount in M bytes.

General Options: Default Units Tab

These options are set on the **Default Units** tab of the **General Options** dialog box.

Select the desired units from each of the following pull-down lists:

- Length
- Angle
- Time
- Temperature
- Torque
- Magnetic Induction
- Frequency
- Power
- Voltage
- Current
- Speed
- Weight
- Resistance
- Inductance
- Capacitance

- Force
- Angular Speed
- Magnetic field strength
- Pressure

General Options: Remote Analysis Options

To launch all analyses as a specific user, rather than the current user, do the following in the **Remote Analysis Options** tab of the General Options. (Note: If any of the remote machines are Unix-based, you must specify the current user.)

- The ANSYS Service should be running on this port for all distributed machines. To change the Ansoft Service Port, click the **Change** button to display an editable dialog.
- The **Send analysis request as** option selection can be **Service User** or **Specified User**. Selecting Specified User enables the fields for user name, password, and domain information.

You can also **Disable access by remote machines,** by using the checkbox.

If a local machine has multiple IP addresses, you can also specify whether to Use Default or a Specified IP address.

The Desktop-Engine connection area lets you specify the Desktop IP address referenced. This option applies only when the local machine has multiple IP addresses and one of these is preferred for communication.

- Use Default
- Specified

Related Topics

Remote Analysis

General Options: WebUpdate Options Tab

These options are set on the **WebUpdate Options** tab of the **General Options** dialog box.

Select one of the following from the **Automatically check for updates every** pull-down list:

- Never
- 30 days
- 120 days
- 180 days

The last time the software was updated, as well as the last attempt, are displayed in the following two fields:

- **Last update date**
- **Last update attempt date**

General Options: Miscellaneous Options Tab

These options are set on the **Miscellaneous** tab of the **General Options** dialog box.

General Options:

- Select or clear the **Show Message Window on new messages** check box.
- Select or clear the **Ensure that new messages are visible in the Message Window Tree** check box.
- Select or clear the **Show Progress Window when starting a simulation** check box.
- Select or clear the **Update reports on file open** check box.

Project Tree Visualization Options

- Select or clear the **Emphasize active command context** (menu and toolbars).
- Select or clear **Change icon when selection does not match active window**. Single click the icon to launch corresponding window. The application title bar shows the active window

The ANSYS Workbench Application **Path** lets you specify a path to an installation, if you have one. This path can be used by the HFSS Optimetrics feature for connecting to the Design Xplorer.

MATLAB Optimization

If you have an installation of MATLAB installed you can use it as an Optimizer. This MATLAB path setting must to point to the version of MATLAB to be used for performing the optimization. .

Note The platform (32/64 bit of the specified version of MATLAB must match the platform of this application)

Show welcome message at startup.

Check this to display a welcome message.

General Options: Desktop Performance Tab

These options are set on the **Desktop Performance** tab of the **General Options** dialog box.

Report Update Options for Design Type:

- The **Design type** is HFSS or HFSS-IE.
- Select or clear **Dynamically update reports and field overlays during edits**.

  If selected, report plots and overlays update dynamically.
- **Dynamically update postprocessing data for new solutions**.

  Updating numerous reports may a significant amount of time. Updating reports during the analysis process can impact the overall time to solution. You may want to vary the times when your reports get updated relative to the impact on overall solve time.

  Five options exist for updating reports during solutions:

  - Automatically - the default. It means update most things immediately.
    For "AdaptivePass" plot context, plots are updated at the end of each solution pass. For "LastAdaptive" or "Transient" the plot is updated at the end of the transient or adaptive solution.
    This option balances report and field plot updating with solution time. For example, reports may be updated after each adaptive pass but field plots will not be updated until the solution is complete.

- Immediately - update reports and plots as soon as data comes from the solver.
  This option will have the greatest impact on the overall solution time but will have the most rapid updating of reports and field plots. Caution should be used in selecting this option. Some types of reports and field plots may take a long time to update, especially as the mesh size increases.
- Never - only manual intervention updates reports.
  This option will prevent updates from impacting the solution time.
- On Completion - as with Never, but a single update is done when the solve completes.
  Note: Reports that are updated on completion are done after the solve has been completed. The time for that update is not included in the solve profile.
- After Each Variation - when performing an Optimetric or parametric analysis, all reports are updated after analysis of each variation has been completed.

The Animation setting lets you set a limit to **Stop computing animation frames when available memory is less than** a value in megabytes. The default is 100.

Desktop Pre/Post Processing settings:

- Number of Processors

  Set the number. The default is the number of logical processors on the machine/2. This option is common for both pre-processing and post-processing. This option only affects pre/post processing in desktop (not solve or simulation)

Currently following pre-processing algorithms can take advantage of multiple processors

- Visualization/faceting of 3D models for 3D products
- Model validation for 3D products
- Auto net identification for Q3D

Setting HPC and Analysis Options

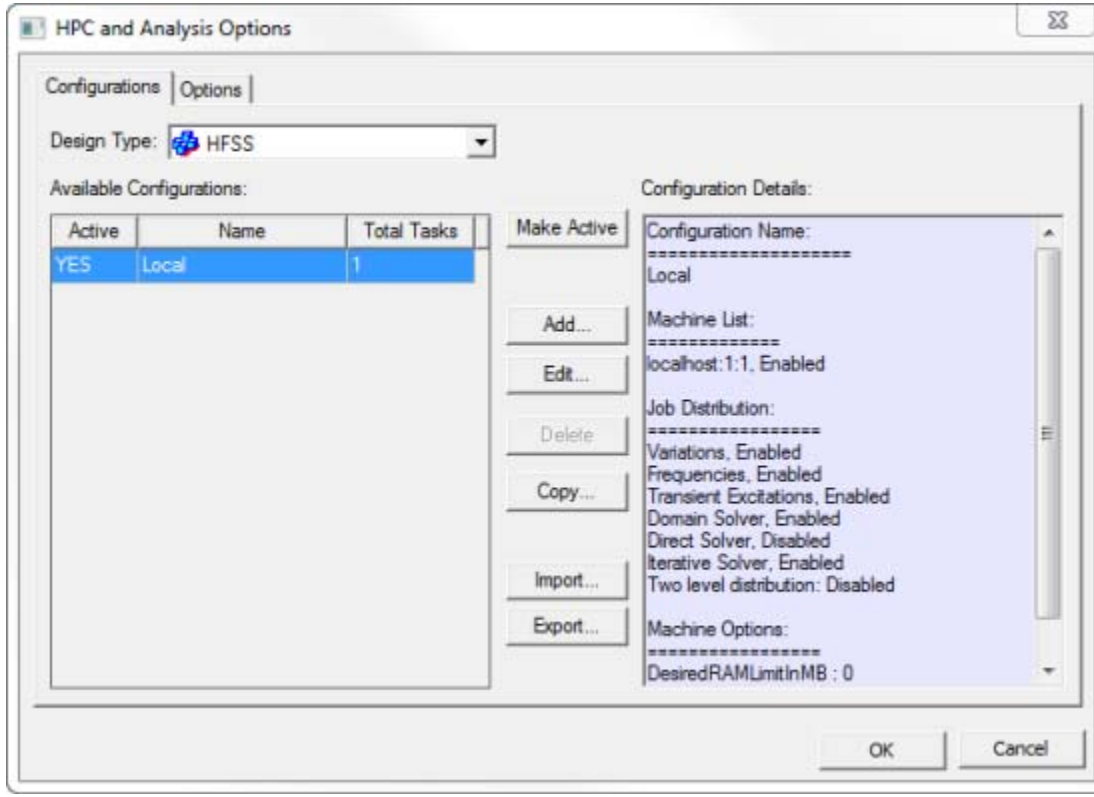
All analysis parameters are accessed via a single dialog. The machine list and options settings have been integrated into analysis configurations. The default configuration is for solving on a single, local machine, You can create many analysis configurations for remote and distributed solutions, and switch between them depending on the job being solved. Multiprocessing has been integrated into the machine lists.

To set the HPC and Analysis Options:

1. Click **Tools>Options>HPC and Analysis Options**. You can also access the dialog by using the **Tools and Analysis Options** icon on toolbar, or the button on the **Solution Setup General**

tab.

The **HPC and Analysis Options** dialog appears, displaying two tabs.



Configurations tab

In the **Configurations** tab, you can select the Design type, and select from a list of available configurations. Selecting a Design Type displays a list of the Available configurations for that type. Selecting a configuration from the list displays the details of that configuration in the text fields. A Name can describe the use for which a configuration has been defined, The Total Tasks column shows the number of tasks that the analysis configuration can execute.

Design Type

You define configurations for HFSS and HFSS-IE design types separately, selecting from the drop down menu.. If you want to use similar analysis parameters for a different design type, you must create a separate analysis configuration for that design type. The active configuration is used when solving an analysis for that design type.

Available Configurations List

From the lists of Available configurations for each design type, you left click from the configuration list to select in then click **Make Active**. Note that the active configuration will be indicated with a YES in the Active column.

Add.. Button

This launches a dialog to create a new analysis configuration.

Edit.. Button

This launches a dialog to edit the currently selected analysis configuration.

Delete Button

This deletes the currently selected analysis configurations.

Note You cannot delete the Local configuration.

Copy... Button

This creates a new analysis configuration, and launches a dialog to edit it. If the dialog is canceled, the new analysis configuration is not created.

Import... Button

This allows the user to import an .acf file to create an analysis configuration.

Note Importing analysis configurations always adds the imported analysis configurations to the current design type. Also, if there is a name conflict between an imported analysis configuration and an existing analysis configuration, the imported configuration is renamed and you are notified.

Export... Button

This allows the user to export the selected analysis configurations to an .acf file. Users can them import the configurations into a different design type, or import them on a different machine.

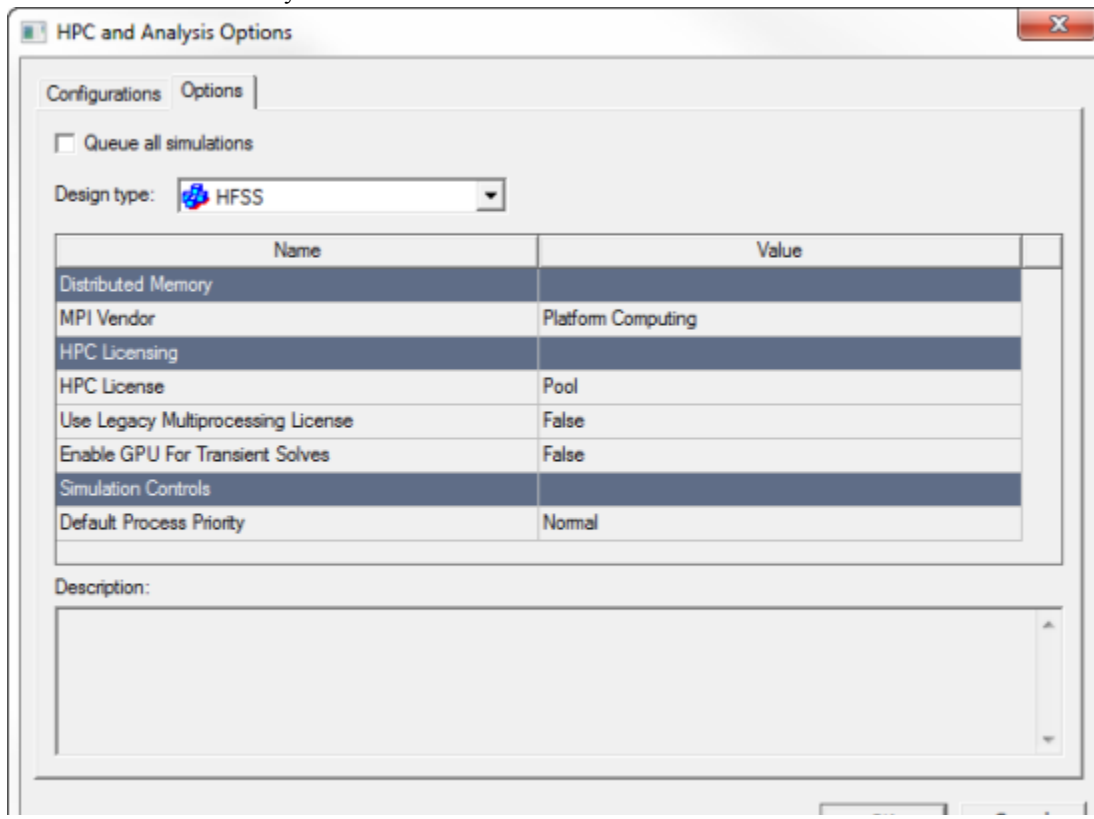
Options Tab

The Options tab in the **HPC and Analysis Options** dialog contains design type specific options. These options are not part of an analysis configuration, instead they are always in effect for the given design type when the following is true:

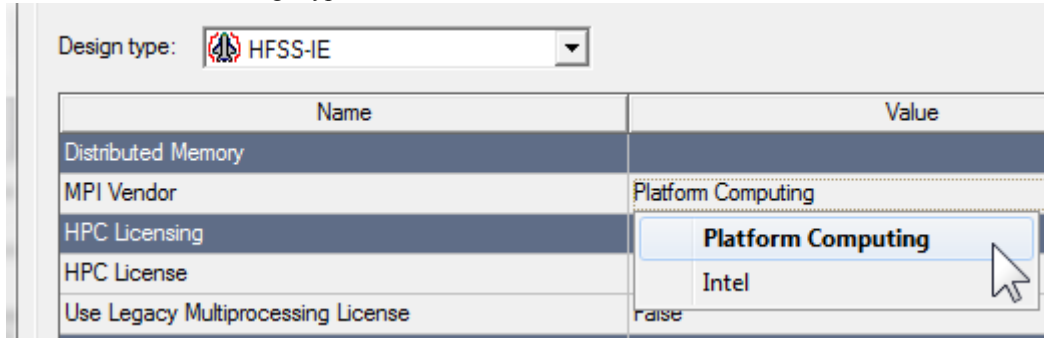
- A design of the matching design type is being solved
- You have not specified corresponding overriding batch options on the command line.

In the **Options** tab you can enable queuing, set the design type, specify the Distributed Memory vendor (that is, MPI for Message Passing Interface), set licensing options and enable GPU for Transient Solves. Solving on a single Windows machine does not require MPI installation. And users running on Linux do not need to install MPI manually. You can also set the Default

Process Priority..



2. For Distributed Memory Options, use the drop down menu to select the MPI Vendor for the selected Design type.



The HFSS and HFSS-IE solvers use the industry standard Message Passing Interface ("MPI") and can perform solutions that distribute memory use across machines in a cluster or network. Memory used by the MPI-enabled HFSS solver is therefore limited by the set of machines that are available rather than the shared memory available on any single machine. This allows you to simulate larger structures than before and to optimally reconfigure the cluster of machines

for the problem at hand. For solving on a single machine, MPI is not required, nor does it provide an advantage.

To use the distributed memory solution in HFSS or HFSS-IE you will need to install MPI software from one of the supported third party vendors on all the machines you intend to use.

You may need to set passwords depending on the MPI vendor for authentication on the machines. Settings within HFSS and HFSS-IE turn on distributed memory solutions and define the list of machines you intend to use. Detailed instructions about how to get distributed memory HFSS solutions up and running are outlined in Distributed Memory Solutions with HFSS.

3. For Linux authentication, you can specify the Remote Spawn Command as RSH or SSH (the default).
4. Select one of the following HPC Licensing Options. The selection determines whether multiprocessing is enabled by Pool or Pack licensing.
   - Pool - HPC Licenses will be used for distribution.
   - Pack - HPC Pack licenses will be used for distribution.

   Solution setups with domains always use HPC licensing. Even though domains are distributed and may use multiprocessing, they do not use distributed solve or multiprocessing licenses - all of this is included in the HPC licensing when solving domains.

   HPC licensing is based on counting the total number of cores in a simulation. For example, when solving a distributed solution across 10 nodes, with multiprocessing enabling 2 cores per node, the total number of cores is 2x10, or twenty cores.

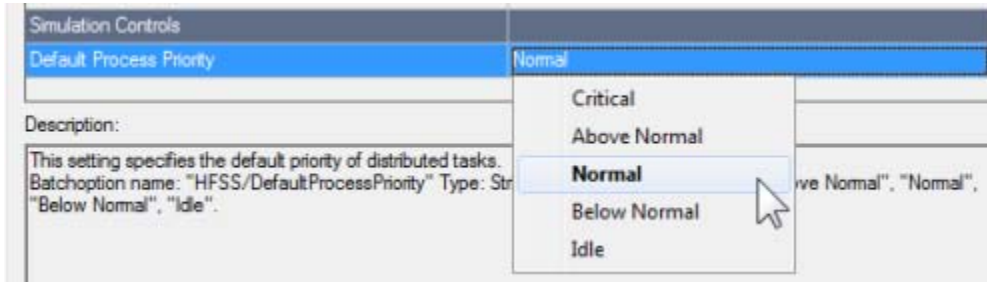
   The HPC License Type determines the type and number of licenses that will be checked out for a given number of cores. For the pool type, one license will be checked out for each core in use. So a simulation with twenty cores would require twenty HPC licenses. For the Pack type, a single pack enables eight cores, and each additional pack enables four times as many cores. So a simulation with twenty cores would require two "Pack" licenses, enabling up to 8x4, or 32, cores.

   If you are solving an HFSS Transient problem, the solver will not checkout hfss_solve (regular solve license). We will only checkout a hfss_transient_solve license. For HFSS Transient problems and HFSS array problems, we will always use HPC licenses for multiprocessing (even if you have NOT chosen the **Use HPC licenses for multiprocessing and distributed frequency points** option).

   For HFSS-IE problems an hfssie_solve license is required. For HFSS problems with radiation boundaries and Use IE Formulation checked, a hfssie_solve license is required in addition to the hfss_solve.

   Number of Machines => the number of machines listed in the Distributed Machine Configuration.
5. Optionally, you can select one of the following from the Default Process Priority pull-down

list:



- Critical (highest) Priority (Not recommended)
- Above Normal Priority (not recommended)
- Normal Priority
- Below Normal Priority
- Idle (lowest) Priority

You can set these values using VB Scripts.

To edit configurations, see Editing Distributed Machine Configurations.

Related Topics

High Performance Computing (HPC) Integration

You can use HPC for the following Example projects:

- Helical Antenna - You can enable Domain Decomposition method and setup HPC to solve this large problem.
- Bandpass Filter - You can setup HPC to distribute the frequencies in the sweep.
- Connector - Since the project uses a frequency sweep, you can set up HPC.

Note Click **File > Open Examples** to access the example projects. For more information about the application of HPC in the Bandpass Filter, see the section **Add HPC Analysis and Options** in *Getting Started with HFSS: Bandpass Filter*.

Specifying the Remote Spawn Command as RSH or SSH (Linux)

An important step in using a high performance cluster is setting up authentication across machines in such a way that the machines can be accessed without a password. By default HFSS and HFSS-IE use SSH authentication on Linux to spawn commands on the remote machines but also supports RSH. The selection of which to use is made on the **Options** tab of the Tools>Options>HPC and Analysis dialog.

SSH

You will need to set up passwordless access to use HFSS-IE on a Linux cluster with SSH or RSH. In general, for SSH, this is accomplished by

1. Verify that you have working SSH servers and clients on your machines.
2. Verify that the server will accept passwordless logins. You may need to edit the /etc/ssh/ssh_d file to allow RSAAuthentication and PubkeyAuthentication.
3. Generating keys on the client system using the ssh-keygen program. Do not use a passphrase so that you can access the machine without a password.
4. Copy the public key generated in step 1 from the ~/.ssh directory to the server. The easiest way to transfer the keys is to use the ssh-copy-id program. Alternately, you can use any file transfer utility. If the server already has a list of existing keys for other clients add the new public key to the list.
5. Test the connection. Login to the client machine using the username that you used to create the identity keys. Open a new shell terminal and attempt to open an SSH login session. For example type: ssh 192.168.0.4 (where the IP address is the address of the machine you are attempting to connect to). The server should allow you to login without requesting a password.

See the documentation for your machines and network for detailed instructions.

RSH

If you choose to use RSH you will need to make sure RSH is installed on all the machines and set the machines up so that you are not prompted for a password. There are different ways to set up password-less RSH so be sure to see the documentation for your machines and network for detailed instructions.

Machine access using RSH without a password is often set up by editing the /etc/hosts.equiv file and adding entries for the hosts you would like to use without a password. This file lists hosts and users that are granted "trusted" access to the system.

If you look at the /etc/hosts.equiv file you should have something similar to the following:

Contents of the /etc/hosts.equiv file:

```
job1.n1.com
job2.n1.com
job3.n1.com
```

The machines job1, job2 and job3 can connect without a password. You may also need to verify that the files /etc/hosts.allow and /etc/hosts.deny are empty. See your local documentation for detailed instructions and troubleshooting suggestions.

Related Topics

Distributed Memory Solutions with HFSS

Distributed Memory Solutions with HFSS-IE

Setting HFSS Options

To set HFSS options:

1. Click **Tools>Options> HFSS Options**.
   The **HFSS Options** window appears:
2. To change the default solution type when you initially insert a project, select one of the following from the Default solution type pull-down list:
   - Eigenmode
   - Driven Modal
   - Terminal
   - Transient
   - Transient Network
3. In the **Material Options** section:
   - Check or uncheck whether to **Include ferrite materials**
   - Set the **Solve Inside threshold** values in Siemens/m.
4. In the **Assignment Options section**, select or clear the following two check boxes and settings.
   - Use Wizards for data input when creating new boundaries

     When this is checked, the creation of boundaries and excitations use Wizard to guide you through the process. When this is not checked, the creation of boundaries and excitations displays a Properties dialog with tabs for different kinds of information.
   - Duplicate boundaries/mesh operations with geometry

     When this is checked, you can duplicate a boundary or excitation when its geometry is pasted or duplicated. See Duplicating Boundaries and Excitations with Geometry.
   - Visualize Boundaries on geometry.

     When this is checked, boundaries on geometries are displayed. Unchecking this turns off boundary visualization, and speeds up the display for complex models.
   - Auto assign terminals on ports

     When this is checked, the commands to assign wave or lumped ports will automatically assign terminals. See Assigning Wave Ports for Terminal Solutions.
5. In the **Post Processing Options** section:
   - Set the default Matrix sort order. This affects the order of the Matrix Data, and is of interest depending on how port names are assigned for that design. The default is ascending alphanumeric. This can also be a User Specified order that defaults to creation order.
6. Select or clear the following check boxes:
   - Save before solving
   - Save Optimetrics field solutions

- Apply variation deletions immediately

Configuring Distributed Analysis

Setting Solver Settings Using VB Scripts

Setting Solver Settings Using VB Scripts

Being able to set the number of processors, desired RAM limit, and maximum RAM limit using VB scripts allows you to solve a project by running a VB script, without having to toggle the computer hardware usage settings.

The following macros allow you to set these options:

```
oAnsoftApp.SetDesiredRamMBLimit <integer num in MB>
oAnsoftApp.SetMaximumRamMBLimit <integer num in MB>
oAnsoftApp.SetNumberOfProcessors <num>

limit = oAnsoftApp.GetDesiredRamMBLimit()
limit = oAnsoftApp.GetMaximumRamMBLimit()
num = oAnsoftApp.GetNumberOfProcessors()
```

Setting HFSS-IE Options

To set the HFSS-IE options:

1. Click **Tools>Options>HFSS-IE Options**.
   The **HFSS-IE Options** window appears:

These options are set on the **General Options** tab of the **HFSS-IE Options** dialog box. Clear or check the following options:

- **Duplicate boundaries with geometry**

  When this is checked, you can duplicate a boundary or excitation when its geometry is pasted or duplicated. See Duplicating Boundaries and Excitations with Geometry.
- **Save before solving**
- **Apply variation deletions immediately**.

  Checking this saves disk space.
- **Save Optimetrics Field solutions.**

  Checking this uses more disk space.
- **Visualize boundaries on geometry.**

  This controls the display of boundaries.
- **Auto-assign terminals on ports**.
- If this option is switched on, then the **Assign Lumped port** command will try to automatically assign the terminals on ports. If this option is switched off, then you assign a lumped port. You have a choice to either:

a. Manually assign a terminal. Select any edge/face of conductor that is touching the port.

b. Use **Excitations>Auto Assign Terminals**. This will bring up a dialog box to get the input from the user to select the objects used for "reference"

There are certain validation checks, like

Lumped port can have only one terminal.

No two terminals can touch each other

Unassigned terminals (not assigned to any port)

- Use Wizards for data input when creating new boundaries

  When this is checked, the creation of boundaries and excitations use Wizard to guide you through the process. When this is not checked, the creation of boundaries and excitations displays a **Properties** dialog with tabs for different kinds of information.

- **Default matrix sort order**. This affects the order of the Matrix Data, and is of interest depending on how port names are assigned for that design. The default is ascending alphanumeric. This can also be a User Specified order that defaults to creation order.

Related Topics

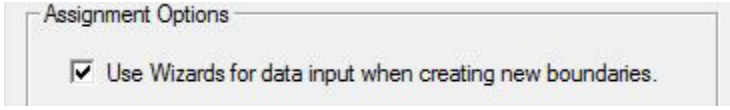
Configuring Distributed Analysis

Distributed Memory Solutions with HFSS-IE

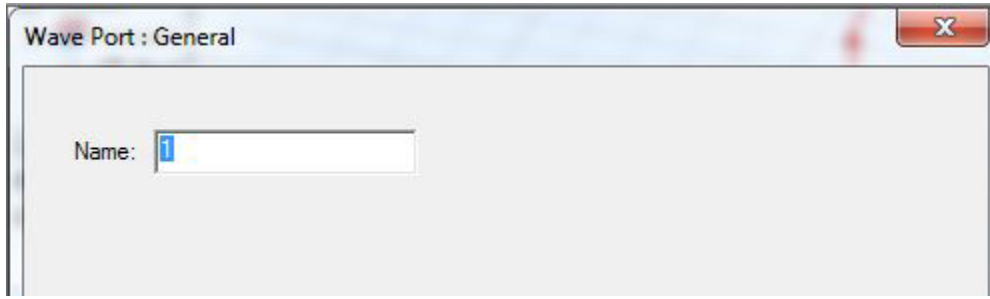
Wave Port Wizard or Multi-tab Dialog Menu

When assigning a wave port HFSS displays either a wave port wizard or a multi-tab dialog menu depending upon the settings on the **General** tab of the **HFSS Options** dialog box. Consider the following cases:

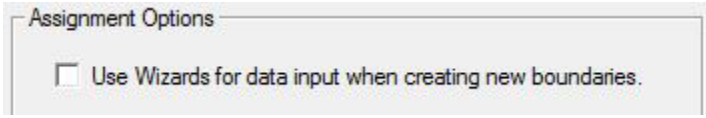
Case 1: The option **Use Wizards for data input when creating new boundaries** is checked.



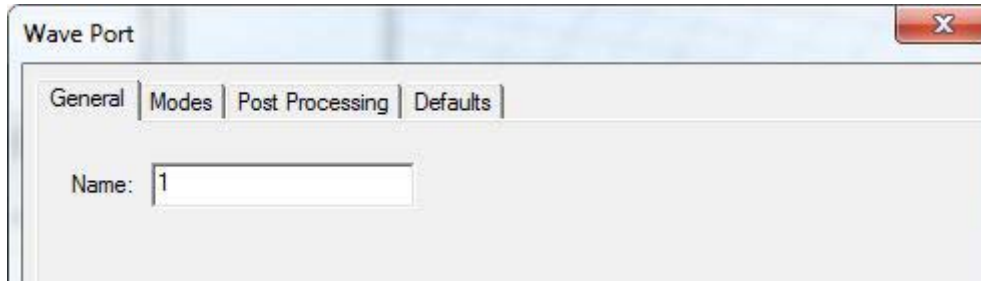
When assigning an excitation, the **Wave Port: General** wizard will appear as shown below.



Case 2: The option **Use Wizards for data input when creating new boundaries** is unchecked.



When assigning the excitation the multi-tab **Wave Port** dialog box appears with the **General** tab selected as shown in the figure below.



Type the port's name in the **Name** text box or accept the default name.

Note To change the default base name, see Setting Default Boundary/Excitation Base Names.

Related Topics

HFSS Options

HFSS-IE Options

Assigning Wave Ports for Modal Solutions

Assigning Wave Ports for Terminal Solutions

Technical Notes: Wave Ports

Technical Notes: Port Solution Theory

Setting Fields Reporter Options

To set the **Fields Reporter** options:

1. Click **Tools>Options>Fields Reporter Options**.

   The **Fields Reporter Options** dialog opens with the Phase Animation tab selected.
2. Specify whether to **Group Field Overlays by Type** (default, yes).
3. Set the default Phase Animation settings for **Scalar Plots** and **Vector Plots.**

   Each of these accepts values for From and To in degrees, and the number of steps.
4. Click the **Mesh Plot** tab to set when dragging the clip plane, to update the plot dynamically Never, Always or When the number of mesh elements is less that a value (default 5000).
5. Click the **Streamline** tab to set the two Streamline drawing stopping criteria and the Streamline marker spacing.

The Streamline drawing stopping criteria are:

Fields magnitude on a streamline data point is <*percentage*> of maximum field value of entire model.

Streamline length is higher than <*value*> times of diagonal length of model bounding box.

The streamline marker spacing the Number of markers per bounding box diagonal.

6. Click OK to accept the settings and close the dialog.

Related Topics

Creating Phase Animations

Plotting the Mesh

Setting Report2D Options

To set Report2D options:

1. Click **Tools>Options>Report2D Options**.
   The **Report2D Options** window appears, displaying twelve available tabs:
   - Curve Tab
   - Axis Tab
   - Grid Tab
   - Header Tab
   - Note Tab
   - Legend Tab
   - Marker Tab
   - Marker Table Tab
   - X/Y Markers Tab
   - Digital Tab
   - General Tab
   - Table Tab

   For properties controlled by checkboxes, you can set values for all curves by clicking the column header cell that contains the property title. Right-clicking on a text field cell displays a context menu that lets you cut, copy and paste values. Right-clicking on a menu cell displays a context menu that lets you copy and paste entire rows.

   You can use a Restore Defaults button.
2. Click each tab, and make the desired selections.
3. Click **OK**.

Report 2D Options: Curve Tab

These options are set on the **Curve** tab of the **Report2D Options** dialog box.

1. Line style -- select the options from the drop down menu. The options are Solid, Dot, Dash,

and Dot dash.

2. Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
3. Width -- set the line width by editing the real value in the text field.
4. Arrows -- use the check box to use arrows on the curve ends.
5. Symbol -- use the check box to have symbols mark the locations of data points on the curve.
6. Sym Freq -- set the symbol frequency by editing the integer value in the text field.
7. Sym Style -- select the symbol to display for the designated data points. The sym style can be box, circle, vertical ellipse, horizontal ellipse, vertical up triangle, vertical down triangle, horizontal left triangle, horizontal right triangle.
8. Fill Sym -- use the check box to set the symbol display as a solid or as hollow.
9. Sym Color -- set the color for the symbol by double clicking to display the Set color dialog. Select a default or custom color and click OK.

Report2D Options: Axis Tab

These options are set on the **Axis** tab of the **Report2D Options** dialog box.

1. Axis Name -- this describes the axis to which the following options refer.
2. Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
3. Auto Scale -- use the check box to toggle whether to auto scale the axis.
4. Min Scale -- if Auto Scale it not selected, edit the real value to set the minimum value of the axis.
5. Max Scale -- if Auto Scale is not selected, edit the real value to set the maximum value of the axis.
6. Auto Units -- use the check box compute the correct units for the axis.
7. Units -- click on the cell to select from a menu of available units if you have not checked Auto Units.
8. Font color -- set the font color of the axis by double clicking to display the Set color dialog. Select a default or custom color and click OK.
9. Edit Font -- click the cell to display the Edit Text Font dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.

Report2D Options: Grid Tab

These options are set on the **Grid** tab of the **Report2D Options** dialog box.

1. Grid Name -- lists the name or letter of the grid. Not editable.
2. Line Style -- select the options from the drop down menu. The options are Solid, Dot, Dash, and Dot dash.
3. Line Color -- set the color by double clicking to display the Set color dialog. Select a default or

custom color and click OK.

Report2D Options: Header Tab

These options are set on the **Header** tab of the **Report2D Options** dialog box. For the Title and subtitle, you can independently specify the following:

1. Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
2. Font -- click the cell to display the Edit Text Font dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.

Report2D Options: Note Tab

These options are set on the **Note** tab of the **Report2D Options** dialog box.

1. Note Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
2. Note Font -- click the cell to display the Edit Text Font dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.
3. Background Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
4. Background Visibility -- use the checkbox to toggle the background for the note on or off.
5. Border Line Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
6. Border Visibility -- use the checkbox to toggle the visibility of the note border.
7. Border Line Width -- set the line width by editing the real value in the text field.

Report2D Options: Legend Tab

These options are set on the **Legend** tab of the **Report2D Options** dialog box.

1. Show Trace Name -- use the checkbox to toggle the visibility of the trace name.
2. Show Solution Name -- use the checkbox to toggle the visibility of the solution name.
3. Show Variation Key -- use the checkbox to toggle the visibility of the variation key.
4. Text Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
5. Text Font -- click the cell to display the Edit Text Font dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.
6. Background Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
7. Border Line Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.

8. Border Line Width -- set the line width by editing the real value in the text field.
9. Grid Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.

Report2D Options: Marker tab

These options are set on the **Marker** tab of the **Report2D Options** dialog box.

1. Marker Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
2. Marker Font -- click the cell to display the Edit Text Font dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.
3. X Marker -- use the following options to set the X Marker properties.
    a. Show Intersection -- checkbox to show the intersection.
    b. XMarker Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
    c. XMarker Font -- click the cell to display the Edit Text Font dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.
    d. Box Background Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
    e. Line Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
    f. Line Style -- select the options from the drop down menu. The options are Solid, Dot, Dash, and Dot dash.
    g. Line Width -- set the line width by editing the real value in the text field.

Related Topics

Modifying Markers on Point Plots

Report2D Options: Marker Table Tab

These options are set on the **Marker Table** tab of the **Report2D Options** dialog box.

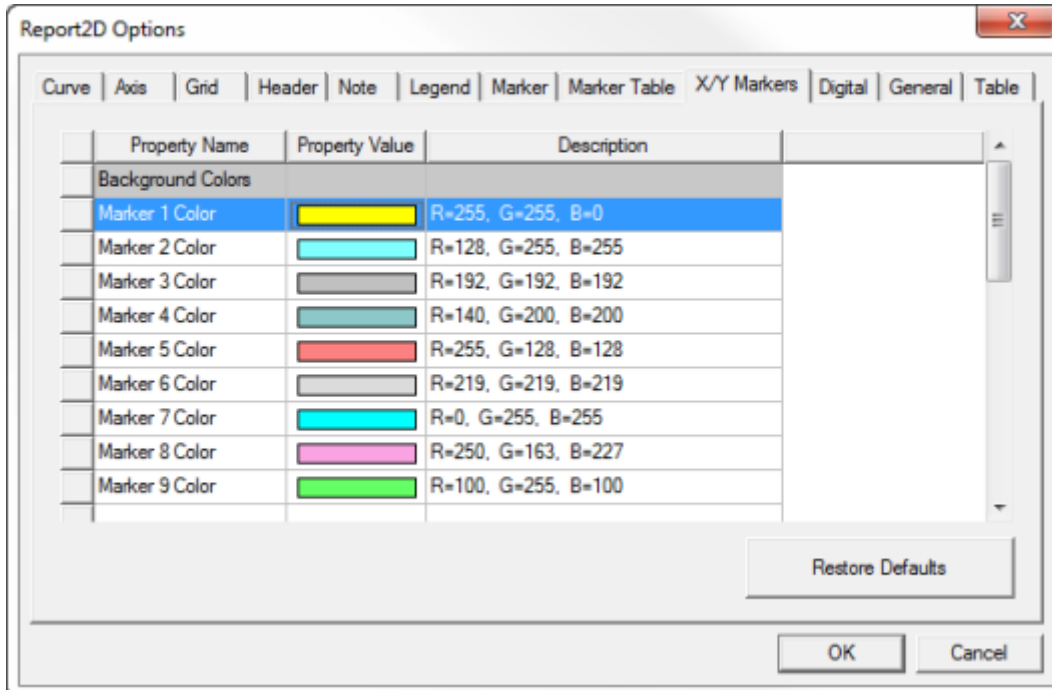
1. Precision -- set the precision for marker placement by editing the real value field.
2. Text Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
3. Text Font -- click the cell to display the Edit Text Font dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.
4. Background Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
5. Border Line Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.

6. Border Line Width -- set the line width by editing the real value in the text field.
7. Grid Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
8. Grid Line Width -- set the line width by editing the real value in the text field.

Report 2D Options: X/Y Markers Tab

Use this tab on the **Report2D Options** dialog box to set the properties for the markers.

Background colors for Markers 1 through 10. You can set these by select the current color to open a a color selection dialog, or by specifying RGB number values.



Properties, including

- On-screen intersection
- Marker Font
- Text color
- Line color
- Line style
- Line width
- Whether to Show Name
- Whether to Snap to Vertex

Inter marker deltas, including

- Whether to show delta
- Delta font
- Delta text color
- Line color
- Line style
- Line width

You can also **Restore Defaults**.

Report2D Options: Digital Tab

These options are set on the **Digital** tab of the **Report2D Options** dialog box.

Digital Literal Foreground color.

Whether to Expand Arrays/Records

Digital Stack Hight in Pixels for the following:

- Analog
- Digital
- Enum
- Event
- Literal.

You can also **Restore Defaults**.

Report2D Options: General Tab

These options are set on the **General** tab of the **Report2D Options** dialog box.

1. Background Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
2. Contrast Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
3. Highlight Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
4. Accumulate Depth -- set to 4 by default.
5. Curve Tooltip Option -- use the checkboxes to toggle the following properties:
   a. Show Trace Name
   b. Show Variation Key
   c. Show Solution Name
6. Clipboard Option - use the drop down menus to specify the following properties:
   a. Capture Aspect Size Ratio -- this can be As Shown or Full Screen.
   b. Capture Background Color -- this can be As Shown or White.

Report2D Options: Table Tab

These options are set on the **Table** tab of the **Report2D Options** dialog box.

1. Text Font -- click the cell to display the Edit Text Font dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.
2. Format -- use the following properties to set the format:
   a. Field Width -- set the table field width by editing the real value in the text field.
   b. Precision -- set the table precision by editing the real value in the text field.
   c. Use Scientific Notation -- use the checkbox to toggle scientific notation on or off.
3. Copy to Clipboard -- use the following checkboxes to toggle the following properties for table copy operations.:
   a. With Header
   b. With Tab Separator -- this improves copy and paste to a spreadsheet.

Setting Modeler Options

To set modeler options:

1. Click **Tools>Options>Modeler Options**.
   The **Modeler Options** window appears, displaying three available tabs:
   - Operation
   - Display
   - Drawing
2. Click each tab, and make the desired selections.
3. Click **OK**.

Modeler Options: Operation Tab

These options are set on the **Operation** tab of the **Modeler Options** dialog box.

1. By default, the modeler deletes tool objects when performing tasks such as Uniting, Subtracting, or Intersecting objects. You may specify that the modeler makes a copy (clone) of the tool object before the operation, keeping the object for subsequent operations. To specify when to clone tool objects, select or clear the following check boxes in the **Clone** section:
   - Clone tool objects before uniting
   - Clone tool objects before subtracting
   - Clone tool options before intersecting
   - Clone tool objects before imprinting
   - Clone tool objects before projecting
2. In the **Coordinate System** section, select or clear the **Automatically switch to face coordinate system** check box.

   By default, the modeler operates within the user selected coordinate system. If this option is

enabled, you can select a face and when a new object creation is started, the modeler first creates a face coordinate system consistent with the selected face and the new object is created within the face coordinate system. With this selection, unchecked, you must manually create a Face Coordinate System before creating an object related to it.

3. In the **Polyline** section, select or clear the **Automatically cover closed polylines** check box.

   By default, surface objects created with the **Polyline** command will be created with a cover so that they become sheet objects. You can choose to leave the polyline as an uncovered object to perform further operations prior to creating a sheet object

   - If checked, closed polylines become sheet objects, and are listed as such in the History tree.
   - If not checked, closed polylines are listed under lines in the History tree.

4. For the **Model Edit** section, select or clear Delete invalid objects created during split operation.
   - If checked, the modeler deletes invalid objects created during split operation.
   - If not checked, invalid objects can be created. Validation issues warnings.

   In the **Model Edit** section, select or clear Automatically imprint wrapped sheets.

5. For the **Select last command on object select** option:
   - If checked, the history tree is expanded after operations on object properties, even if the tree is collapsed for the item.
   - If not checked, when you select an object in 3D view, only the object selected, and current tree collapse/expand state is preserved

6. For the **Expand history tree on object select** option:
   - If checked, selecting an object automatically opens the history tree.
   - If not checked, the history tree does not open on object selection. This can be useful for speeding the display of multiple object selections, or complex objects.

7. **UDM/UDP computation for Optimetrics Analysis**

   Options are:
   - Engine computes the geometry
   - Desktop computes the geometry

   Geometry computation with CAD Integration (dynamic links) and geometry sharing in ANSYS Workbench is always by Desktop.

Modeler Options: Display Tab

These options are set on the **Display** tab of the **Modeler Options** dialog box.

1. To specify a default color for a 3D Modeler drawing object or action (such as on select):
   - Select the object or action from the **Default color** pull-down list and click the color button.
     The **Color** window appears.
   - Select a color for the selected object or action, and click **OK**.

2. To specify the default for the **View>Render** setting for new projects, select **WireFrame** or **SmoothShade** from the **Default view render** pull-down list.

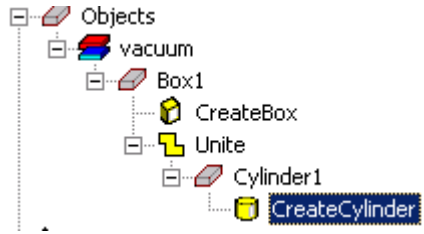
   When dealing with complicated geometries, choose **WireFrame** rendering. This is faster than shaded rendering.
3. To set the **Default transparency**, move the slider, or enter a numerical value.
4. Select or clear the **Show orientation of selected objects** check box.
5. Select or clear the **Highlight selection dynamically** check box.
6. Select or clear **Highlight UV Isolines**.

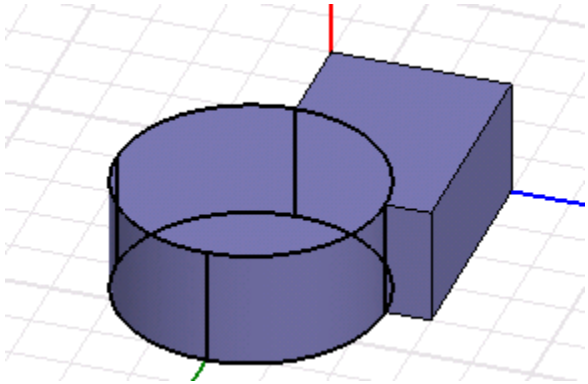
   For models with curved faces, you may prefer to clear this selection to simplify the wire-frame display, so the rendering will be faster.
7. Under **Default tree layout**, select or clear the **Group objects by material** check box.
8. Under **History operations visualization**, select or clear the **Visualize history of objects** check box.

   The option lets you view an outline of each part that comprises an object when the given part is selected in the model history tree. This can help you visualize an object that has been merged with another object. A change to the option takes effect only when you restart HFSS.

   Clearing this selection removes visualization of objects that are part of the model history. For large models, this is faster and uses less memory.

   The following figure shows an example history tree with an object selected and the outline view of that object in the Main window.





Modeler Options: Drawing Tab

These options are set on the **Drawing** tab of the **Modeler Options** dialog box.

1. To specify snap settings, select or clear the following check boxes in the **Snap Mode** section:
   - Grid
   - Vertex
   - Edge Center
   - Face Center
   - Quadrant
   - Arc Center
2. Enter how near the mouse needs to be to click a grid item in the **Mouse Sensitivity** box, in pixels.
3. Select or clear the **Show measures dialog** check box.

   The specifies whether a Properties dialog appears on the creation of a new primitive.
4. The **Operation Data Mode** controls whether you draw new objects directly via the mouse, or whether a **Properties** dialog opens for you to enter dimensions for the object. The **Dialog** mode drawing feature works with the equation based line, and all two and three dimensional objects.
   - Point mode - mouse drawing.

- Dialog - enter dimensions in the properties dialog.

You can also use **F3** for Point mode and **F4** for dialog mode.

5. To have a **Properties** dialog display whenever you create a new object in the modeling window, check the box for **Edit properties of new primitives**.

Report Setup Options

To set Report Setup options:

1. Click **Tools>Options>Report Setup** or in a Reporter dialog click the Options button.

   The **Report Setup** dialog appears.
2. Use the text field to specify the maximum number of significant digits to use when displaying numeric values.
3. Specify the drag and drop behavior by clicking the radio button.
   - Drag item data
   - Drag item definition
4. The Quantity selection value specifies the matrix size for using a tree display for matrix quantities. This is helpful when dealing with larger matrices. The default is 50. When the number of matrix elements is larger than the number, the Quantities field uses a tree structure to divide matrix quantities into groups by their first element name. The initial display shows groups, without initially listing group members.

   A folder Node is not selectable. A click on a folder node toggles (Expand or collapse) the node.

   When any of a folder's child nodes is selected it becomes bolded.

   Mouse Click on Quantity Node (Tree leaf node). Shift and Ctrl key only apply to multiple selection dialogs:
   - Without Shift and Ctrl key - Select the quantity and unselect all previous selected quantities.
   - Only with Ctrl key down - Toggle the selection of the quantity. No affect on other selected quantities.
   - Only with Shift key down - Do range selection, deselect any selected quantity that is outside of the range.
   - Both Shift and Ctrl key down - Do range selection, but don't deselect any selected quantity.

- Ctrl+a - Selected all quantities in a multiple selections dialog.

Range selection: Select quantity nodes between the last mouse clicked quantity node and the newly click on quantity node. Folder nodes in between won't be selected but their children will be selected. So those folder nodes will be in a bolded state.

Exporting Options Files

The options files at all levels that may affect a product running as a specific user on a specific host may easily be exported. The Desktop UI may be used to copy these files to a user specified directory. A script command may also be used to copy these files to a specified directory.

Using the Desktop UI

You can export options files by selecting the **Tools>Options>Export Options Files ...** This brings up a browser dialog that you use to select the destination directory for the options files. Pressing the Cancel button will cancel the export command. Pressing the Open button will copy all of the config files for the current user and current host to the specified directory. Config files for the install, install_machine, user, and user_machine levels will be copied, if they exist. One additional file, admin.XML, will also be copied to the destination directory. This file does not contain user configurable options, and it is not discussed above.

Using a Script

A Desktop command has been added that exports the options config files. Here is a summary of the Desktop ExportOptionsFiles command:

ExportOptionsFiles

Use: Copies the options config files to the DestinationDirectory.

Command: **Tools>Options>Export Options Files ...**

Syntax: ExportOptionsFiles <DestinationDirectory>

Return Value: None

Parameters: <DestinationDirectory>

Type: <string>

Example:

oDesktop.ExportOptionsFiles "D:/test/export/"

Example Script Using the Desktop ExportOptionsFiles Command

A simple script demonstrating the use of this command is shown below. This script will copy the options config files to the directory D:/test/export.

```
Dim oAnsoftApp
Dim oDesktop
Set oAnsoftApp = CreateObject("AnsoftHfss.HfssScriptInterface")
Set oDesktop = oAnsoftApp.GetAppDesktop()
oDesktop.ExportOptionsFiles "D:/test/export/"
```

Related Topics

Setting Options in HFSS

Setting Options via Configuration Files

Example Uses for Export Options Features

Setting Options via Configuration Files

In addition to setting options from the Desktop UI, you can also set options in several configuration files. Options that you set from the Desktop UI will override the option settings from the configuration files. The option settings from the configuration files are used if they are not overridden using the Desktop UI. Option settings in the configuration files may apply to all users or only to a specific user, and they may apply to all hosts or only to specific hosts. There are four levels, listed below from most specific (highest precedence) to most general (lowest precedence):

- host dependent user options (apply to the specified user on the specified host only)
- host independent user options (apply to the specified user on all hosts)
- host dependent default options (apply to all users on the specified host)
- installation default (default for all users on all hosts)

A setting at any level will override settings at lower levels in the list above. If there is no setting in any file, then the application default value will be used.

Behavior Examples

Rules for Modifying Option Settings

Configuration File Locations

Products with Multiple Desktop Applications

Table of Directories and Files

Behavior Examples

For example, consider running an application as user jsmith on host host123. If there is no host dependent user setting for the "Expand Project Tree on Insert" option in the host dependent user options config file for user jsmith on host host123, but there is a setting for the "Expand Project Tree on Insert" option in the host independent user options config file for user jsmith, then the latter setting will be used if it is not overridden using the Desktop UI. Any settings in the host dependent default options config file or the installation default config file will be ignored.

As another example, consider running an application as user jdoe on host host123. If there is no setting for the "Expand Project Tree on Insert" option in the host dependent user user options config file for jdoe on host123 or in the host independent user options config file for user jdoe or in the host dependent default options config file for host host123, then the value from the installation default config file will be used, if present.

Rules for Modifying Option Settings

Option settings displayed in the Desktop UI follow the above rules. That is, if there is a setting in any of the option config files, then the setting from the highest priority config file is displayed in the Desktop UI. If there is no setting in any of the option config files, then the global default value is used. You can modify settings using the Options dialog boxes in the Desktop UI. If the dialog box is closed with the "Cancel" button, then changes made on any of the tabs are discarded. If the dialog box is closed with the "OK" button, then any settings that have been changed by the user are written to the host dependent user options config file. The changed values written to this file are

then used the next time that the application is run by the same user on the same host. The Desktop UI option settings are not written to any of the other option config files.

Configuration File Locations

Two of the configuration files (the host dependent default options config file and the installation default config file) reside in the installation config directory, which is the "config" subdirectory of the installation directory. The other two configuration files (the host dependent user options config file and the host independent user options config file) reside in a user specific config directory for the application. The user specific config directory for an ANSYS Electromagnetics application is in a subdirectory of the user's My Documents directory on Windows, or a subdirectory of the user's HOME directory on Linux. The subdirectory pathname is AnsysEM/<ApplicationPathAndVersion>/config, where ApplicationPathAndVersion is the usually the product name concatenated with the product version, such as HFSS14.0.

Products with Multiple Desktop Applications

For products that have multiple Desktop applications, each Desktop application will have a separate user specific config directory, with a different value for the ApplicationPathAndVersion directory name. For example, the Maxwell 15.0 product contains two Desktop applications: Maxwell and Maxwell Circuit Editor. For the Maxwell application, the user specific config subdirectory pathame is AnsysEM/Maxwell15.0/config. For the Maxwell Circuit Editor application, the user specific config subdirectory pathame is AnsysEM/Maxwell Circuit Editor 15.0/config. For products that contain multiple Desktop applications, the installation default option settings and the host dependent default options settings each reside in a single file for all applications, while the host dependent user option settings and host independent user option settings for each application are in separate files. Although the installation default option settings and the host dependent default options settings for all applications are each written to a single file, the settings for different applications are in separate portions of the file.

Table of Directories and Files

The table below shows the directories and files, where the "Level Name" is the name used to describe an options config file when using the UpdateRegistry? tool, described below.

Config File	Level Name	File Name	Linux Directory Pathname	Windows Directory Pathname
host dependent user options	user_machine	hostname_user.XML	$HOME/ AnsysEM/ ApplicationPathAndVersion/config	MyDocuments/ AnsysEM/ ApplicationPathAndVersion/config
host independent user options	user	user.XML		
host dependent default options	install_machine	hostname.XML	InstallationDirectory/config	InstallationDirectory/config
installation default	install	default.XML		

NOTES:

$HOME is the user's home directory on Linux

ApplicationPathAndVersion is the Application or Product name concatenated with the Product Version

MyDocuments is the user's "My Documents" directory on Windows

InstallationDirectory is the directory where the product is installed

The table below shows specific file names and directory names for a typical Maxwell 15.0 installation on the Linux and 32 bit Microsoft Windows platforms. These are the files that apply to user jsmith and hostname host123.

Application	Config File	Level Name	File Name	Linux Directory Pathname	32 Bit Windows Directory Pathname
Maxwell	host dependent user options	user_machine	host123_user.XML	/home/jsmith/AnsysEM/Maxwell15.0/config	MyDocuments/AnsysEM/Maxwell15.0/config
	host independent user options	user	user.XML		
	host dependent default options	install_machine	host123.XML	/opt/AnsysEM/maxwell15.0/Linux/config	C:/Program Files/AnsysEM/Maxwell15.0/Win32/config
	installation defaul	install	default.XML		
Maxwell Circuit Editor	host dependent user options	user_machine	host123_user.XML	/home/jsmith/AnsysEM/Maxwell Circuit Editor15.0/config	MyDocuments/AnsysEM/Maxwell Circuit Editor15.0/config
	host independent user options	user	user.XML		
	host dependent default options	install_machine	host123.XML	/opt/AnsysEM/maxwell15.0/Linux/config	C:/Program Files/AnsysEM/Maxwell15.0/Win32/config
	installation default	install	default.XML		

NOTES:

/home/jsmith is the home directory of user jsmith on Linux

MyDocuments is the "My Documents" directory for user jsmith on Windows

The Maxwell 15.0 product is installed in directory /opt/AnsysEM/maxwell15.0 on Linux

The Maxwell 15.0 product is installed in directory C:/Program Files/AnsysEM/Maxwell15.0/Win32 on 32 Bit Microsoft Window

Related Topics

Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

A command line tool, UpdateRegistry, is available to modify option settings in the options config files. You can use this command to add, change or remove settings from any of the option config files. This tool is included in the installation directory of each product.

This feature is intended to make it easier for different users to use ANSYS Electromagnetics tools installed on shared directories or network drives.

The UpdateRegistry command has multiple command line formats as shown below. The -Set format is used to set or change an option value. The -Delete format is used to delete an option setting. The following command line options are mutually exclusive: -Set, -Get, -Delete, -GetKeys, and -FromFile.

UpdateRegistry -Get Command

This command is used to view an option value in an option config file. If the setting exists in the specified config file or files, then the value, the value type and the config file where the value was found will be reported. If no value is found, then that will also be reported.

Usage:

```
UpdateRegistry -Get -ProductName <name> -RegistryKey <keyPath>
[ -RegistryLevel <level> ]
```

<name>

Required. The application or product name and version, as described above. Examples: HFSS15.0, Maxwell15.0, or "Maxwell Circuit Editor15.0". If the name contains spaces, it must be quoted.

<keyPath>

Required. The pathname of the option setting.

Example:

```
Desktop/Settings/ProjectOptions/AnimationMemory.
```

<level>

Optional. A string denoting which config file to search. One of: install, install_machine, user and user_machine. If the level is not specified, then all config files are searched in order of precedence.

UpdateRegistry -GetKeys Command

This command is used to view the allowed key names for all of the option settings, or to view a subset of the key names that match a string. For each key displayed, the current value, if any, is also

reported. If a key has a value in multiple config files, then only the highest precedence value is reported.

Usage:

```
UpdateRegistry -GetKeys [ <pattern>] -ProductName <name> [ -
Case ]
```

<pattern>

Optional. If no pattern is specified, then all allowed key names are reported. If a pattern is specified, then only keys that match the pattern are shown. Examples: Settings/Project. If the name contains spaces, then it must be quoted. By default, the pattern match is case insensitive. If the -Case command line option is specified, then the pattern match is case sensitive.

<name>

Required. The application or product name and version, as described above. Examples: HFSS14.0, Maxwell15.0, or "Maxwell Circuit Editor15.0". If the name contains spaces, it must be quoted.

UpdateRegistry -Set Command

This command is used to add or modify an option setting in an option config file. If the option config file does not exist, it will be created. If the setting does not exist in the specified config file, it will be added. If the setting already exists in the specified config file, then the value will be changed to the specified value.

Usage:

```
UpdateRegistry -Set -ProductName <name>
-RegistryKey <keyPath>
-RegistryValue <value> [ -RegistryLevel <level>
<name>
```

Required. The application or product name and version, as described above. Examples: HFSS14.0, Maxwell15.0, or "Maxwell Circuit Editor15.0". If the name contains spaces, it must be quoted.

<keyPath>

Required. The pathname of the option setting. Example:

```
Desktop/Settings/ProjectOptions/AnimationMemory.
```

<value>

Required. The new value of the option, typically a string or a number. If the value contains spaces, it must be quoted.

<level>

Optional. A string denoting which config file to modify. One of: install, install_machine, user and user_machine. If the level is not specified, then the user_machine (host dependent user options) file is modified.

UpdateRegistry -Delete Command

This command is used to remove an option setting from an option config file. If the setting does not exist in the specified config file, the file will not be changed. If the setting exists in the specified

config file, then it will be removed. A setting may need to be removed from an option config file, to allow the setting from a lower priority file to be used by the application.

Usage:

```
UpdateRegistry -Delete -ProductName <name>
-RegistryKey <keyPath>
[ -RegistryLevel <level> ]
```

<name>

Required. The application or product name and version, as described above. Examples: HFSS14.0, Maxwell15.0, or "Maxwell Circuit Editor15.0". If the name contains spaces, it must be quoted.

<keyPath>

Required. The pathname of the option setting. Example:

`Desktop/Settings/ProjectOptions/AnimationMemory.`

<level>

Optional. A string denoting which option config file to modify. One of: install, install_machine, user and user_machine. If the level is not specified, then the user_machine (host dependent user options) file is modified.

UpdateRegistry -FromFile Command

You can use this form of the UpdateRegistry command to set multiple key-value pairs from a file with a single UpdateRegistry command. You specify the -FromFile command line option. This option must be followed by a filename. The file may contain multiple entries, where each entry contains a registry key and a registry value. The key-value pairs are added to the registry level specified by the -RegistryLevel command line option; if no -RegistryLevel is specified, then the default registry level (user_machine) is used.

UpdateRegistry File Format

The file format is similar to the -batchoptions file format. An example UpdateRegistry file is shown below:

```
$begin 'AddEntries'
   'TempDirectory'='C:/temp/AnsysEM'
   'Hfss/HPCLicenseType'='Pool'
   'Hfss/UseLegacyMultiprocessingLicense'=1
$end 'AddEntries'
```

Additional notes on the file format:

- The file may contain an arbitrary number of entries, one per line.
- Leading whitespace on each line is ignored. Spaces or tabs may be used to make the file more readable.

Registry key pathname:

- The registry key pathname appears before the equal sign "=" on each line.
- Each registry key pathname must be enclosed in single quotes.

Registry value:

- The registry value appears after the equal sign on each line.
- Integral registry values must not be enclosed in quotes.
- All other registry values are treated as strings, and must be enclosed in single quotes.
- The forward slash "/" may be used as a directory separator on Windows and Linux. The back slash "\" may be used as a directory separator on Windows only.
- The back slash "\" is used as an escape character in the value string. That it, this character removes the special meaning of the following character.
- The single quote character normally ends the value string. The back slash may be used to remove this special meaning, and include a single quote in the string.
- To use a back slash as a directory separator on Windows, it must be escaped. That is, a double back slash "\\" is used to denote a single directory separator.

Alternative UpdateRegistry File Format

- Analysis Configuration File format, which is exported from the HPC and Analysis Options dialog.

Related Topics

Setting Options via Configuration Files

Example Uses for Export Options Features

User Options and the Update Registry Tool

Batchoptions Command Line Examples

Example Uses for Export Options Features

The **Tools>Options>Export Options** feature is intended to make it easier for different users to use ANSYS Electromagnetics tools installed on shared directories or network drives. This section outlines some use cases enabled by this feature.

Options That Apply to All Users

Example Searching for a Registry Key Pathname

Example for Setting an Installation Default Value

Example for Setting a Host Dependent Default Value

Example for Reverting from a User Defined Option Value to the Administrator Default

Related Topics

Exporting Options Files

Setting Options via Configuration Files

Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

User Options and the Update Registry Tool

Batchoptions Command Line Examples

Options That Apply to All Users

In many cases, an ANSYS Electromagnetics tool installation is administered and maintained by a single user or group and used by a number of other users or groups. The permissions of the ANSYS Electromagnetics tool installation may be set so that the administrator may add, delete or modify files, but other users may only read or execute these files. The administrator may set the recommended option settings in the installation default config file and/or the host dependent default options config file. These config files reside within the installation directory hierarchy, and should generally have the same permissions as other ANSYS Electromagnetics tool installation files. This allows that administrator to control these settings, but does not allow other users to add, remove, or change these settings.

Each user can override any of these settings, if needed. This may be done using the Desktop UI, which affects the host dependent user options config file. It may also be done using the host independent user options config file. If user has overridden an option setting in either of the user files, the user may revert back to the option settings provided by the administrator by removing the setting of the same option in the host dependent user option config file and/or the host independent user option config file.

For global defaults, the administrator may set a value in the installation default config file. These settings will to apply to all users on all hosts.

In some cases, there are significant differences between the capabilities of different hosts. The host dependent default config file may be used to specify different default values on some hosts. Any setting in a host dependent default config file would affect all users running on the specified host. The installation default value is used if there is no value specified for the setting in the host dependent default config file for the current host. Note that the host dependent default config file is named hostname.XML, where hostname is the name of the host.

Related Topics

Example Uses for Export Options Features

Example Searching for a Registry Key Pathname

Both administrators and ordinary users may occasionally use the UpdateRegistry command line tool to add, change or delete settings. To use this tool, the registry key pathname must be known by the user. The -GetKeys option may be used to quickly search for a key pathname if some information is known about it. For example, if the administrator knows that there is a setting related to issuing warning messages when available disk space is low, but she does not know the exact key name, the following command may list some of the keys related to disk space:

```
UpdateRegistry -GetKeys disk -ProductName HFSS14.0
```

This will display a list of all keys that match the string "disk" case insensitively. Typical output may look like the following:

```
Registry keys matching pattern <disk> case insensitively:
```

```
Desktop/Settings/ProjectOptions/DiskLimitForAbort: value is <0>
at level <user_machine>
```

Example for Setting an Installation Default Value

The normal default for the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting is 0 MB. If the administrator is concerned that running out of disk space might be a common problem, the administrator could set the installation default for the Desktop/Settings/ProjectOptions/DiskLimit-ForAbort setting to 1000 MB, for example. This limit would then apply to all users running on all hosts. The administrator could use the following command to change this setting for HFSS 14.0:

```
UpdateRegistry -Set -ProductName HFSS14.0 -RegistryKey
Desktop/Settings/ProjectOptions/DiskLimitForAbort
-RegistryValue 1000
-RegistryLevel install
```

Related Topics

Example Uses for Export Options Features

Example for Setting a Host Dependent Default Value

For this example, we assume that all hosts have two cores, except for three hosts, bighost1, big-host2, and bighost3, that have eight cores each. Because most hosts have two cores, the administrator has set the Hfss/Preferences/NumberOfProcessorsDistributed option value to 2 in the installation default config file, as described above. The administrator may then set the Hfss/Preferences/NumberOfProcessorsDistributed option value to 8 in the host dependent default config files for the three hosts having 8 cores, bighost1, bighost2 and bighost3. The administrator may login to host bighost1, and run the following command to change this setting for the host dependent default options config file for host bighost1 on HFSS 14.0:

```
UpdateRegistry -Set -ProductName HFSS14.0
-RegistryKey Hfss/Preferences/NumberOfProcessorsDistributed
-RegistryValue 8
-RegistryLevel install_machine
```

To make this change for the other two hosts, the administrator would login to bighost2 and big-host3, in turn, and run the same command on each of those hosts.

Related Topics

Example Uses for Export Options Features

Example for Reverting from a User Defined Option Value to the Administrator Default

Consider the case in which HFSS 14.0 was installed and the administrator initially did not set a value for the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting in the default installation config file. User jsmith (who always uses host jshost) wanted to be warned before disk space dropped to zero, so he set the Desktop/Settings/ProjectOptions/DiskLimitForAbort to 100 MB using the UI. This setting is recorded in the host dependent user options config file for host jshost

and user jsmith. Now the administrator learns that many users are running into disk space issues, so that administrator sets the installation default value for the setting Desktop/Settings/ProjectOptions/DiskLimitForAbort to 1000 MB, as in the above example.

When user jsmith runs HFSS 14.0 on host jshost, the disk limit is 100 MB, not 1000 MB, because the host dependent user options config file overrides all of the other config files. User jsmith may revert to the administrator provided default by removing this setting from the host dependent user options config file for host jshost and user jsmith. The following command may be run by user jsmith on host jshost to remove this setting:

```
UpdateRegistry -Delete -ProductName HFSS14.0
-RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
-RegistryLevel user_machine
```

If user jsmith had added a value for this setting to the host independent user options config file, then user jsmith would also run the following command to remove this setting from the host independent user options config file:

```
UpdateRegistry -Delete -ProductName HFSS14.0
-RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
-RegistryLevel user
```

Related Topics

Setting Options via Configuration Files

Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

User Options and the Update Registry Tool

Batchoptions Command Line Examples

Example Uses for Export Options Features

User Options and the Update Registry Tool

When you change an options value using the Desktop UI, the new value is stored in the host dependent user options config file. You can also use the UpdateRegistry tool to add or modify settings in the host dependent user options config file. You cannot use the Desktop UI to remove settings from the host dependent user options config file, however. You must use the UpdateRegistry tool to remove settings from the host dependent user options config file.

If a user has not explicitly created a host dependent user options config file or a host independent user options config file, then when a user first runs an ANSYS Electromagnetics tool on a host, all settings will come from the host dependent default options config file or the installation default options config file. Any settings for another host in a host dependent user options config file will not be carried over to the new host. This may be inconvenient if the user has preferred option settings that differ from the settings that apply to all users, especially if the user runs the ANSYS Electromagnetics tool on a number of different hosts. In this case, the user may set these option values in the user's host independent user options config file. Then, these option values will be used on all new hosts, overriding any values set by the administrator to apply to all users. Any changes made in the UI will only affect the user's host dependent user options config file for the current host.

Example of Removing a Host Dependent User Option Setting

Example Adding a Host Independent User Option Setting

Getting a Value from a Specific Configuration File

Getting a Value Using Precedence Rules

Setting the Temporary Directory

Temporary Directory Configuration File Format

Setting or Removing Temporary Directory Values in Configuration Files: UpdateRegistry Command

Setting the Temporary Directory Using the GUI

Example of Removing a Host Dependent User Option Setting

For this example, user jsmith always uses host jshost to run HFSS 14.0. At some point, jsmith set the Autosave interval in the **General Options** dialog, **Project Options** tab to 1000 edits, and this value was written to the jsmith's host dependent user options config file for host jshost. Now, jsmith wants to remove this setting and return to the default value of 10. User jsmith may run the following command on host jshost to remove the Desktop/Settings/ProjectOptions/AutoSaveInterval option value from this config file:

```
UpdateRegistry -Delete -ProductName HFSS14.0
-RegistryKey Desktop/Settings/ProjectOptions/AutoSaveInterval
-RegistryLevel user_machine
```

Related Topics

User Options and the Update Registry Tool

Example Adding a Host Independent User Option Setting

Setting the Temporary Directory

Temporary Directory Configuration File Format

Setting the Temporary Directory Using the GUI

Example Adding a Host Independent User Option Setting

Consider the case in which there is no value set for the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting for all users for HFSS 14.0. The default is then 0 MB. User jsmith uses a variety of hosts and wants to be warned whenever disk space drops to 250 MB on any host. User jsmith may use the following command to set the Desktop/Settings/ProjectOptions/DiskLimitForAbort option value to 250 MB for all hosts:

```
UpdateRegistry -Set -ProductName HFSS14.0
-RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
-RegistryValue 250 -RegistryLevel user
```

Related Topics

User Options and the Update Registry Tool

Setting the Temporary Directory

Temporary Directory Configuration File Format

Setting the Temporary Directory Using the GUI

Getting a Value from a Specific Configuration File

In the previous example, the user jsmith may decide to check the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting in the host independent user configuration file before making any changes to this setting. The following command may be used to quickly view this setting for HFSS 14.0 before making the change:

```
UpdateRegistry -Get -ProductName HFSS15.0 -RegistryKey Desktop/
Settings/ProjectOptions/DiskLimitForAbort -RegistryLevel user
```

Getting a Value Using Precedence Rules

In may cases, the user is more interested in the value of a setting that will be applicable when running the product than in the setting in a single configuration file. If the -Get option is used with no -RegistryLevel specified, then the value reported is the value found in the highest precedence configuration file. If the user jsmith is interested in the highest precedence value for the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting, then the following command may be used to report this information:

```
UpdateRegistry -Get -ProductName HFSS14.0 -RegistryKey Desktop/
Settings/ProjectOptions/DiskLimitForAbort
```

Setting the Temporary Directory

As for other option settings, the Temporary Directory may be configured with an installation default value, as well as a host dependent default value, a host independent user specified value and a host dependent user specified value. The Temporary Directory settings are stored in different files from the other option settings. These files are located in the same directories as the configuration files for the other option settings. The following table shows the directories and files used to store the Temporary Directory settings.

Config File	Level Name	File Name	Linux Directory Pathname	Windows Directory Pathname
host dependent user specific Temporary Directory	user_machine	hostname.cfg	$HOME/AnsysEM/ApplicationPathAndVersion/config	MyDocuments/AnsysEM/ApplicationPathAndVersion/config
host independent user specific Temporary Directory	user	default.cfg		
host dependent default default Temporary Directory	install_machine	hostname.cfg	InstallationDirectory/config	InstallationDirectory/config
installation default Temporary Directory	install	default.cfg		

NOTES:

$HOME is the user's home directory on Linux

ApplicationPathAndVersion is the Application or Product name concatenated with the Product

Version

MyDocuments is the user's "My Documents" directory on Windows

InstallationDirectory is the directory where the product is installed

As for other options, the settings in these files have precedence in the following sequence: user_machine (highest precedence), user, install_machine, install (lowest precedence). The installer will create the file at the install level. By default, the other files are not created. If the Temporary Directory is set to an empty string in a configuration file, then that setting is ignored.

Related Topics

User Options and the Update Registry Tool

Example Adding a Host Independent User Option Setting

Temporary Directory Configuration File Format

Setting or Removing Temporary Directory Values in Configuration Files: UpdateRegistry Command

Setting the Temporary Directory Using the GUI

Temporary Directory Configuration File Format

This section describes the format of the Temporary Directory configuration files. The format is the same for files at all four levels: user_machine, user, install_machine, and install. These files are text files, so any text editor may be used to modify or create Temporary Directory configuration files.

An example temporary directory configuration file is shown below:

```
$begin 'Config'
tempdirectory='C:/TEMP/AnsysEM'
$end 'Config'
```

The temporary directory specified by this configuration file is C:/TEMP/AnsysEM.

Additional notes:

The string containing the pathname of the temporary directory must be enclosed in single quotes.

The forward slash "/" may be used as a directory separator on Windows and Linux. The back slash "\" may be used as a directory separator on Windows only.

The back slash "\" is used as an escape character in the tempdirectory string. That it, this character removes the special meaning of the following character.

The single quote character normally ends the tempdirectory string. The back slash may be used to remove this special meaning, and include a single quote in the string.

To use a back slash as a directory separator on Windows, it must be escaped. That is, a double back slash "\\" is used to denote a single directory separator.

On Windows, a UNC path normally begins with two back slash characters. In a tempdirectory string, each of these back slash characters must be doubled, so four consecutive back slashes "\\\\" are used in the config file.

UNC Example

Config file:

```
$begin 'Config'
tempdirectory='\\\\hostxyz\\TEMP\\abc'
$end 'Config'
```

Here hostxyz is a host with a sharename TEMP having subdirectory abc used as the Temporary Directory. This shows that four back slashes are required for UNC names and that back slashes used as directory separators must be doubled.

Single Quote Example

Config file:

```
$begin 'Config'
tempdirectory='C:/TEMP/ab\'cd'
$end 'Config'
```

Temporary directory is C:/TEMP/ab'cd. This shows how to include a single quote in a tempdirectory pathname. It also shows that forward slashes may be used as directory separators on Windows.

Related Topics

User Options and the Update Registry Tool

Example Adding a Host Independent User Option Setting

Example for Setting the Temporary Directory

Setting the Temporary Directory Using the GUI

Setting the Temporary Directory From the Command Line

Setting or Removing Temporary Directory Values in Configuration Files: UpdateRegistry Command

The UpdateRegistry command line tool, described above, may be used to view, add, change or remove the Temporary Directory setting from any of the Temporary Directory config files. The registry key for viewing or modifying the Temporary Directory is TempDirectory. The -Get, -Set, and -Delete options are valid for viewing a Temporary Directory setting, adding or changing a Temporary Directory setting, or deleting a Temporary Directory setting. The -GetKeys option does not list the Temporary Directory key.

Related Topics

User Options and the Update Registry Tool

Setting the Temporary Directory Using the GUI

As for other options, the Temporary Directory may be viewed or set using the Desktop GUI. In the **Tools>Options>General Options** dialog, the Temp Directory setting appears in the Directories group box on the **Project Options** tab. Activating the Override checkbox allows you to enter a desired directory pathname in the edit box or to click on the "..." button to bring up a directory file

browser dialog, from which you can select a temp directory. Values set in this manner are written to the user_machine level configuration file for the Temporary Directory. If the Override checkbox is unchecked, then when the OK button is pressed, the user_machine level setting for the Temporary Directory is changed to an empty string. This enables setting from the next highest precedence config file. The config file which provides the currently active Temporary Directory setting is shown under the Temp Directory edit box in the **Project Options** tab of the **General Options** dialog, if the Override checkbox is unchecked.

Related Topics

User Options and the Update Registry Tool

Example Adding a Host Independent User Option Setting

Example for Setting the Temporary Directory

Temporary Directory Configuration File Format

Batchoptions Command Line Examples

Running HFSS from the Command Line

Setting the Temporary Directory From the Command Line

The temporary directory may be set from the command line, using the -batchoptions command line option. See Running HFSS from the Command Line. The Batchoptions Command Line Examples section below includes examples that show how to set the Temporary Directory from the command line.

Related Topics

User Options and the Update Registry Tool

Example Adding a Host Independent User Option Setting

Example for Setting the Temporary Directory

Temporary Directory Configuration File Format

Batchoptions Command Line Examples

Running HFSS from as Command Line

Batchoptions Command Line Examples

The **-batchoptions** entries command line argument may be used to specify one or more batchoptions settings on the command line. To specify multiple entries using a single -batchoptions argument, the entries should be enclosed in double quotes. Alternatively, the batchoptions may be specified in a file using the **-batchoptions** <filename> command line argument format. In this case, the filename is an absolute or relative pathname of the file containing the batchoptions, as described above.

The two approaches may not be combined: either all batchoptions must be in a file or all batchoptions must be specified explicitly on the command line.

Batchoptions File Format

Example -BatchOptions with -Remote (Windows)

Example -Batchsolve with -Machinelist (Windows)

Example -Batchsolve with -Machinelist (Linux)

Example -Batchsolve for Local (Windows)

Related Topics

Running HFSS From the Command Line

High Performance Computing (HPC) Integration

Batchoptions File Format

An example batchoptions file is shown below:

```
$begin 'Config'
   'Desktop/ProjectDirectory'='C:/test/projects'
   'Desktop/Settings/NumberofProcessors'=2
   'HFSS/NumberCoresPerDistributedTask'=2
   $end 'Config'
```

Additional notes on the file format:

- The file may contain an arbitrary number of batchoption entries, one per line.
- Leading whitespace on each line is ignored. Spaces or tabs may be used to make the file more readable.
- Option pathname:

  The option pathname appears before the equal sign "=" on each line.

  Each option pathname must be enclosed in single quotes.

  Option pathnames are case insensitive.
- Option value:
  - The option value appears after the equal sign on each line.
  - Integral option values must not be enclosed in quotes.
  - All other options values are treated as strings, and must be enclosed in single quotes.
  - The forward slash "/" may be used as a directory separator on Windows and Linux. The back slash "\" may be used as a directory separator on Windows only.
  - The back slash "\" is used as an escape character in the value string. That it, this character removes the special meaning of the following character.

    The single quote character normally ends the value string. The back slash may be used to remove this special meaning, and include a single quote in the string.

    To use a back slash as a directory separator on Windows, it must be escaped. That is, a double back slash "\\" is used to denote a single directory separator.

Example -BatchOptions with -Remote (Windows)

In this example, we run a batch Maxwell analysis of project file project1.mxwl which contains a 3D design. We want all temporary files and directories created in directory C:\temp\maxwell instead of

using the installation default for the Temporary Directory. We decide that the analysis will be done on a remote host, at IP address 12.34.56.78. Because of limited memory on the remote host, we decide to run the analysis using only a single COM engine. Because the remote host has four cores, we decide to use four threads for multiprocessing, for both distributed and non-distributed parts of the analysis. We can use the **-Remote** option to specify that there will be a single remote COM engine.

Here is a sample command line for this analysis, where the project file \\somehost\projects\project1.mxwl is located in a shared directory specified using a UNC path:

```
maxwell -BatchSolve -Remote -Machinelist list=12.34.56.78
-batchoptions "TempDirectory='C:/temp/maxwell'
Desktop/Settings/ProjectOptions/NumberofProcessors=4
Maxwell3D/NumberCoresPerDistributedTask=4"
\\somehost\projects\project1.mxwl
```

An alternative is to use the -Distributed command line option. Because the -Machinelist list contains only one host, there is a single remote COM engine in this case, also.

```
maxwell -BatchSolve -Distributed -Machinelist list=12.34.56.78
-batchoptions "TempDirectory='C:\\temp\\maxwell' Desktop/
Settings/ProjectOptions/NumberofProcessors=4 Maxwell3D/
NumberCoresPerDistributedTask=4"
\\somehost\projects\project1.mxwl
```

The above command lines show that the forward slash "/" may be used as a directory separator on Windows. The back slash "\" may also be used as a directory separator on Windows, but it must be doubled to "\\" because the back slash is also an escape character.

Related Topics

Batchoptions Command Line Examples

Example -Batchsolve with -Machinelist (Windows)

Example -Batchsolve with -Machinelist (Linux)

Example -Batchsolve for Local (Windows)

Example -Batchsolve with -Machinelist (Windows)

Suppose that we want to run a batch HFSS analysis of project file project1.hfss. Because all of our hosts have multiples of 2 cores, we specify that we will use two threads for multiprocessing for both the distributed (NumberOfProcessorsDistributed) and non-distributed (NumberOfProcessors) parts of the job. The analysis contains a sweep that will be distributed across three hosts, adam, bill, and charlie. The hosts adam and bill have four cores each, so we run two distributed COM engines on each of these hosts, each using two threads. Host charlie has only two cores, so we specify only one distributed COM engine on this host. This COM engine will also use two threads. We specify a desired RAM limit of 6 GB and a maximum RAM limit of 8 GB for this analysis. The RAM limits are specified in KB, so the desired RAM limit is 6291456 KB, and the maximum RAM limit is 8388608 KB.

Here is a sample command line for this analysis, where the project file \\dennis\projects\project1.hfss is located in a shared directory specified using a UNC path:

```
hfss -BatchSolve -Distributed
-Machinelist list=adam,adam,bill,bill,charlie
-batchoptions "Hfss/Preferences/MemLimitHard=8388608
Hfss/Preferences/MemLimitSoft=6291456
Hfss/Preferences/NumberOfProcessors=2
Hfss/Preferences/NumberOfProcessorsDistributed=2"
\\dennis\projects\project1.hfss
```

Related Topics

Batchoptions Command Line Examples

Example -BatchOptions with -Remote (Windows)

Example -Batchsolve with -Machinelist (Linux)

Example -Batchsolve for Local (Windows)

Example -Batchsolve with -Machinelist (Linux)

In this example, we run a batch HFSS analysis of project file project2.hfss, which contains an HFSS-IE design. We have four identical hosts host1,host2, host3, and host4 for analysis, and each host has 4 cores. We do not use multiprocessing for the distributed analysis, so NumberOfProcessorsDistributed=1. As each host has four cores, we specify multiprocessing using 4 threads for the non-distributed part of the analysis, so NumberOfProcessors=4. Because we do not use multiprocessing for the distributed analysis, we will run four distributed COM engines on each host, with a singe core available for each engine. As in Example 1, we specify a desired RAM limit of 6 GB and a maximum RAM limit of 8 GB for this analysis. The RAM limits are specified in KB, so the desired RAM limit is 6291456 KB, and the maximum RAM limit is 8388608 KB.

Here is a sample command line for this analysis, where the project file /home/jsmith/projects/project2.hfss is located in a shared directory:

```
hfss -BatchSolve -Distributed
-Machinelist file=/home/jsmith/hosts/list2
-batchoptions "HFSS-IE/Preferences/MemLimitHard=8388608
HFSS-IE/Preferences/MemLimitSoft=6291456
HFSS-IE/Preferences/NumberOfProcessors=4
HFSS-IE/Preferences/NumberOfProcessorsDistributed=1"
/home/jsmith/projects/project2.hfss
```

For this example, the hostnames are in the text file /home/jsmith/hosts/list2. Here is the file contents:

```
host1
host1
host1
host1
host2
```

```
host2
host2
host2
host3
host3
host3
host3
host4
host4
host4
host4
```

Related Topics

Batchoptions Command Line Examples

Example -BatchOptions with -Remote (Windows)

Example -Batchsolve with -Machinelist (Windows)

Example -Batchsolve for Local (Windows)

Example -Batchsolve for Local (Windows)

In this example, we run a batch Designer analysis of project file testproject.adsn on the local host. We want all temporary files and directories created in directory C:\temp\designer instead of using the installation default for the Temporary Directory. Because the local host has four cores, we decide to use four threads for multiprocessing, for both distributed and non-distributed parts of the analysis.

Here is a sample command line for this analysis, where the project file \\host123\projects\testproject.adsn is located in a shared directory specified using a UNC path:

```
designer -BatchSolve -Local -batchoptions
"TempDirectory='C:/temp/designer'
'Planar EM/SolverOptions/NumProcessors'=4
'Planar EM/SolverOptions/NumProcessorsDistrib'=4"
\\host123\projects\testproject.adsn
```

Note that the batchoptions pathnames 'Planar EM/SolverOptions/NumProcessors' and 'Planar EM/SolverOptions/NumProcessorsDistrib' must be in single quotes because they both contain embedded spaces.

Related Topics

Batchoptions Command Line Examples

Example -BatchOptions with -Remote (Windows)

Example -Batchsolve with -Machinelist (Windows)

Example -Batchsolve with -Machinelist (Linux)

Batchoptions and Analysis Configurations in the Registry

Analysis configurations are used to specify machines, cores, and options for local, remote, and distributed analysis, including capabilities that are enabled by HPC licenses.

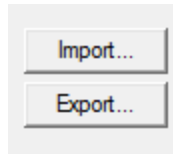
How Analysis Configurations are Stored in the Registry

There are several major changes related to how configurations are organized. A configuration contains information in addition to the machines that are to be used in an analysis. For example, the number of tasks and the number of cores to allocate to the analysis for each machine in the list is also included in the configuration. Other options may be specific to one product or design type, for example, certain job distribution types and memory limits. In order to support different options for different design types, analysis configurations are now associated with the design type. Previously, configurations applied to all design types within a given product. Now, it is possible to create independent configurations with the same name, but associated with different design types, and that the appropriate configuration for the design being solved will be used.

Copying a Configuration from one Design Type or Product to Another

To copy a configuration from one design type (or product) to another:

1. On the **Configurations** tab of the **HPC and Analysis Options** dialog, use the **Export…** button to export the configuration to a file.



2. Switch to the destination design type (or product) and use the **Import…** button to import the configuration data.

   Any data that is not applicable to the destination design type is ignored; any settings present in the destination design type that were not present in the source configuration will be assigned default values. The user may then edit the copy, as desired.

Using HPC and Analysis for Configurations

Due to the complexity of the registry values for the configurations we do not recommend directly editing these values using the UpdateRegistry tool. Instead, use the **HPC and Analysis Options** dialog to edit or create a configuration. (See Setting HPC and Analysis Options.) Configurations created or edited using the GUI are stored in the user_machine level of the registry. A configuration may be created for one of the other registry levels using several steps.

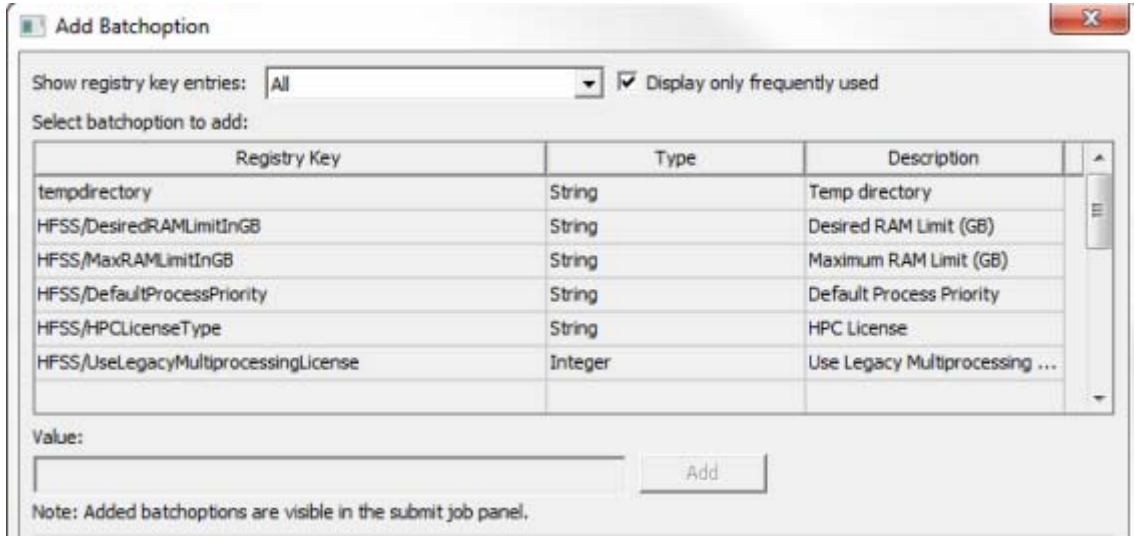
1. First, create the configuration using the Analysis Configurations GUI, then export the configuration to a file.
2. Next, delete the configuration using the GUI so that it will not be present in the user_machine level. Exit the GUI.
3. Use the UpdateRegistry tool to import the data into the desired registry level using the -FromFile option to specify the file exported via the GUI, and using the -RegistryLevel option to

specify the registry level where the configuration is to be stored. For example, an administrator may use this approach to create a configuration at the install level that may be used by any user on any machine.

Batch Options Corresponding to Configurations

The batch options corresponding to configurations are affected by the changes to the organization of configuration information. A number of batch options are changed. For example, the batch options settings HFSS/Preferences/NumberOfProcessors and HFSS/Preferences/NumberOfProcessorsDistributed are no longer used. The new batch option setting, HFSS/NumCoresPerDistributedTask, resembles the obsolete HFSS/Preferences/NumberOfProcessorsDistributed setting, but there is no equivalent to the obsolete HFSS/Preferences/NumberOfProcessors setting. The non-distributed portion of the analysis uses all cores that were allocated for distributed tasks on the given machine.

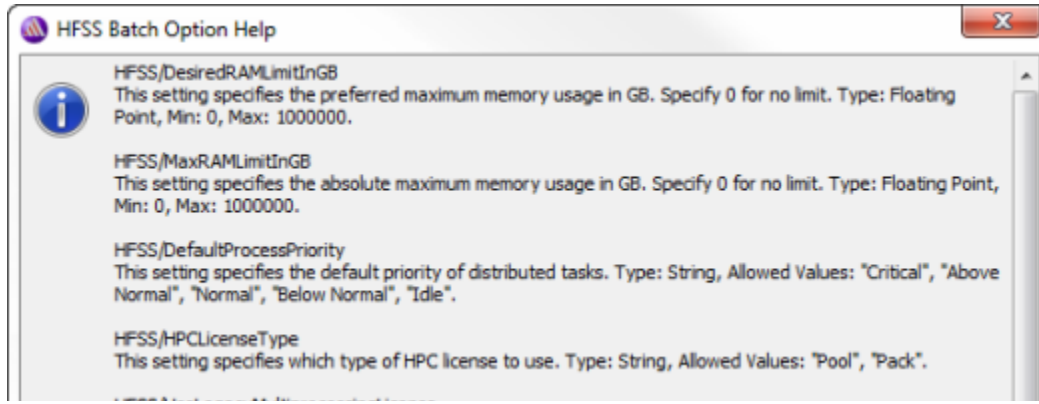
When you submit jobs to a cluster, you can specify batch options using the job submission GUI.



When using the GUI, you can select the batch options from a list, making typographical errors less common. For the most commonly used batch options, there is detailed information about the allowed values.

To assist users who need to specify batch options and are unable to use the job submission GUI, a new help option is added to all products. If the product is launched with the -batchoptionhelp com-

mand line argument, a message box is displayed which lists and describes the most common batch options specific to the product.



Setting Analysis Configurations Using the User Interface

The Desktop User Interface may be used to select the configuration to be used for each design type. Each configuration is identified by a unique name. These settings may be viewed and modified using the **HPC and Analysis Options** dialog and modified using the **Analysis Configuration** dialog.

See Setting HPC and Analysis Options.

Working with Variables

A variable is a numerical value, mathematical expression, or mathematical function that can be assigned to a design parameter in HFSS. You can assign a variable to any dimension or material property, or output value. Variables are useful in the following situations:

- You expect to change a parameter often.
- You expect to use the same parameter value often.
- You intend to run a parametric analysis, in which you specify a series of variable values within a range to solve.
- You intend to optimize a parameter value by running an optimization analysis.
- You intend to run a convergence on an output variable.
- You intend to calculate derivatives for variables.
- You intend to animate a plot against a variable.

There are two types of variables in HFSS:

Project Variables A project variable can be assigned to any parameter value in the HFSS project in which it was created. HFSS differentiates project variables from other types of variables by prefixing the variable name with the following symbol: **$**. You can manually include the symbol **$** in the project variable's name, or HFSS will automatically append the project variable's name after you define the variable. Project Variables can be designated as Design, ArrayIndex or Separator variables but not as Post Processing variables.

Design Variables A design variable can be assigned to any parameter value in the HFSS design in which it was created. From the Design Variables **Properties** dialog, you can Add, Add Array, Edit, or Remove Design variables. Design Variables can be designated as Design, ArrayIndex or Separator variables and as Processing Variables.

Related Topics

Defining an Expression

Defining Mathematical Functions

Assigning Variables

Specify Expressions for Adaptive Convergence

Specifying Output Variables

Using Optimetrics for Design Analysis

Choosing a Variable to Optimize

Selecting Objects by Variable

Exporting Variables for Documentation

Viewing Variables

Adding a Project Variable

A project variable can be assigned to a parameter value in the HFSS project in which it was created. HFSS differentiates project variables from other types of variables by prefixing the variable name with the following symbol: **$**. You can manually include the symbol **$** in the project variable's name when you create it, or HFSS will automatically append the project variable's name with the symbol after you define the variable.

1. Click **Project**>**Project Variables**.
   - Alternatively, right-click the project name in the project tree, and then click **Project Variables** on the shortcut menu.

   The **Properties** dialog box appears.
2. Under the **Project Variables** tab, click **Add**.

   The **Add Property** dialog box appears.
3. In the **Name** text box, type the name of the variable.

   Project variable names must start with the symbol **$** followed by a letter. Variable names may include alphanumeric characters and underscores (_). The names of intrinsic functions and the pre-defined constant pi (π) cannot be used as variable names.

   You can sort the project variables by clicking on the Name column header. By default, variables are sorted in original order. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.
4. Select a radio button for the variable use:

Selected Use	Setable Properties
Variable	Unit Type, Units, Value.
Separator	Value
Array Index Variable	Associate Array variable, Value

   Each selection affects the settable options.
5. For Project Variables in the **Unit Type** text box you can use the drop down menu to select from the list of available unit types. "None" is the default.

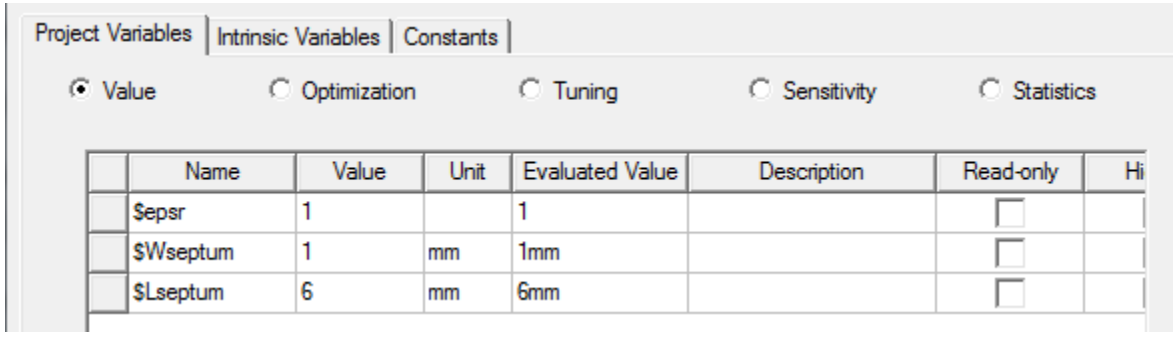
   When you select a Unit Type, the choices in drop down menu for the Units text box adapt to that unit type. For example, selecting Length as the Unit Type causes the Unit menu to show a range of metric and English units for length. Similarly, if you select the Unit Type as Resistance, the Units drop down lists a range of standard Ohm units.
6. In the **Value** text box, type the quantity that the variable represents. If you did not specify the Unit Type and Unit, you can also optionally, include the units of measurement.

Warning If you include the variable's units in its definition (in the **Value** text box), do not include the variable's units when you enter the variable name for a parameter value.

The quantity can be a numerical value, a mathematical expression, or a mathematical function. The quantity entered will be the *current,* (or *default) value* for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent variable will automatically change to those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

7. Click **OK**.

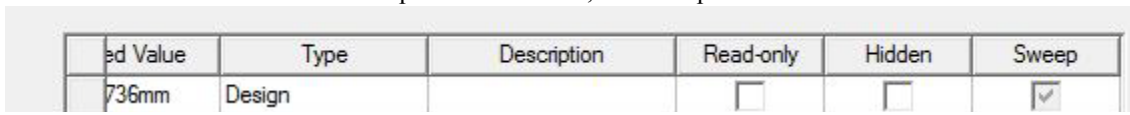
   You return to the **Properties** dialog box. The new variable and its value are listed in the table. If the value is an expression, the evaluated value is shown. Updating the expression also changes the evaluated value display. Any dependent variables also have evaluated values changed.



8. Optionally, type a description of the variable in the **Description** text box.
9. Optionally, select **Read Only**. The variable's name, value, unit, and description cannot be modified when **Read Only** is selected.
10. Optionally, select **Hidden**. If you clear the **Show Hidden** option, the hidden variable will not appear in the **Properties** dialog box.
11. You can also designate a variable as Sweep. You may need to use the scroll bar or resize the dialog to view the check boxes.

The Sweep checkbox lets you designate variables to include in solution indexing as a way to permit faster post processing. Variables with Sweep unchecked are not used in solution indexing. If a solution exists, checking or unchecking a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click OK to dismiss the warning dialog.

If a variable has dependent variables, the Sweep checkbox is disabled and cannot be changed.



12. Click **OK**.

The new variable can now be assigned to a parameter value in the project in which it was created.

Related Topics

Deleting Project Variables

Defining an Expression

Defining Mathematical Functions

Exporting Variables for Documentation

Sweeping a Variable in a Report

Intrinsic Variables in a Project

The **Properties** dialog for the project also lists the intrinsic variables under the **Intrinsic Variables** tab.

Name	Unit	Description
_I1 to _I9	mA	Terminal current in user-defined model (A).
_t		Variable to define parametric equation-based curve.
_u_,_v		Variable to define parametric equation-based surface.
_V1 to _V9	mV	Port Voltage in user defined model (V).
Ang	Ang	Angle. Post Processing variable
Distance	mm	
Electrical Degree	deg	Electric degree of the rotating machine. Not settable by user.
F	GHz	Frequency of circuit system analysis (Hz).
F1, F2, F3	GHz	Frequency tones 1, 2, 3 in harmonic balance analysis. (Hz).
Fnoi	GHz	Offset noise frequency in harmonic balance noise analysis.
Freq	GHz	Frequency. Post processing variable, not settable by the user.
Ia, Ib	mA	Post processing variables, not settable by the user.
Index		Identifier for a data point.
IWave Phi, IWave Theta	deg	Incident wave spherical coordinate variables. Phi is the angle from the origin in the z direction and Theta the angle from the x-axis.

Name	Unit	Description
Normalized Deformation, Distance		Post processing variables, not settable by the user.
OP	mW	Post processing variables, not settable by the user.
Phase	deg	Angle of a complex number.
Phi	deg	Phi is the angle from the origin in the z direction
R	mm	R is the cylindrical coordinate system variable..
Rho		Rho is the spherical coordinate system variable.
Rspeed	rpm	Speed of the machine.
Spectrum	GHz	Post processing variable.
Temp	cell	Analysis temperature (deg)
Theta	deg	Coordinate0, the angle from the x-axis, not settable by the user.
Time	ns	Time point in transient analysis
Vac, Vbe, Vce, Vds, Vgs	mV	Post processing variables, not settable by the user.
X,Y,Z	mm	Point coordinate variables in the modeler.
ZAng	deg	A spherical coordinate system variable
ZRho		A spherical coordinate system variable.

Deleting Project Variables

To delete a project variable:

1. Remove all references to the variable in the project, including dependent variables.
2. **Save** the project to erase the command history.
3. Click **Project>Project Variables** to display the **Properties** dialog with list of variables.
4. Select the variable and click **Remove** and **OK**.

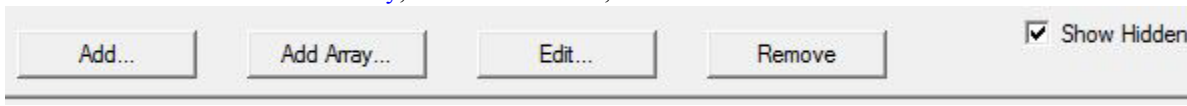
Adding a Design Variable

A design variable is associated with an HFSS design. A design variable can be assigned to a parameter value in the HFSS design in which it was created. You can also add a variable defined with an array of values.

1. Click **HFSS or HFSS-IE>Design Properties**.
   - Alternatively, right-click the design name in the project tree, and then click **Design Properties** on the shortcut menu.

   The Local Variables **Properties** dialog box appears. From the Properties dialog you can Add, Add Array, Edit.. or Remove variables. This section describes **Add**. Use the links for descrip-

tions of Add Array, .Edit and Remove,



Any existing Design variables are listed in the **Properties** dialog with the name followed by cells for Value, Unit, Evaluated Value, Type, Description, and Read-only and Hidden check boxes. A Show Hidden checkbox on the lower right of the Properties dialog controls the appearance of any Hidden variables.

Initially, you leave the radio button with Value selected until you have defined a variable. The other radio buttons let you enable defined variables for Optimization, Tuning, Sensitivity or Statistics. Selecting one of these radio buttons add a new column to the Variable definition row for which you can check or uncheck Include for regular variables for that kind of Optimetrics simulation. For further discussion, see Optimetrics.

2. Click **Add**.

   The **Add Property** dialog box appears.

3. In the **Name** text box, type the name of the variable.

   Variable names must start with a letter, and may include alphanumeric characters and underscores (_). The names of intrinsic functions and the pre-defined constant pi (π) cannot be used as variable names.

   You can sort the project variables by clicking on the Name column header. By default, variables are sorted in original order. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

4. Select a radio button for the variable use:

Selected Use	Setable Properties
Variable	Unit Type, Units, Value
Array Index Variable	Associate Array variable, Value
Separator	Value
Post Processing Variable	Unit Type, Units, Value

   Each selection affects the settable options.

5. In the **Unit Type** text box you can use the drop down menu to select from the list of available unit types. "None" is the default.

   When you select a Unit Type, the choices in drop down menu for the Units text box adapt to that unit type. For example, selecting Length as the Unit Type causes the Unit menu to show a range of metric and English units for length. Similarly, if you select the Unit Type as Resistance, the Units drop down lists a range of standard Ohm units.

6. In the **Value** text box, type the quantity that the variable represents. Optionally, include the

units of measurement.

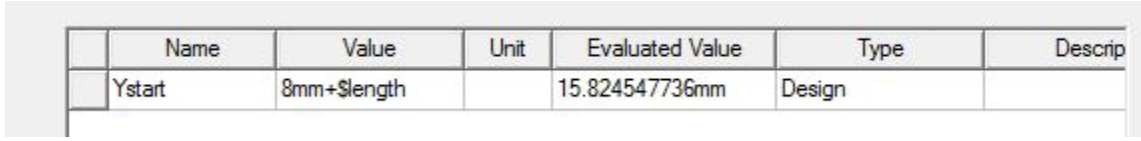
Note If you include the variable's units in its definition (in the **Value** text box), do not include the variable's units when you enter the variable name for a parameter value.

The quantity can be a numerical value, a mathematical expression, or a mathematical function. The quantity entered will be the *current* (or *default value)* for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent variable will automatically change to those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

Note Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

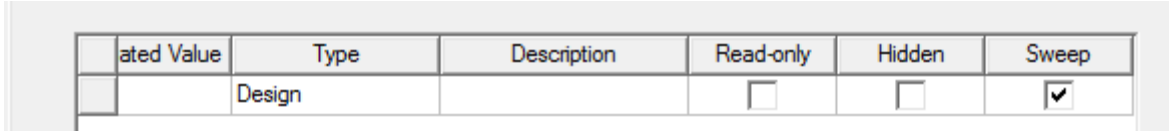
7. Click **OK**.

   You return to the **Properties** dialog box. The new variable and its value are listed in the table. If the value is an expression, the evaluated value is shown. Updating the expression also changes the evaluated value display. Any dependent variables also have evaluated values changed.



	Name	Value	Unit	Evaluated Value	Type	Descrip
	Ystart	8mm+$length		15.824547736mm	Design	

8. Optionally, type a description of the variable in the **Description** text box.
9. You can designate a variable as Read-only, Hidden, or Sweep. You may need to use the scroll bar or resize the dialog to view the check boxes.



	ated Value	Type	Description	Read-only	Hidden	Sweep
		Design		☐	☐	☑

The Sweep checkbox lets you designate variables to include in solution indexing as a way to permit faster post processing. Variables with Sweep unchecked are not used in solution indexing. If a solution exists, checking or unchecking a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click OK to dismiss the warning dialog.

If a variable has dependent variables, the Sweep checkbox is disabled and cannot be changed.

	ed Value	Type	Description	Read-only	Hidden	Sweep
	736mm	Design		☐	☐	☑

10. Click **OK**.

The new variable can now be assigned to a parameter value in the design in which it was created.

Related Topics

Add Array of Values for a Design Variable

Editing a Design Variable

Deleting Design Variables

Deleting Project Variables

Defining an Expression

Defining Mathematical Functions

Exporting Variables for Documentation

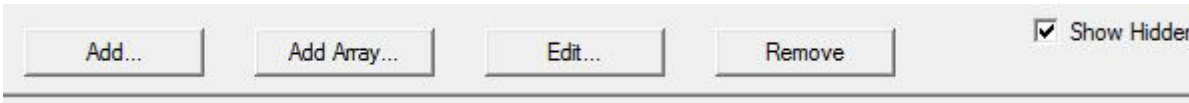
Sweeping a Variable in a Report

Add Array of Values for a Design Variable

A design variable is associated with an HFSS design. You can also add a variable defined with an array of values.

1. Click **HFSS or HFSS-IE**>**Design Properties**.
   - Alternatively, right-click the design name in the project tree, and then click **Design Properties** on the shortcut menu.

   The Local Variables **Properties** dialog box appears. From the Properties dialog you can Add, Add Array, Edit... or Remove variables. Select a variable to enable the Edit... or Remove buttons. This section describes Add Array. Use the links for descriptions of Add,.Edit and Remove,



   Any existing Design variables are listed in the **Properties** dialog with the name followed by cells for Value, Unit, Evaluated Value, Type, Description, and Read-only and Hidden check boxes. A Show Hidden checkbox on the lower right of the Properties dialog controls the appearance of any Hidden variables.

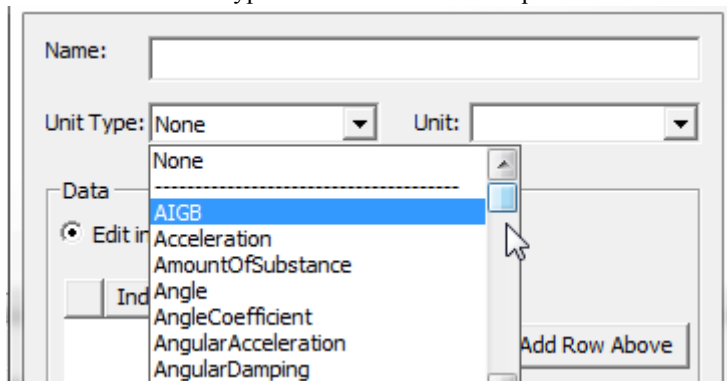
   Initially, you leave the radio button with Value selected until you have defined a variable. The other radio buttons let you enable regular variables for Optimization, Tuning, Sensitivity or Statistics. Array variables cannot be enabled for Optimetrics use.

2. Click the **Add Array...**. button.

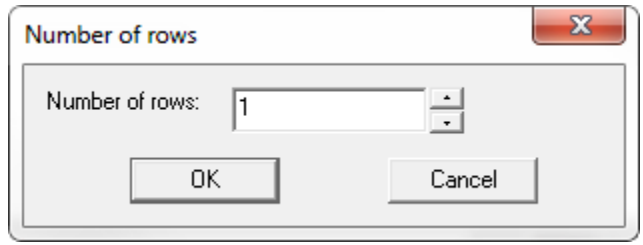
   The **Add Array** dialog displays.

3. Specify a variable Name in the text field.

4. Select a Unit Type and Units from the drop down menus.



5. To specify the array with Edit in Grid Selected, you can begin by clicking the **Append Rows...** button to display the **Number of Rows** dialog. (For Edit in plain text field, see below.)



6. Specify a value and click OK.

   This displays a list of indexed data rows in the **Add Array** dialog. You can type any data value in the cells. If you enter alphanumeric text in a cell it must be delimited by double quotes. You can edit the rows relative a row selection by clicking buttons to Add Row Above, Add Row Below, or Remove Row. All cells must contain a value.

7. When you have completed the array, click **OK** to close the dialog,

   The Array variable is listed in the **Design Properties** dialog as a Local Variable. The array variable value field includes the array contents in brackets with the unindexed data values delimited by commas.

   If you elected to edit the array Edit in plain text field in the **Add Array** dialog, the bracketed and comma delimited format is used.

Related Topics

Design Variables

Editing a Design Variable

Deleting Design Variables

Deleting Project Variables

Defining an Expression

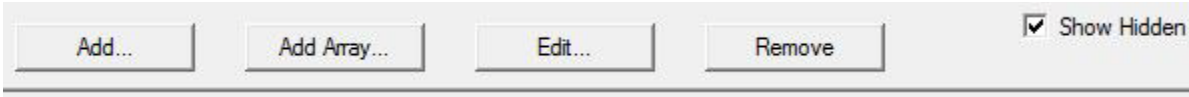
Defining Mathematical Functions

Editing a Design Variable

A design variable is associated with an HFSS design. A design variable can be assigned to a parameter value in the HFSS design in which it was created. You can also add a variable defined with an array of values.

1. Click **HFSS or HFSS-IE>Design Properties**.
   - Alternatively, right-click the design name in the project tree, and then click **Design Properties** on the shortcut menu.

   The Local Variables **Properties** dialog box appears. From the Properties dialog you can Add, Add Array, Edit.. or Remove variables. This section describes **Edit**. Use the links for descriptions of Add, Add Array and Remove,



   Any existing Design variables are listed in the **Properties** dialog with the name followed by cells for Value, Unit, Evaluated Value, Type, Description, and Read-only and Hidden check boxes. A Show Hidden checkbox on the lower right of the Properties dialog controls the appearance of any Hidden variables.

   Initially, you leave the radio button with Value selected until you have defined a variable. The other radio buttons let you enable defined variables for Optimization, Tuning, Sensitivity or Statistics. Selecting one of these radio buttons add a new column to the Variable definition row for which you can check or uncheck Include a regular variable for that kind of Optimetrics simulation. Array variables cannot be Enabled for Optimetrics. For further discussion, see Optimetrics.
2. Select a variable to highlight it.
3. Click the **Edit** button.

   For regular variables, the **Add Property** dialog for that variable opens, and for array variables, the **Edit Array** variable dialog for that variable opens.
4. Complete the edits and **OK** the dialog to apply them.

Related Topics

Design Variables

Exporting Variables for Documentation

Deleting Design Variables

To delete a design variable:

1. Remove all references to the variable in the design, including dependent variables.
2. **Save** the project to erase the command history.
3. Click **HFSS or HFSS-IE>Design Properties** to display the **Properties** dialog with list of local variables.

4. Select the variable and click **Remove** and **OK**.

Related Topics

Design Variables

Deleting Project Variables

Adding Datasets

Datasets are collections of plotted data points that can be extrapolated into an equation based on the piecewise linear makeup of the plot. Each plot consists of straight line segments whose vertices represent their end points. A curve is fitted to the segments of the plot and an expression is derived from the curve that best fits the segmented plot. The created expression is then used in piecewise linear intrinsic functions. You can add datasets at either the Project Level or the Design level. They can be for various purposes, including to define frequency dependent port impedances or frequency dependent global variables, and boundary definitions.

Design level datasets can be used in geometry entities like part commands, coordinate systems, points, and planes. Datasets do not work with equation based surfaces or curves. Datasets can be used in directly with piecewise linear functions in expressions or indirectly through variables that can refer to the dataset. Datasets can be used in the following operations.

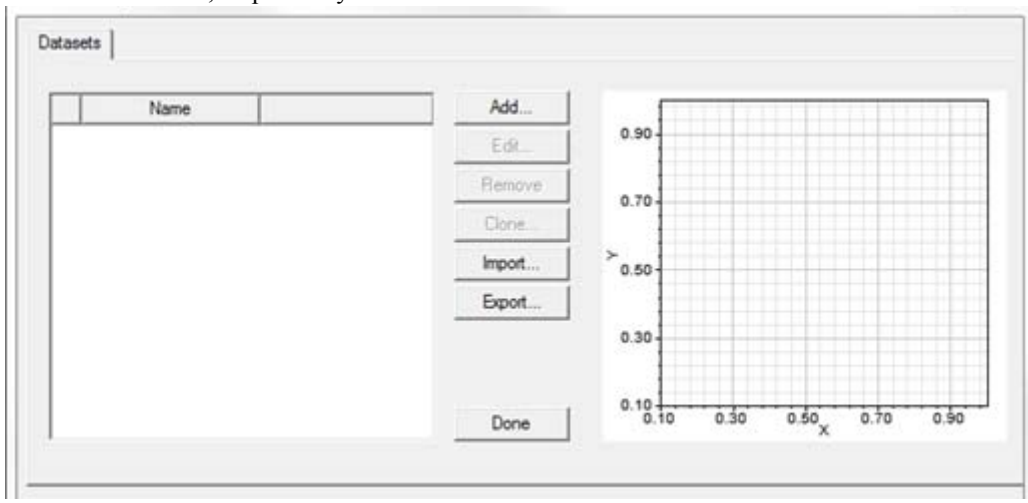
- Creation or edit of geometry.

  When a geometry uses a dataset directly, edit dataset invalidates the solution. When a geometry uses a variable that is defined by dataset, edit dataset does not invalidate the solution.
- Animation based on a variable which can index in datasets.
- Copy/paste of geometry. If a part refers to a design dataset, it will be pasted to destination design.

1. For a Project level dataset, click **Project**>**Datasets**. For a Design level dataset, click **HFSS**> or **HFSS-IE**>**Design Datasets**.

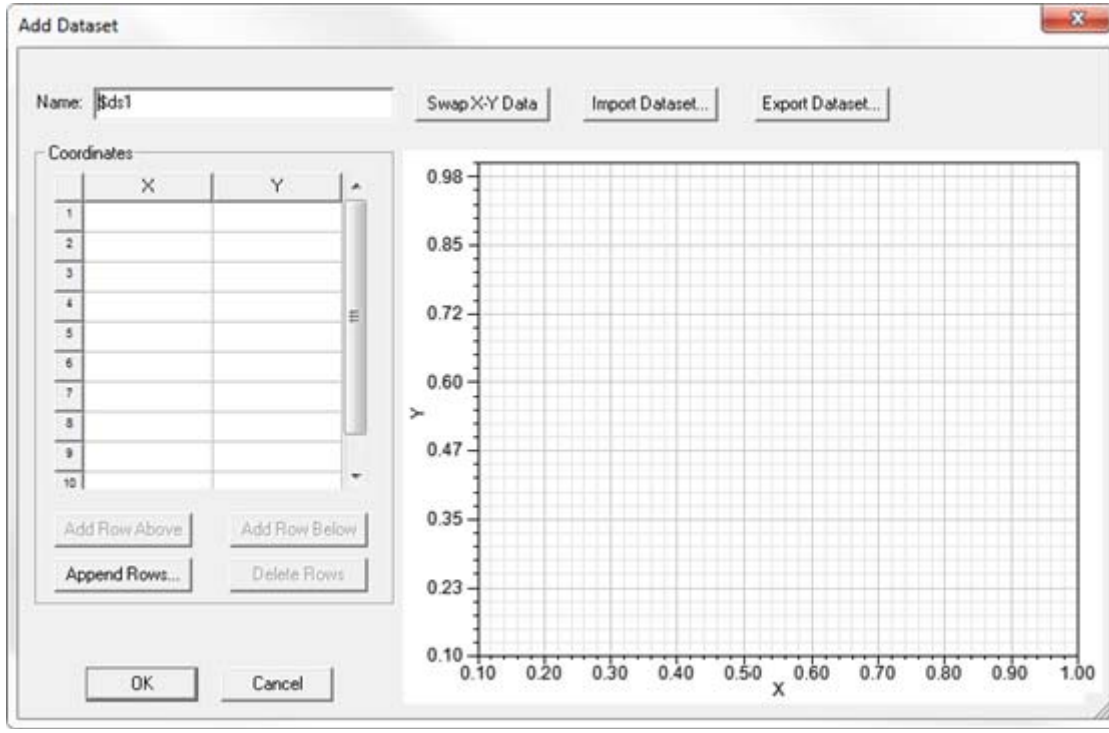
   The **Datasets** dialog box appears. This lists any existing datasets for the Project or Design

level, respectively.



2. Click **Add**.

   The **Add Dataset** dialog box appears.



The dialog contains fields for the Dataset name, and a table for x- and y- coordinates. It contains a graphic display that draws a line for the coordinates you add. It also includes buttons for

the following functions:

- **Swap X-Y Data** - this swaps the x- and Y- coordinates and adjusts the graphical display.
- **Import Dataset** - this provides a way to import data sets from an external source. The format is a tab separated points file. Clicking the button opens a file browser window.
- **Export Dataset** -- this provides a way to export the current dataset to a tab separated points file. Clicking the button opens a file browser window.
- **Add Row Above** - adds a new row to the table above the selected row.
- **Add Row Below** - adds a new row to the table below the selected row/
- **Append Row** - opens a dialog that lets you specify a number of rows to add to the table.
- **Delete Row** - deletes the selected row or rows.

3. Optionally, type a name other than the default for the dataset in the **Name** text box.
4. Enter the x- and y- coordinates by one of the following methods
   - **Import Dataset**
   - Type the x- and y-coordinates for the first data point in the row labeled **1**.Type the x- and y-coordinates for the remaining data points in the dataset using the same method.

     After you type a point's coordinates and move to the next row, the point is added to the plot, adjusting the display with each newly entered point.
5. When you are finished entering the data point coordinates, click **OK**.

The dataset plot is extrapolated into an expression that can be used in parametric analyses, boundary definitions, or assigned to a material property value.

Related Topics

Adding a Design Variable

Modifying Datasets

Using Piecewise Linear Functions in Expressions

Using Dataset Expressions

Modifying Datasets

1. For Project level datasets, click **Project**>**Datasets**. For Design level datasets, click **HFSS or HFSS-IE>Design Datasets**.

   The **Datasets** dialog box appears.
2. Click the dataset name you want to modify, and then click **Edit**.

   The **Edit Dataset** dialog box appears.
3. Optionally, type a name other than the default for the dataset in the **Name** text box.
4. Type new values for the data points as desired.

   The plot is adjusted to reflect the revised data points.
5. When you are finished entering the data point coordinates, click **OK**.

Related Topics

Adding Datasets

Adding a Design Variable

Modifying Datasets

Using Piecewise Linear Functions in Expressions

Using Dataset Expressions

Defining Mathematical Functions

A mathematical function is an expression that references another defined variable. A function's definition can include both expressions and variables.

The following mathematical functions may be used to define expressions:

Basic Functions	/, +, -, *****, mod (modulus), ****** (exponentiation), - (Unary minus), == (equals), **!** (not), **!=** (not equals), > (greater than), < (less than), >= (greater than equals), <= (less than equals), **&&** (logical and), || (logical or)
Intrinsic functions	**if**, **sqn**, **abs**, **exp**, **pow**, **ln** (natural log), **log** (log to the base 10), **lg** (log to the base 2), **sqrt**, **floor**, **ceil**, **round**, **rand** (returns a random number between 0 and 1), **deg**, **rad**
Trigonometric expressions	**sin**, **cos**, **tan**, **asin**, **acos**, **atan**, **sinh**, **cosh**, **tanh**

The predefined variables X, Y, Z, Phi, Theta, R, and Rho must be entered as such. X, Y, and Z are the rectangular (cartesian) coordinates. Phi, Theta, and Rho are the spherical coordinates. R is the cylindrical radius, and Rho is the spherical radius.

If you do not specify units, all trigonometric expressions expect their arguments to be in radians, and the inverse trigonometric functions' return values are in radians. If you want to use degrees, you must supply the unit name **deg**. When the argument to a trigonometric expression is a variable, the units are assumed to be radians. These function names are reserved and may not be used as variable names.

As far as expression evaluation is concerned: units are conversion factors (that is, from the given unit to SI). Note also that the evaluated value of an expression) is always interpreted as in SI units.

Related Topics

Setting Coordinate Systems

Expressing Cartesian, Cylindrical or Spherical Coordinates

Defining an Expression

Expressions are mathematical descriptions that typically contain intrinsic functions, such as sin(x), and arithmetic operators, such as +, -, *, and /, well as defined variables. For example, you could define: x_size = 1mm, y_size = x_size + sin(x_size). Defining one variable in terms of another makes a dependent variable. Dependent variables, though useful in many situations, cannot be the subject of optimization, sensitivity analysis, tuning, or statistical analysis.

The Constants tab of the **Project Variables** dialog lists the available pre-defined constants. These may not be reassigned a new value.

Name	Value	Description
abs0	-273.15	
Boltz	1.3806503E-023	Boltzmann constant (J/K)
c0	299792458	Speed of light in vacuum (m/s)
elecq	8.854187817e-012	Permittivity of vacuum (F/m)
eta	376.730313461	Electron Charge (C)
g0	9.80665	
mathE	2.718281828	.
pi	3.14159265358979	Ratio of circle circumference
planck	6.6260755e-034	.
u0	1.25663706143582e-06	Permeability of vacuum (H/m)

Numerical values may be entered in ANSYS's shorthand for scientific notation. For example, $5x10^7$ could be entered as **5e7**.

Related Topics

Defining Mathematical Functions

Using Valid Operators for Expressions

Using Intrinsic Functions for Expressions

Using Piecewise Linear Functions for Expressions

Using Dataset Expressions

Using Valid Operators for Expressions

The operators that can be used to define an expression or function have a sequence in which they will be performed. The following list shows both the valid operators and the sequence in which they are accepted (listed in decreasing precedence):

( )	parenthesis	1
!	not	2
^ (or **)	exponentiation (If you use "**" for exponentiation, as in previous software versions, it is automatically changed to "^".)	3
-	unary minus	4

*	multiplication	5
/	division	5
+	addition	6
-	subtraction	6
==	equals	7
!=	not equals	7
>	greater than	7
<	less than	7
>=	greater than or equal to	7
<=	less than or equal to	7
&&	logic and	8
||	logic or	8

Related Topics

Defining an Expression

Using Intrinsic Functions in Expressions

HFSS recognizes a set of intrinsic trigonometric and mathematical functions that can be used to define expressions. Intrinsic function names are reserved, and may not be used as variable names.

The following intrinsic functions may be used to define expressions:

Function	Description	Syntax
abs	Absolute value (|x|)	abs(x)
sin	Sine	sin(x)
cos	Cosine	cos(x)
tan	Tangent	tan(x)
asin	Arcsine	asin(x)
acos	Arccosine	acos(x)
atan	Arctangent (in range of -90 to 90 degrees)	atan(x)
atan2	Arctangent (in range of -180 to 180 degrees)	atan2(y,x)
asinh	Hyperbolic Arcsine	asinh(x)
atanh	Hyperbolic Arctangent	atanh(x)
sinh	Hyperbolic Sine	sinh(x)

cosh	Hyperbolic Cosine	cosh(x)
tanh	Hyperbolic Tangent	tanh(x)
even	Returns 1 if integer part of the number is even; returns 0 otherwise.	even(x)
odd	Returns 1 if integer part of the number is odd; returns 0 otherwise.	odd(x)
sgn	Sign extraction	sgn(x)
exp	Exponential (e^x)	exp(x)
pow	Raise to power (x^y)	pow(x,y)
if	If	if(cond_exp,true_exp, false_exp)
pwl	Piecewise Linear	pwl(dataset_exp, variable)
pwl_periodic	Piecewise Linear for periodic extrapolation on x	pwl_periodic(dataset_exp, variable)
sqrt	Square Root	sqrt(x)
ln	Natural Logarithm (The "log" function has been discontinued. If you use "log(x)" in an expression, the software automatically changes it to "ln(x)".)	ln(x)
log10	Logarithm base 10	log10(x)
int	Truncated integer function	int(x)
nint	Nearest integer	nint(x)
max	Maximum value of two parameters	max(x,y)
min	Minimum value of two parameters	min(x,y)
mod	Modulus	mod(x,y)
rem	Fractional part (remainder)	rem(x.y)

Note If you do not specify units, all trigonometric functions interpret their arguments as radians. Likewise, inverse trigonometric functions' return values are in given in radians. When the argument to a trigonometric expression is a variable, the units are assumed to be radians. If you want values interpreted in degrees, supply the argument with the unit name **deg**.

Related Topics

Defining an Expression

Using Piecewise Linear Functions in Expressions

The following piecewise linear intrinsic functions are accepted in expressions:

```
pwl (dataset_expression, variable)
pwl_periodic (dataset_expression, variable)
```

The **pwl** function interpolates along the x-axis and returns a corresponding y value. The **pwl_periodic** function also interpolates along the x-axis but periodically.

Related Topics

Adding Datasets

Adding a Design Variable

Modifying Datasets

Using Dataset Expressions

Using Dataset Expressions

In the simplest sense a dataset is a collection of data. It can take the following form:

$ds1((x$_0$,y$_0$), ..., (x$_n$,y$_n$))

A dataset (such as $ds1) once created may be used as the first parameter to piecewise linear (**pwl** and **pwl_periodic**) functions, and may also be assigned to variables, in which case the variable may be used as the second parameter to **pwl** and **pwl_periodic** functions.

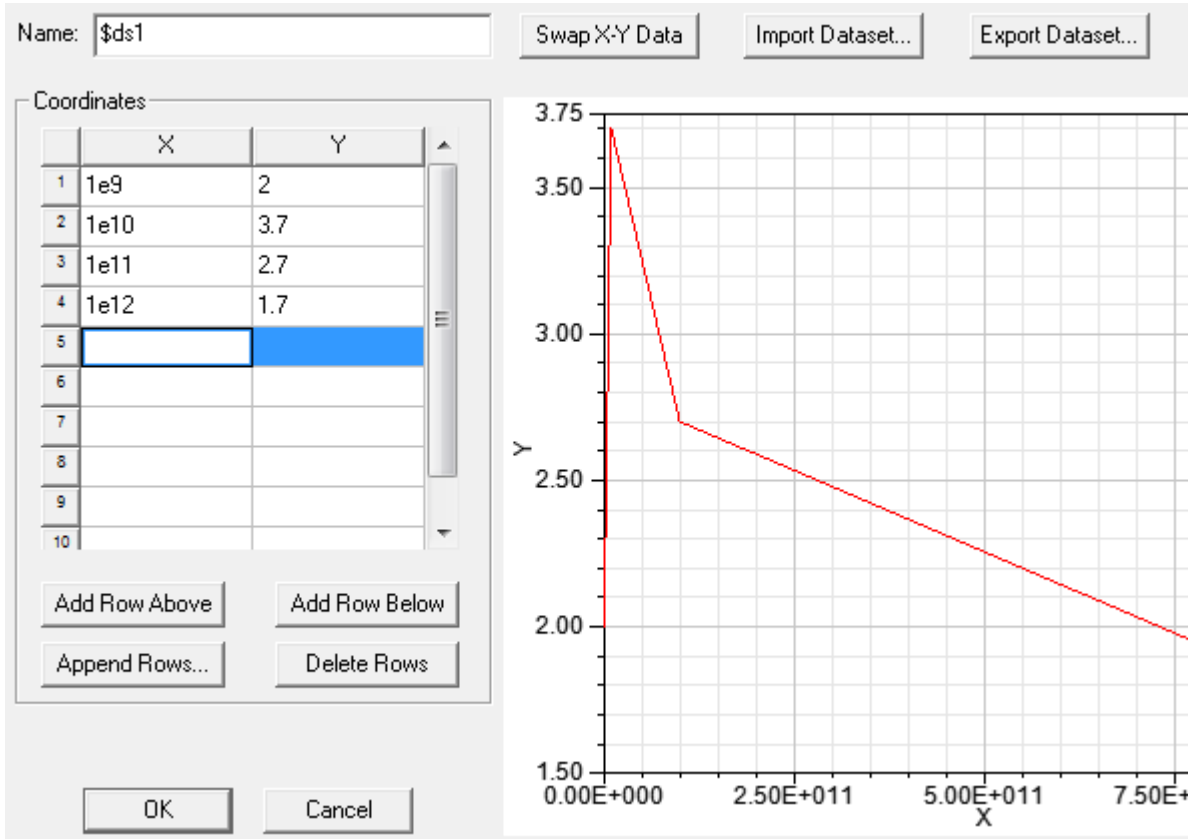
A dataset is generated using a series of points in a plot on the **Datasets** dialog box. Each plot consists of straight line segments whose vertices represent their end points. A curve is fitted to the segments of the plot. This curve which best fits the segmented plot consists of the co-ordinates used in the creation of the dataset.

The dataset thus created may be used in the piecewise linear intrinsic functions.

Note The following example aims to only inform how to perform the task of using a dataset. The values used are arbitrary.

1. Click **HFSS**> **Design Datasets**
2. Click **Add** and set the **Name** field and **Coordinates** as desired and click **OK**.

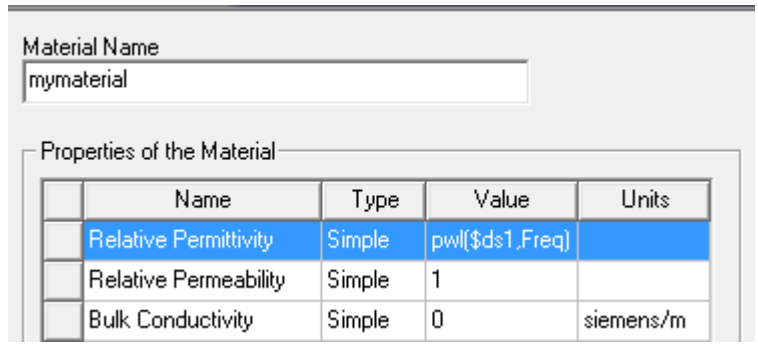
   The dataset is created.



Note By default the $ sign is assigned to the dataset even if you do not use one while naming it.

3. Go to **Tools**>**Edit Configured Libraries**>**Materials** to open the **Edit Libraries** dialog box.
4. Click **Add Material.**
5. Type in the piecewise linear function and use the dataset $ds1 as shown in the figure below.
6. Click **OK.**

Thus the example shows how to make a material property frequency dependent using a dataset as the first parameter to a **pwl** function.



Related Topics

Adding Datasets

Adding a Design Variable

Modifying Datasets

Using Piecewise Linear Functions in Expressions

Assigning Variables

To assign a variable to a parameter in HFSS:

- Type the variable name or mathematical expression in place of a parameter value in a **Value** text box.

  If you typed a variable name that has not been defined, the **Add Variable to *DesignName*** dialog box will appear, enabling you to define the design variable.

  If you typed a variable name that included the **$** prefix, but that has not been defined, the **Add Variable to Project** dialog box will appear, enabling you to define the project variable.

Note You can assign a variable to nearly any design parameter assigned a numeric value in HFSS. See the HFSS online help about the specific parameter you want to vary to determine if can be assigned a variable.

Related Topics

Working with Variables

Choosing a Variable to Optimize

Before a variable can be optimized, you must specify that you intend for it to be used during an optimization analysis in the **Properties** dialog box.

1. If the variable is a design variable, do the following: Click **HFSS or HFSS-IE**>**Design Properties**.

   If the variable is a project variable, do the following: Click **Project**>**Project Variables**.

   The **Properties** dialog box appears.

2. Click the tab that lists the variable you want to optimize.
3. Click the row containing the variable you want to optimize.

Note Dependent variables cannot be optimized.

4. Select the **Optimization** option above.
5. For the variable you want to optimize, select **Include**.

   The selected variable will now be available for optimization in an Optimetrics setup defined in the current design or project.

Note Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

6. Optionally, override the default minimum and maximum values that Optimetrics will use for the variable in every optimization analysis. During optimization, the optimizer will not consider variable values that lie outside of this range.

Related Topics

Setting up an Optimization Analysis

Including a Variable in a Sensitivity Analysis

Before a variable can be included in a sensitivity analysis, you must specify that you intend for it to be used during a sensitivity analysis in the **Properties** dialog box.

1. If the variable is a design variable, do the following: Click **HFSS or HFSS-IE**>**Design Properties**.

   If the variable is a project variable, do the following: Click **Project**>**Project Variables**.

   The **Properties** dialog box appears.
2. Click the tab that lists the variable you want to include in the sensitivity analysis.
3. Click the row containing the variable you want to include in the sensitivity analysis.

Note Dependent variables cannot be included in a sensitivity analysis.

4. Select the **Sensitivity** option above.
5. For the variable you want to include in the sensitivity analysis, select **Include**.

   The selected variable will now be available for sensitivity analysis in a sensitivity setup defined in the current design or project.

Note Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

6. Optionally, override the default minimum and maximum values that Optimetrics will use for the variable in every sensitivity analysis. During sensitivity analysis, Optimetrics will not consider variable values that lie outside of this range.

7. Optionally, override the default initial displacement value that Optimetrics will use for the variable in every sensitivity analysis. During sensitivity analysis, Optimetrics will not consider a variable value for the first design variation that is greater than this step size away from the starting variable value.

Related Topics

Setting up a Sensitivity Analysis

Choosing a Variable to Tune

Before a variable can be tuned, you must specify that you intend for it to be tuned in the **Properties** dialog box.

1. If the variable is a design variable, do the following: Click **HFSS or HFSS-IE**>**Design Properties**.

   If the variable is a project variable, do the following: Click **Project**>**Project Variables**.

   The **Properties** dialog box appears.
2. Click the tab that lists the variable you want to tune.
3. Click the row containing the variable you want to tune.

Note Dependent variables cannot be tuned.

4. Select the **Tuning** option above.
5. For the variable you want to tune, select **Include**.

Note Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

6. Click **OK**.

   The selected variable will now be available for tuning in the **Tune** dialog box.

Related Topics

Produce Derivatives for Selected Variables

Tuning a Variable

Example Projects: Tune a Coax Fed Patch Antenna

Including a Variable in a Statistical Analysis

Before a variable can be included in a statistical analysis, you must specify that you intend for it to be used during a statistical analysis in the **Properties** dialog box.

1. If the variable is a design variable, do the following: Click **HFSS** or **HFSS-IE**>**Design Properties**.

   If the variable is a project variable, do the following: Click **Project**>**Project Variables**.

   The **Properties** dialog box appears.
2. Click the tab that lists the variable you want to include in the statistical analysis.

3. Click the row containing the variable you want to include in the statistical analysis.

Note Dependent variables cannot be included in a statistical analysis.

4. Select the **Statistical** option above.
5. For the variable you want to include in the statistical analysis, select **Include**.

   The selected variable will now be available for statistical analysis in a statistical setup defined in the current design or project.

Note Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

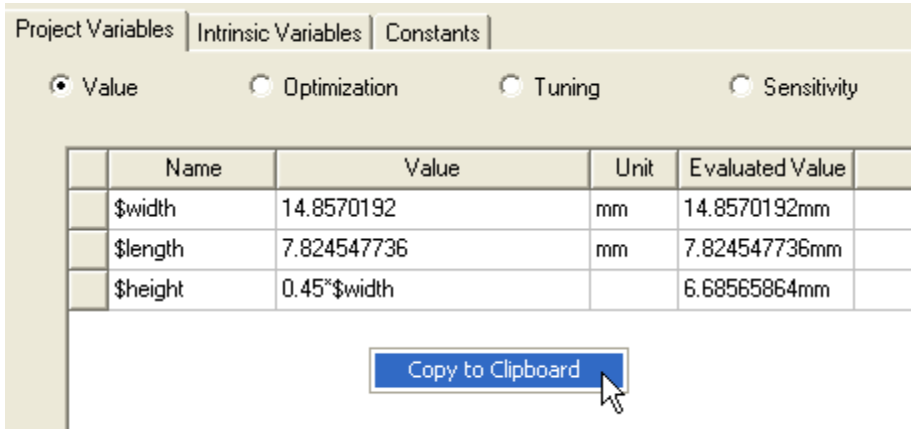
6. Optionally, override the distribution criteria that Optimetrics will use for the variable in every statistical analysis.

Related Topics

Setting up a Statistical Analysis

Exporting Variables for Documentation

By right-clicking in the Project Variables dialog, you can copy a a list of variables and their values to the clipboard. You can then paste these variables elsewhere.



The pasted variables appear in a tab separated column format. Fields that do not contain values are left blank.:

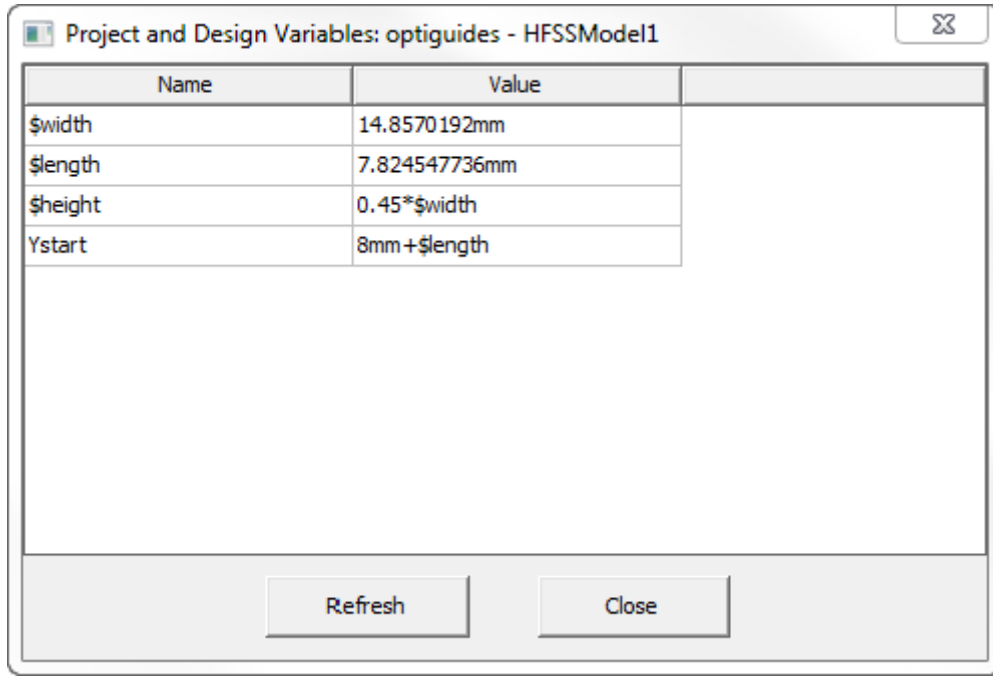
Name	Value	Unit	"Evaluated Value	"Description	Read-only	Hidden
$width	14.8570192	mm	14.8570192mm		false	false
$length	7.824547736	mm	7.824547736mm		false	false
$height	0.45*$width		6.68565864mm		false	false

Related Topics

Assigning Variables

Viewing Variables

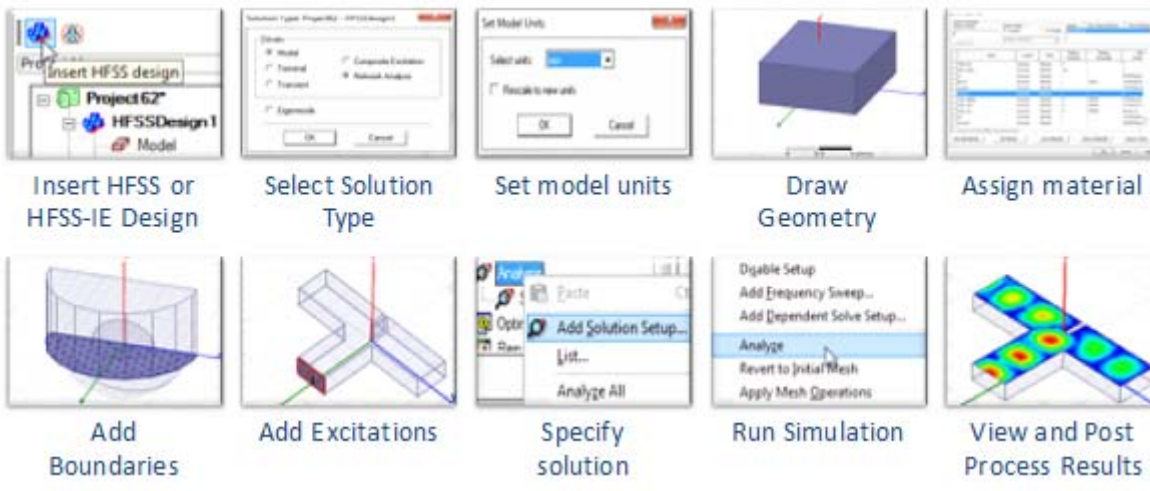
At any time within the desktop, you can select **View>Variables**, and this will pop-up a secondary window of all the defined variables. This window is persistent while you edit the properties of an object, so you can see what variables have already been defined.



4

Setting up an HFSS Design

To set up an HFSS design, follow this general procedure. After you insert a design, you do not need to perform the intermediate steps sequentially, but you must complete them before you generate a solution.



1. Insert an HFSS or HFSS-IE design into a project.
2. Select the solution type.
3. Set the model's units of measurement.
4. Draw the model geometry and assign material characteristics to objects.
5. Assign boundaries, which specify the field behavior at the edges of the problem region and object interfaces.
6. For Driven solution-type designs, assign excitations - sources of electromagnetic fields and

charges, currents, or voltages on objects or surfaces.

For Composite Excitation Solutions, you provide a complete set of excitations via the **Edit Sources** dialog before solving.

7. Specify how HFSS or HFSS-IE will compute the solution.
8. (Optional) Set up any Optimetrics you want to run.
9. Run the simulation.
10. View solution results, post-process results, view reports, and create field overlays.

Note After a period of idleness of 10 minutes, HFSS gives up its license. A renewal of activity automatically requests a license. Such idle notifications do not occur during solves.

Inserting an HFSS Design

The first step is to insert an HFSS or HFSS-IE design to the active project.

To insert an HFSS or HFSS-IE design:

- Click **Project**>**Insert HFSS Design**  or **Insert HFSS-IE Design**  .

  The new design is listed in the project tree. It is named HFSSDesign*n* or HFSS-IEDesign*n* by default, where *n*
  is the order in which the design was added to the project.

  The **3D Modeler** window appears to the right of the Project Manager. You can now create the model geometry.

Note Click the plus sign to the left of the design icon in the project tree to expand the project tree and view specific data about the model, such as its boundary assignments.

Related Topics

Setting up an HFSS or HFSS-IE Design

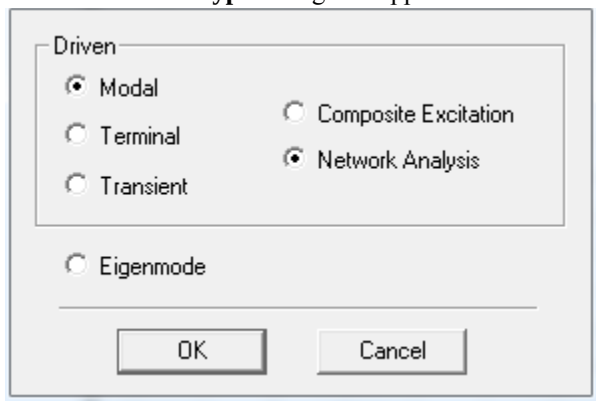
Setting the Project Tree to Expand Automatically

Selecting the Solution Type

Before you draw the model for an HFSS project, specify the design's solution type. As you set up your design, options available in the user interface will depend upon the selected solution type.

1. Click **HFSS**>**Solution Type**.

   The **Solution Type** dialog box appears.



2. Select one of the following solution types:

Driven Solutions

Modal	For calculating the mode-based S-parameters of passive, high-frequency structures such as microstrips, waveguides, and transmission lines which are "driven" by a source, and for computing incident plane wave scattering. **Network Analysis** is the default and functions as before. Composite Excitation provides a method for solving fields in a large frequency domain problem.
Terminal	For calculating the terminal-based S-parameters of passive, high-frequency structures with multi-conductor transmission line ports which are "driven" by a source. This solution type results in a terminal-based description in terms of voltages and currents. Some modal data is also available. **Network Analysis** is the default and functions as before. Composite Excitation provides a method for solving fields in a large frequency domain problem.

Transient — For calculating problems in the time domain. It employs a time-domain ("transient") solver. For **Transient** your choice of **Composite Excitation** or **Network Analysis** affects the options for the setup. If you select **Network Analysis** the setup includes an Input Signal tab for the simulation.

Typical transient applications include, but are not limited to:

- Simulations with pulsed excitations, such as ultra-wideband antennas, lightning strikes, electro-static discharge;
- field visualization employing short-duration excitations;
- time-domain reflectometry.

Not Driven

Eigenmode — For calculating the eigenmodes, or resonances, of a structure. The Eigenmode solver finds the resonant frequencies of the structure and the fields at those resonant frequencies. Eigenmode designs cannot contain design parameters that depend on frequency, for example a frequency-dependent impedance boundary condition.

Related Topics

Technical Notes: Solution Types

Technical Notes: Transient Solution Theory

Composite Excitation Solution Types

The Composite Excitation Solution type for Driven Model, Driven Terminal, and Transient provides a way to quickly solve fields in a large frequency domain problem.

For Driven Model and Driven Terminal, you provide a complete set of excitations before solving. Those excitations will be accessed via the Edit Sources dialog, and the resulting excitation vector is passed to the solver. The solver will solve only that set of excitations, and will produce:

- a single field file representing the composite excitation
- a set of surface radiated field files
- a field header file
- a vector of Active S parameters
- the port impedances
- gamma
- the port field display files

Using the **Edit Sources** dialog, you define a set of excitations as input to the solver. The right click menu on the selected excitation will bring up an **Edit Sources** dialog limited to display excitations to the selected excitation (voltage, current, terminal) or to the terminals and modes of a selected wave port. The right click menu on the excitation folder in the project tree will bring up the full **Edit Sources** dialog. Edit Sources is available from the Fields folder.

When the solution type is single excitation, the **Edit Sources** dialog will hide the total voltage and apply port post processing options - these are not supported in this mode. The buttons to load and store excitation vectors are fully supported.

In a single excitation problem:

- changes to the excitation vector are a major edit
- only design and project variables are permitted (no post process variables)
- entries in the excitation vector may be expressions
- the user can sweep those variables in Optimetrics and the reporter
- the excitation vector for each variation to be solved will be passed to the solver
- Total voltage and port post processing effects will be unavailable
- The excitation vector will be validated as part of the boundary module.

The **Edit Sources** dialog can show either the complete excitation vector or be limited to the vector for a single excitation. The latter functionality is available in the network versions of Driven Terminal, Driven Modal, and Transient.

The Load and Save functions in **Edit Sources** will be available in both network and non-network.

The solver will produce a single field file for each solved variation. This field file (and its header) will occupy the same directory structure as was used for network solutions.

The solver will produce a single vector of Active S parameters. That data will occupy a single column of the network data solution which would have been produced by a network solve. The network data will include port Zo and gamma.

Field reports and field overlays will be available as in a Network Analysis solution.

- Edit Sources is a major edit. So, it will invalidate all plots.
- Port renormalization, deembedding, and lumped port calibration are also major edits
- Animation of major intrisics (frequency) or design variables is supported.
- Field reports using post process variables (for example, to move a non-model point through the design) are supported.

Port field displays are supported as for Network Analysis.That is, port fields represent a uniform excitation at each port, NOT for the excitations specified by the user in **Edit Sources**. The reason is that we still need to solve the ports individually (even non-excited ports) before applying the user excitations.

For Composite Excitation Solution types:

- Differential pairs are not supported.
- Derivatives are not supported.
- The reporter cannot display S parameters or any data derived from them (Y, Z, …).
- Gamma, terminal (and modal) port impedance will be available. The Matrix display panel will be limited to Gamma and port impedances. There will be no touchstone export.
- The reporter can plot fields as well as Active S (and Active Y, etc.). However, the Active S is provided by the solver, rather than by post processing. Active Y, Z and VSWR are functions of Active S and port Zo.
- Radiated fields are supported.

5

HFSS Transient

HFSS Transient includes two solution types within HFSS. It employs a time-domain ("transient") solver. Typical applications include, but are not limited to:

- Simulations with pulsed excitations, such as ultra-wide band antennas, lightning strikes, electro-static discharge;
- field visualization employing short-duration excitations;
- time-domain reflectometry.

The interface of HFSS Transient is much like the Terminal Driven solution type in the frequency domain. The geometry creation and the mesh are almost identical. Boundary conditions and definitions of ports or incident waves are almost the same, except for some restrictions in cases where certain frequency-domain options do not apply in the time domain. Therefore, you can often change an existing Terminal Driven design into a Transient design and vice versa. (**Note**: before doing so, you must remove the setups and solutions.)

Notable differences for transient solution types are:

- Materials cannot have arbitrary frequency dependence in a time-domain analysis. For lossy dielectrics, a Debye model is applied to ensure that the loss remain physical across the spectrum. For lossy metals, a Padé approximation is used for the same reason.
- When lossy materials touch a port, the port solver will launch an excitation that fits the lossless case. This is inherent to most time-domain solvers. The 3D part of the simulation will take losses into account correctly.
- The Analysis Setup for the transient solver is different from that of the frequency domain. Two types of analysis setup are offered:
    a. **Transient Network Analysis** solution types - excitations are identical on all active ports and are simulated one at a time to facilitate the data collection. Saved fields can be visualized, and voltages and currents can be monitored. Your choice in setting the Transient solution type affects the options for the setup. If you select **Transient Network Analysis** the setup includes a tab for the Excitation for the simulation.

b. **Transient** Composite Excitation solution types - Different ports can have different excitations. All Active excitations are launched in one simulation. No S-parameters or TDR results may be available, but saved fields can be visualized, and voltages and currents can be monitored.

- Fields will only be saved on pre-selected surfaces, because saving all 3D fields for all times would take more disk space and I/O time than generally desired.
- If you change the size of the radiation boundary for a Transient simulation, you should expect to see changes in delay for far field results. This is because there are two origins of system for transient: one for space and one for time. For space, the original is defined the same way as frequency domain. For time, the original is located a point infinitely far away. This choice of this time origin is logical because if you place the origin at the original of a space system (in this case the center of air box) then there will be very very long time delay for every transient far field. So for the cases we have here, when air box is larger, the distance from the air box (or current source) is shorter to the time origin. Therefore we are expecting to see different time delays as radiation boundary size changes.

Related Topics

HFSS Transient Getting Started Guides

Adding a Solution Setup for Transient Solutions

Active and Passive Excitation in HFSS Transient

Differential Pairs in HFSS Transient Network

Transient Solution Theory

Selecting the Report Type in HFSS Transient

Procedure for Viewing Transient Radiated Fields

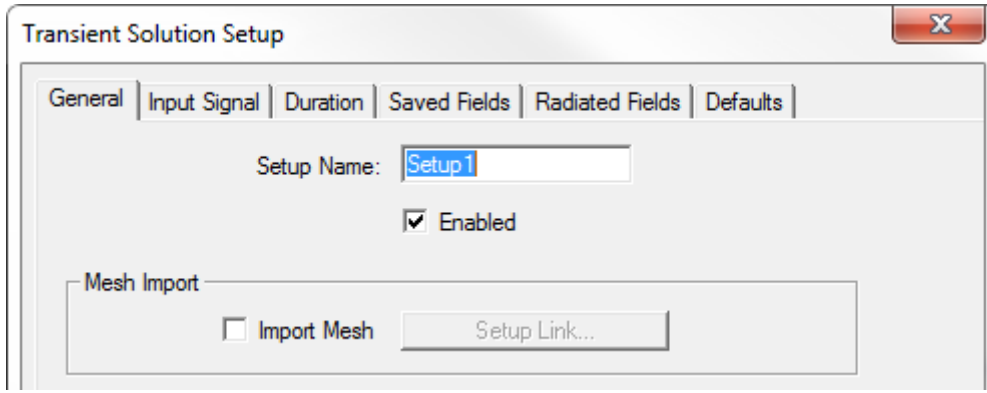
Transient GPU Acceleration

Adding a Solution Setup for Transient Solutions

When you specify the Transient solution type, you also specify the focus of this setup as either **Transient Network Analysis** or **Transient**. The **Transient Network Analysis** setup includes an **Input Signal** tab, as shown in the figure. If the model contains a radiation boundary, the setup includes a Radiated Fields tab.

To add a Solution Setup for a Transient design:

1. Click **HFSS>Analysis Setup>Add Solution Setup** to open the **Transient Setup** dialog with the **General** tab selected.



2. Optionally check Import Mesh for mesh linking.

   Otherwise, the mesh for the transient simulation is generated by a regular frequency-domain simulation. For that simulation, the software decides on the appropriate frequency at which to perform the adaptive passes. It uses mixed element orders and the iterative solver.

3. For Adaptive solutions, you can specify:
   - Maximum Number of Passes
   - For **Device characterization** solutions, you specify a Maximum Delta S.

     For **Field Visualization** solutions, you specify a Maximum Delta Energy for convergence per pass.

4. For a **Transient Network Analysis** solution, select the Input Signal tab to create a time profile.
5. Select the Duration tab to specify the simulation stop criteria.
6. Use the Saved Fields tab to select the face or object lists for which to calculate fields. By checking the boxes you can enable the Start saving fields time, and Save fields at interval. When you select an object list, you can also specify the **Maximum Number of Samples**.
7. If your design includes a radiation boundary, you can select the Radiated Fields tab to if you intend to view radiated fields.

Related Topics

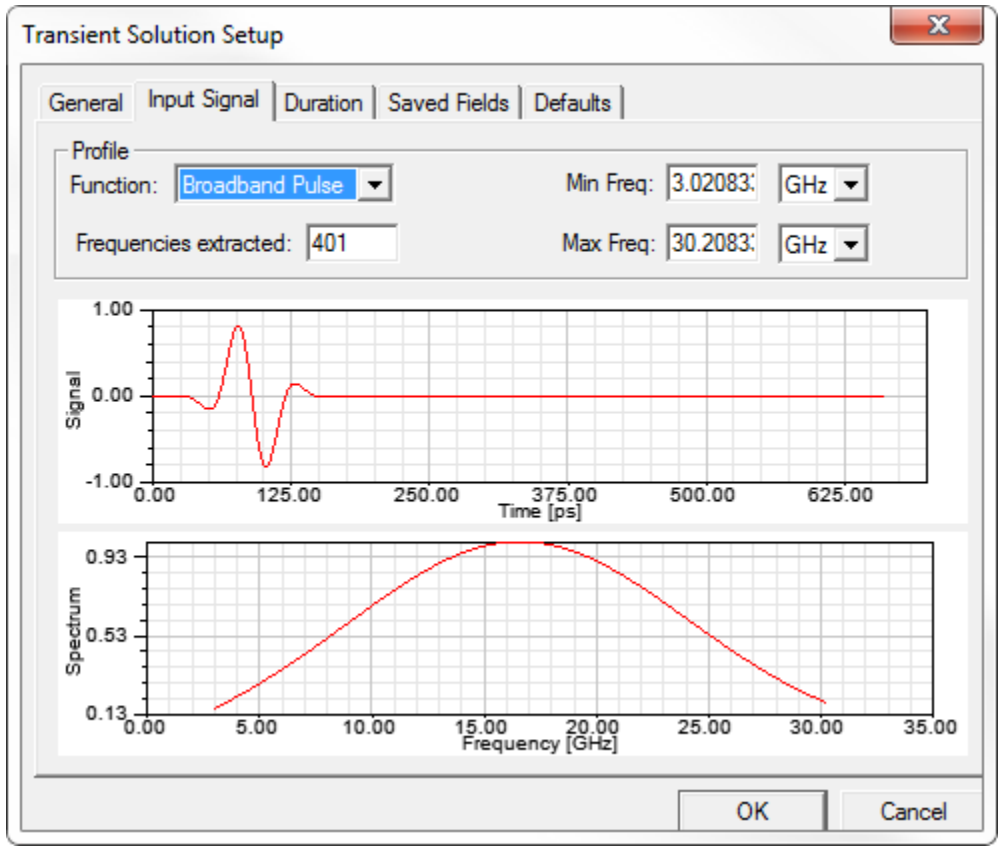
HFSS Transient Getting Started Guides

Transient Network Analysis Input Signal Tab: a Time Profile

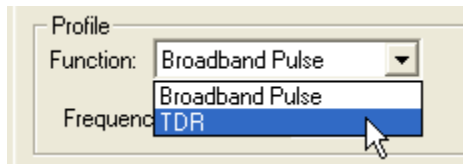
A time profile defines the pulse used to excite a **Transient Network Analysis** solution type. To create a time profile:

1. Select the **Input Signal** tab of a **Transient Network Analysis** setup.

   The **Input Signal** tab lets you select the Function set and other parameters. The profile includes two plots. The upper plot shows the excitation of interest given the currently specified parameters. The lower plot shows the energy spectrum of the upper plot



2. You specify the function to use for the Profile by selecting from the drop down menu.



For network analysis, all excitations are the same.

The wave plots are updated if you select a different function or edit the parameters.

3. Set the parameters for the selected Broadband Pulse or TDR function.

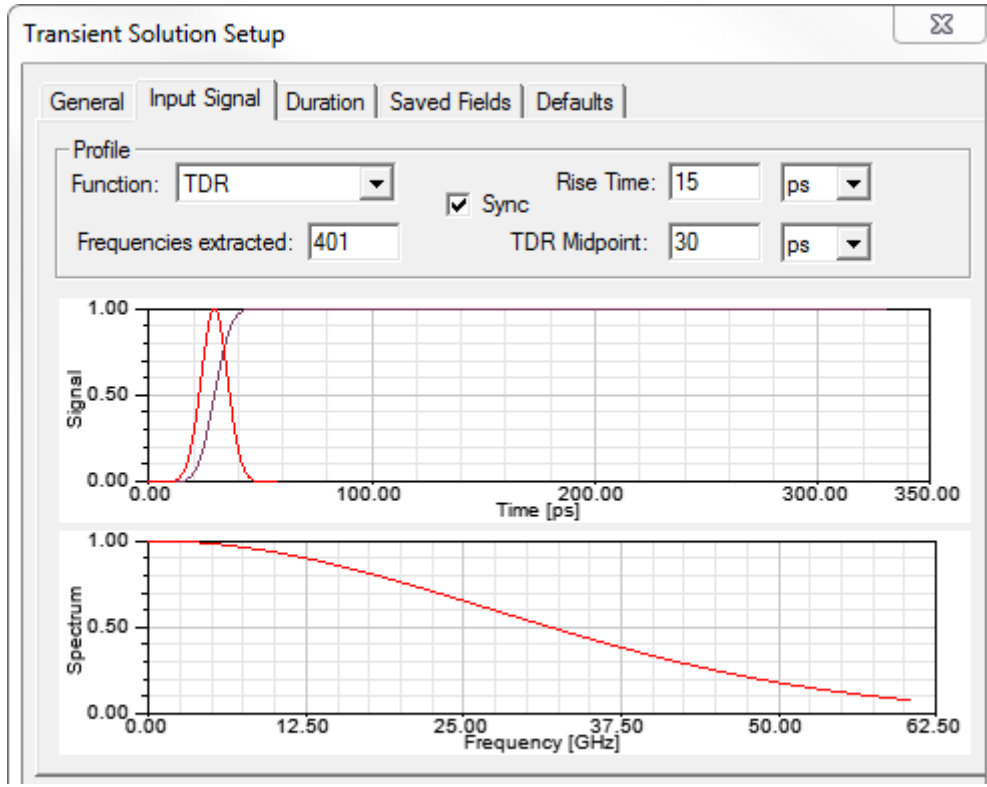
   The following table lists the available functions and the parameters that apply to the upper plot. The energy spectrum plot parameters are always Frequency and Max Frequency.

Function	Upper Plot Parameters	Description
Broadband Pulse	Min Freq. Frequencies extracted Max Freq.	A time profile that has as strong spectral content between the Min and Max frequencies. This can be used for such purposes as extracting S-parameters in that frequency range. A Min Freq. of 0 Hz. is allowed. This changes the shape of the pulse.
TDR	Rise Time Frequencies extracted TDR Midpoint Sync	This resembles a Sweep with a Min Freq. of 0 Hz. For convenience, however, this time profile is specified by the rise time rather than the frequency range. This also enables TDR output. The rise time is defined as from 10% to 90% of the peak signal value. TDR Midpoint is the time where the input TDR signal is at 50 % of the peak value. The Sync check box lets you automatically synchronize the signal midpoint and rise time such that minimum allowed midpoint is used for a given rise time or the maximum rise time for a given midpoint. If you leave the box unchecked, you can specify a different delay. If the delay is not valid, the dialog does not close, and you are prompted to provide a value.

You can utilize the TDR Midpoint feature for consistent results when comparing input signals with different rise times. One way to do this is to delay the TDR input signals so that the mid-point of the rise time always occurs at the same time point in the simulation. The major peaks and dips of TDRz for the different rise times will typically be aligned and the main differences of the results is the extra resolution obtained with a shorter rise time. For example, if you want to study the TDRz signal on a device for the rise times 30ps, 40ps, and 50 ps request a TDR Midpoint of 100ps for each signal by unchecking the sync button. Note that 100 ps is the smallest allowed midpoint for the slowest rise time - 50 ps.

The smallest midpoint allowed is 2 times the rise time to better capture the tails of the gaussian pulse used for the simulation. When sync has been checked the software automatically selects the minimum midpoint allowed for a given rise time or the maximum rise time allowed for a given midpoint.

Changing the delay invalidates solutions for the given solution setup.



4. Complete the Solution Setup for Transient Solutions.

Related Topics

Adding a Solution Setup for Transient Solutions

Transient Solution Theory

Duration Tab for Transient Solution Setup

Active and Passive Excitation in HFSS Transient

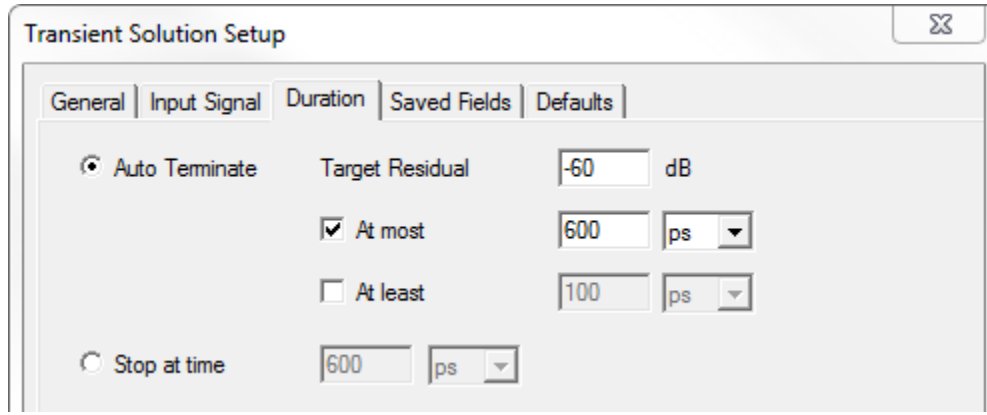
Duration Tab for Transient Solution Setup

Use the **Duration** tab of the Transient Solution setup to specify the either Auto Terminate and/or At most time and periods.

1. If you select the Auto Terminate radio button, you can also edit the Steady State Criteria value. The Transient Solver can run until the steady state criteria is reached. The value is related to the maximum field remaining in the computational domain at a given time. When the maximum field has fallen to this fraction of its all-time high, the simulation is considered complete and the analysis stops.

   The time duration of a simulation can be controlled by specifying a Target Residual in db20. The residual is defined to be the ratio of the global RMS value of the electric field at time t to the maximum RMS value in the time range zero to time t . See the technical note here.

2. Optionally, you can also specify and **At most** limit, given in either a Time or periods, or as an At least limit, given in either a Time or periods.



Related Topics

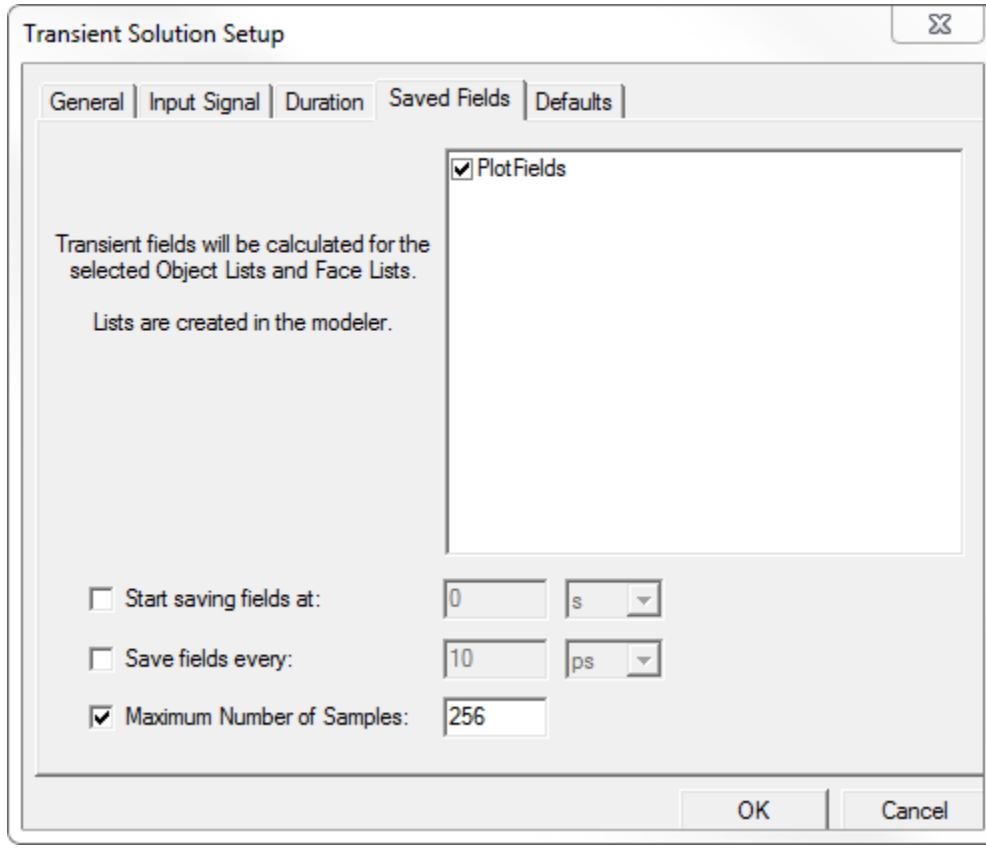
Transient Network Analysis Solution Input Signal Tab: A Time Profile

Active and Passive Excitation in HFSS Transient

Saved Fields Tab for Transient Solution Setup

The **Saved Fields** tab for a transient solution setup includes a an explanation that **Transient fields will be calculated for the selected** Object lists and Face lists.

Such lists are convenient for specifying the most relevant components.



If lists exist, you can also choose when to Start saving fields, the save interval, and specify the Maximum Number of Samples.

Transient solutions support plotting rE, a real vector versus time, in reporter. Patterns and 3D plots will also be supported.

Related Topics

Adding a Solution Setup for Transient Solutions

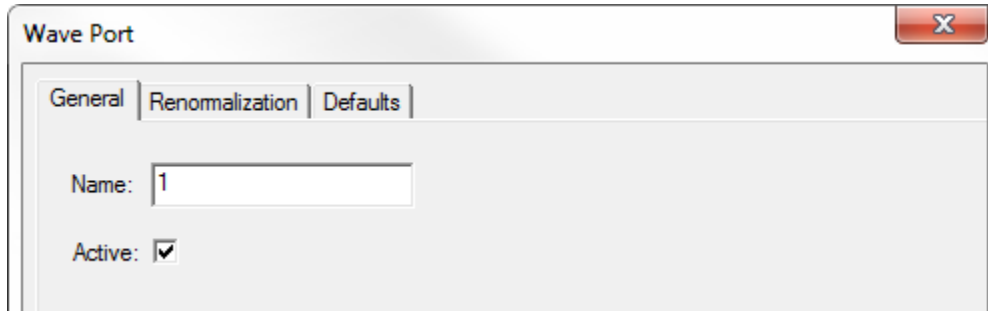
Transient Network Analysis Solution Input Signal Tab: A Time Profile

Transient Solution Theory

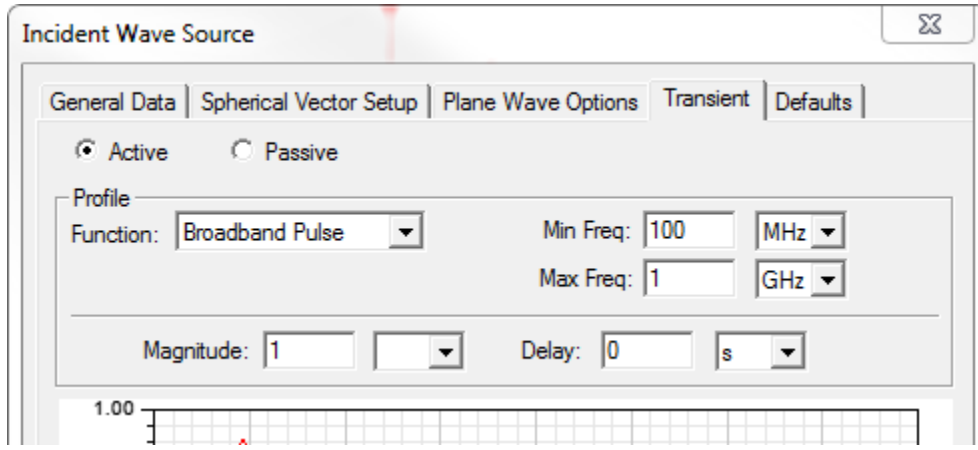
Active or Passive Excitations in HFSS Transient

In HFSS Transient, each excitation (port, plane wave, voltage, or current source) shall be "active" (with all ports active they each get an excitation one at a time, and a full S-matrix will be produced). If you want to save simulation time, you can make only one or a few ports of interest "active". The passive ports will act as terminations. You will get only a partial S-matrix.

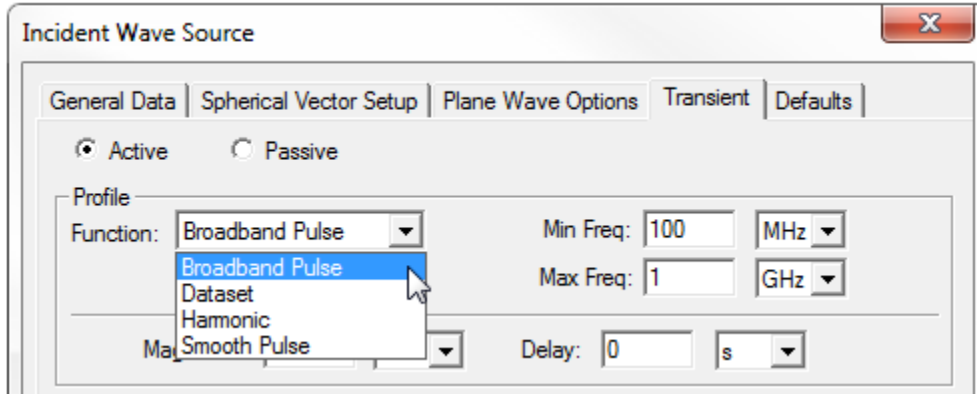
For wave ports, you specify "Active" using the checkbox on the **General** tab for the excitation.



For an incident wave source, you use the Active or Passive radio buttons on the **Transient** Tab. You can also specify Magnitude and Delay..

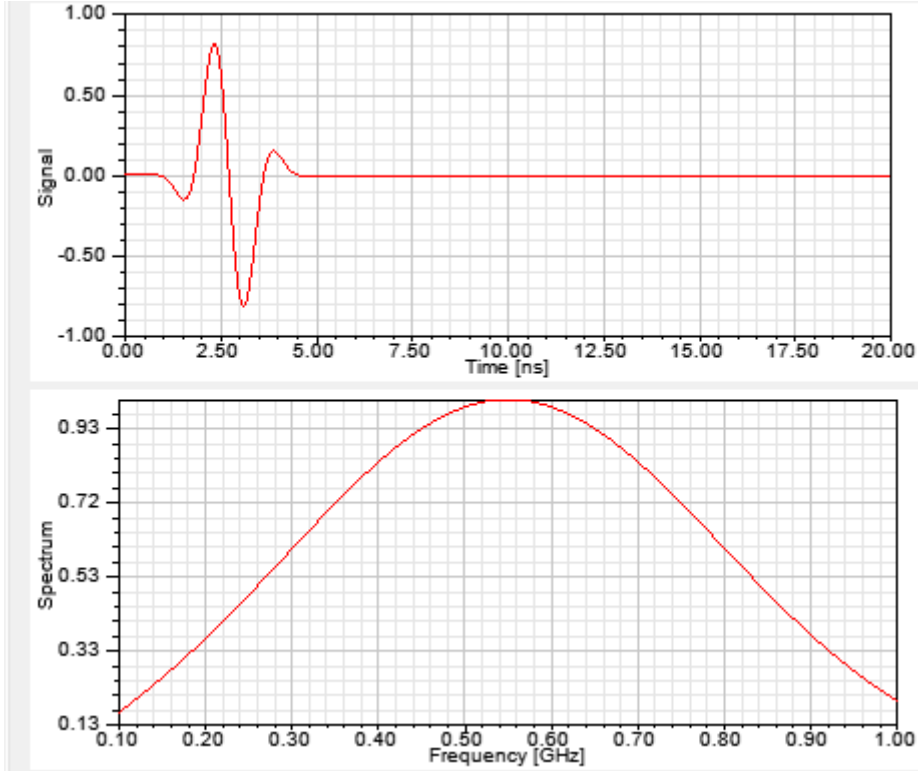


For excitations in HFSS Transient Composite Excitation solutions (and not HFSS Transient Network solutions), if you select Active, you can specify additional parameters for the excitation.



The Transient Composite Excitation Solution type has more time profiles. In this case, the time profiles are defined as part of the excitations, because excitations can have individual profiles in a Composite Excitation solution analysis. The profile includes two plots. The upper plot shows the

excitation of interest given the currently specified parameters. The lower plot shows the energy spectrum of the upper plot. The lower plot parameters include the Min and Max frequencies.



The following table describes the parameters for the upper plots.

Function	Upper Plot Parameters	Description
Broadband Pulse	Min Freq. Frequencies extracted Max Freq.	A time profile that has as strong spectral content between the Min and Max frequencies. This can be used for such purposes as extracting S-parameters in that frequency range. A Min Freq. of 0 Hz. is allowed. This changes the shape of the pulse.

Function	Upper Plot Parameters	Description
Data Set	Amplitude Max Freq.	See Adding Datasets for a description of how to create a dataset.
Gaussian	T0 Amplitude Width	The T0 parameter specified the start time of the pulse with the given Width and Amplitude.
Harmonic	Ramped Periods Amplitude Frequency	The harmonic displays as a regular sine wave.

Related Topics

Adding a Solution Setup for Transient Solutions

Transient Network Analysis Solution Input Signal Tab: A Time Profile

Setting Magnitude and Delay for Transient Excitations

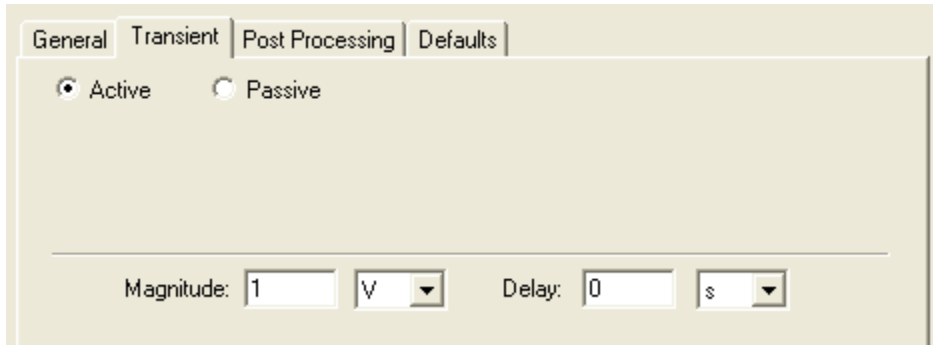
Differential Pairs in HFSS Transient Network

Technical Notes: Transient Solution Theory

Technical Notes: Excitations in the Time Domain

Setting Magnitude and Delay for Transient Composite Solution Excitations

In HFSS Transient Composite Excitation solutions, each excitation (port, plane wave, voltage, or current source) can be given a Magnitude and Delay.



For Transient:

- Scaling and delay time can be set on the input signal for each excitation. This appears only in Transient Composite Solution, not Network Analysis. Scaling is a unitless multiplier, and delay time is in appropriate time units.
- Changing scaling or delay time will invalidate solutions.

- Scaling and time delay are passed to the solver and used for the simulation.
- Legacy projects set the scaling to 1 and time delay to 0. These are also used as the defaults.

Related Topics

Adding a Solution Setup for Transient Solutions

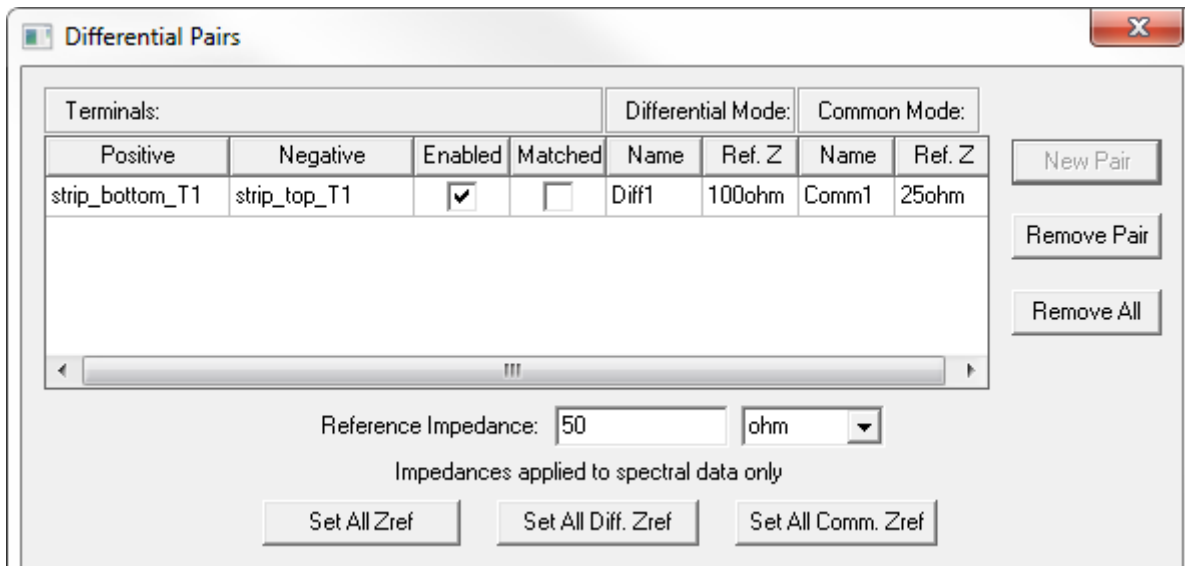
Transient Network Analysis Solution Input Signal Tab: A Time Profile

Differential Pairs in HFSS Transient Network

For post process calculation of differential pairs in Transient Network problems, you can assign differential pairs in spectral terminal problems for transient designs. For Transient Composite Excitation problems differential pairs are not supported.

In Transient Network, you can solve the single-ended matched terminal problem. HFSS Transient uses the same post processing algorithms that HFSS uses to compute the differential and common modes for spectral values. In addition, a separate algorithm computes the differential pairs values for transient data. You can switch between single-ended and differential views of both spectral and transient data. Renormalization of spectral data is supported, but renormalization does not apply to transient data.

The dialog for transient Differential pairs is slightly different than for HFSS. The dialog includes columns for Enabled and Matched, rather than Active. Passive ports are not shown in the diff pairs setup dialog.



A checkbox for "Matched" for each differential pair indicates whether to use matched or differential and common impedances. This Matched checkbox also appears in driven terminal designs and uses the same postprocessing semantics. The transient solver uses the Matched value when the Design setting "Apply when solving" is selected.

Related Topics

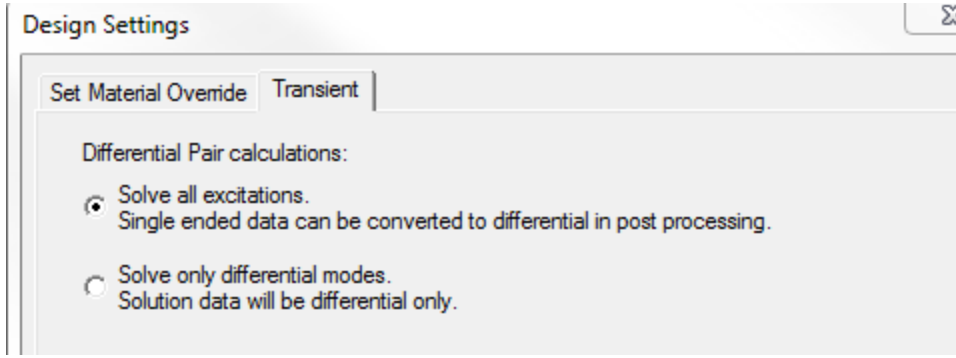
Setting Up Differential Pairs

Design Settings for HFSS Transient

Reports for Transient Network with Differential Pairs

Design Settings for HFSS Transient

The **HFSS**>Design Settings command displays a dialog with tabs for **Set Material Override** and **Transient**. The Transient tab includes options for Differential Pair calculations.



If your design contains differential pairs, you can select whether to solve all excitations so that single-ended data can be converted to differential in post processing, or to solve differential modes only.

Related Topics

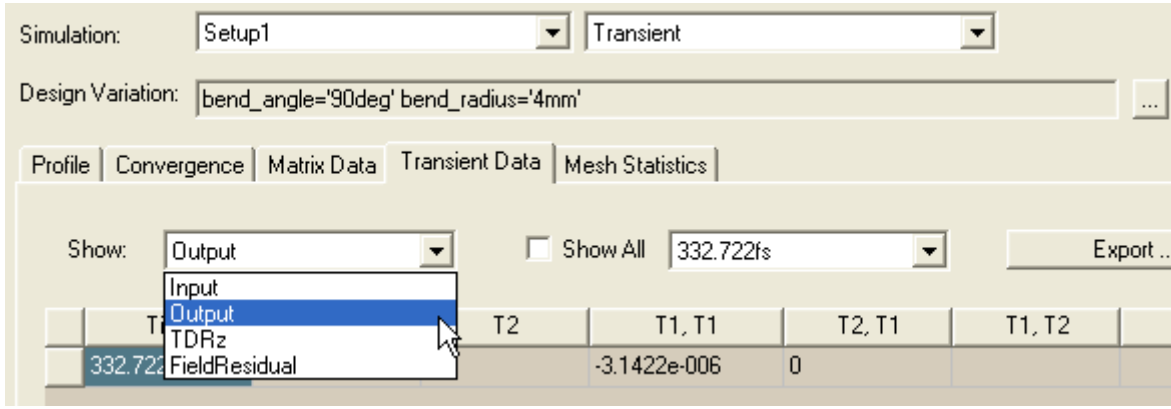
Setting Up Differential Pairs

Differential Pairs in HFSS Transient Network

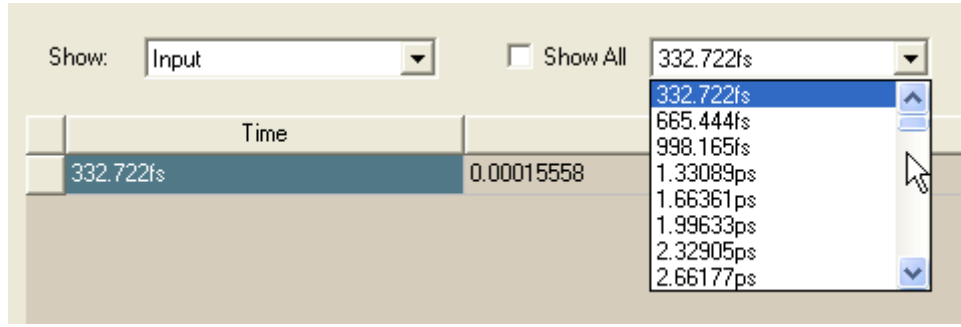
Transient Solution Data

For Transient solutions, the Solution Data dialog includes a **Transient** tab. You access is by clicking **HFSS>Results>Solution Data**.

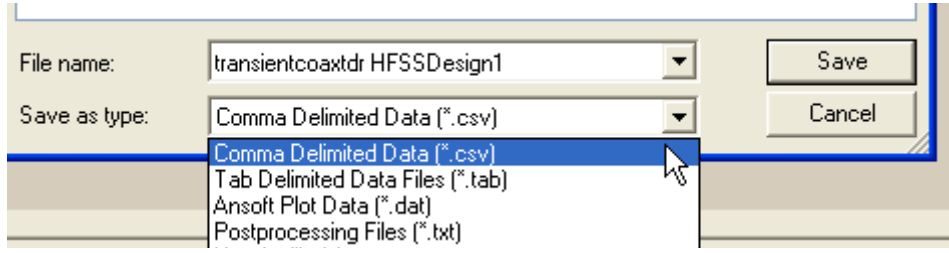
1. To enable the display of transient data on the **Transient** tab, you must first select Transient from the Simulation drop down menu.



2. Then selecting from the Show drop down menu, you can select Input, Output, TDRz or Field Residual to display.
3. You can select which time step to display from the drop down, or check Show All.



4. To export the transient data, click Export. This opens File browser winder that lets you specify a File name, location, and data format.

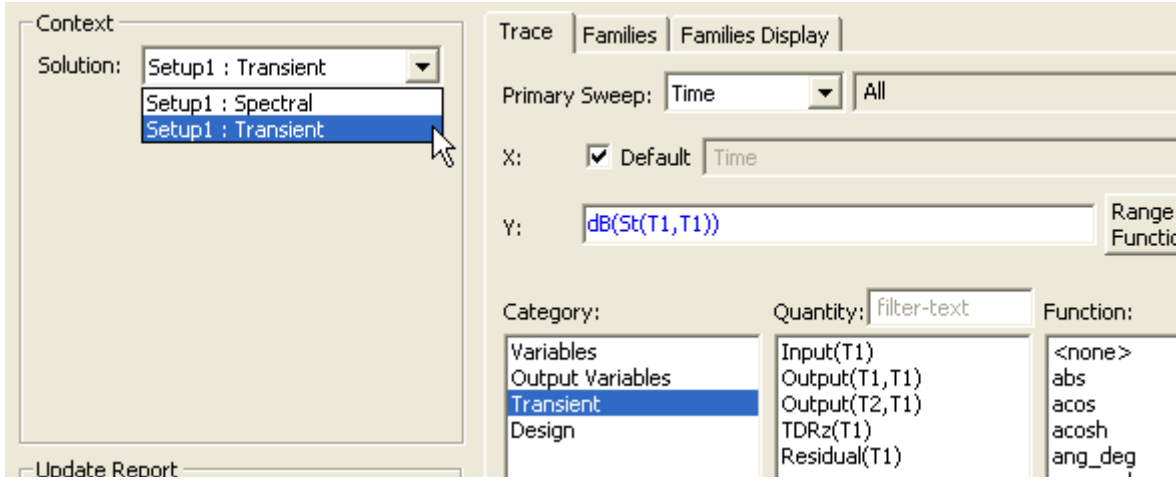


For Transient, the format selections are:

.csv	Comma Delimited Data.
.tab	Tab Delimited Data files.
.dat	ANSYS Plot Data files
.txt	Post Processing Files

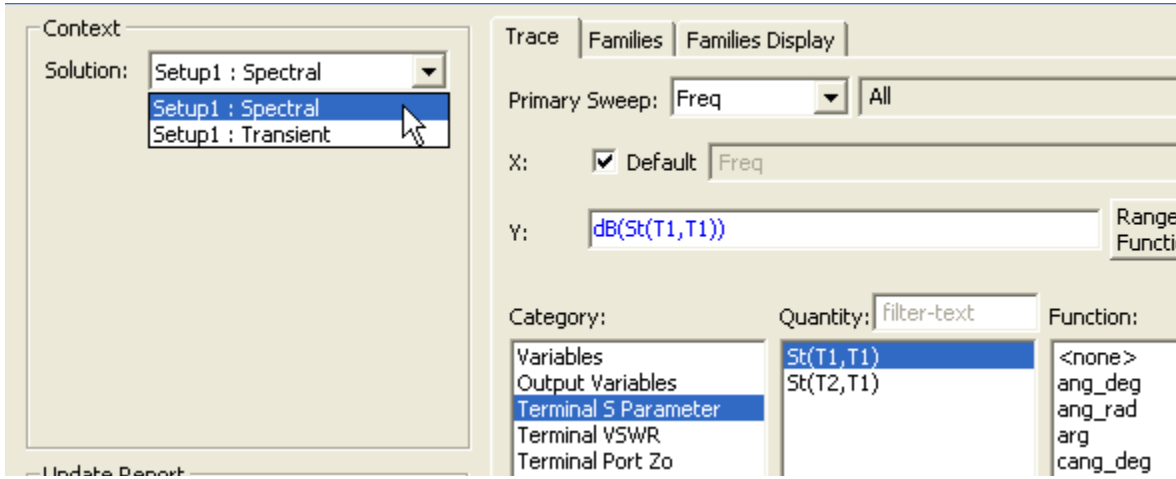
Selecting the Report Type in HFSS Transient

Creating a Report in HFSS Transient differs from standard HFSS in Solution selections. If the Solution Type is Transient, the Solution Context is Transient. If the Solution Type is Transient Network Analysis, you can also select Spectral.



With Transient selected, the Quantities include Input and Output related to the terminals in your design, as well as TDRz and Residual.

For Transient Network Solution type designs, if you select Spectral, the Category and Quantity lists offer different selections:, including Terminal S Parameters, VSWR, and Port Zo.



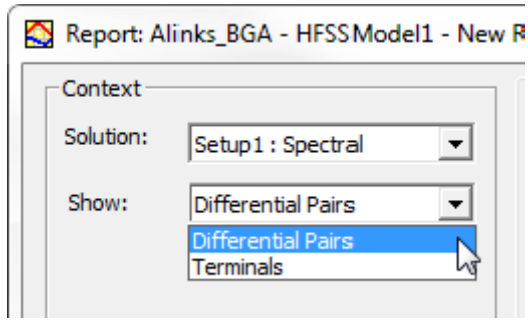
Related Topics

Creating Reports

Reports for Transient Network with Differential Pairs

Reports for Transient Network with Differential Pairs

For a Transient Network design with differential pairs defined, the Reporter interface allows selection of single-ended or differential signals just as for driven terminal. The Differential Pairs menu is shown for transient and spectral parameters, and allows you to switch between the two representations. The menu does not appear if differential pairs are not defined.



If the Design Setting "Apply when solving" is in force, it will be restricted to 'Differential Pairs' if they are present and 'Single-Ended' otherwise.

For the Transient solution (as opposed to Spectral), selecting "Differential Pairs" will limit the selected quantities to "input, output, and TDR". For the Spectral solution, all quantities will be available.

As with single-ended S parameters, post process differential pair values will not be available until the solve completes. Post process differential transient signals may be available depending on the method produced by the solver group.

The Transient Display supports the transformation between single-ended and differential transient signals under the same circumstances as the reporter.

Related Topics

Creating Reports

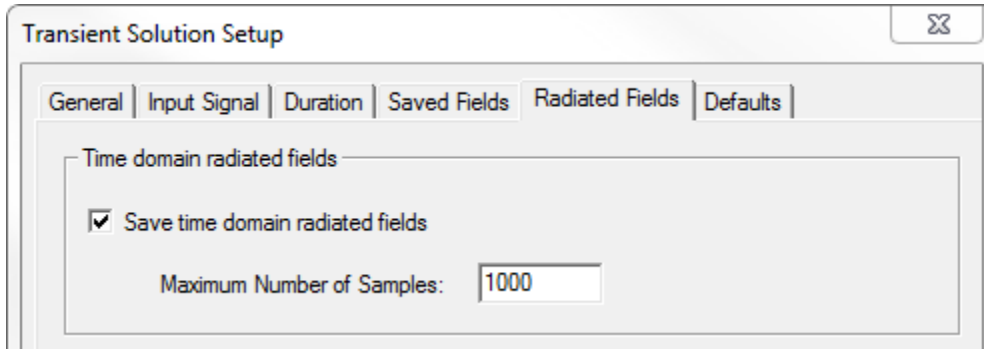
Selecting the Report Type in HFSS Transient

Differential Pairs in HFSS Transient Network

Procedure for Viewing Transient Radiated Fields

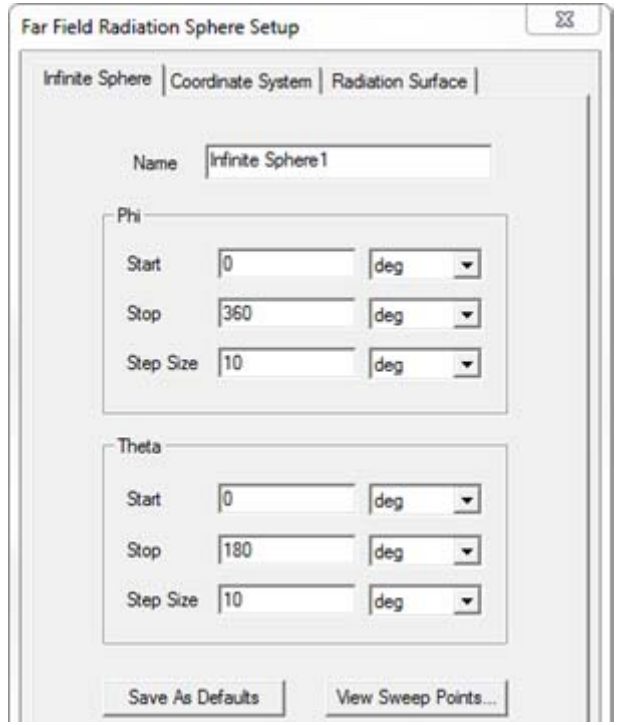
To display transient radiated fields:

1. Add a radiation boundary. Radiated field calculations will only be done for designs with radiation boundaries.
2. If a radiation boundary is present the transient Solve Setup contains the **Radiated Fields** tab with a "Save time domain radiated fields" checkbox. Select this option to make radiated fields available from a given setup. This applies to Transient with or without Network Analysis.



3. Under **HFSS>Radiation**, you can **Insert Far Field Setup> Infinite Sphere**. This menu is enabled for designs with radiation boundaries, even if no setups are saving radiated fields. The setup dialog resembles the one for frequency domain, but without the Radiation Surface tab. Use this dialog to set up the Theta and Phi sampling and, if needed, the local coordinate sys-

tem. You can create multiple Infinite Sphere setups in a single design.



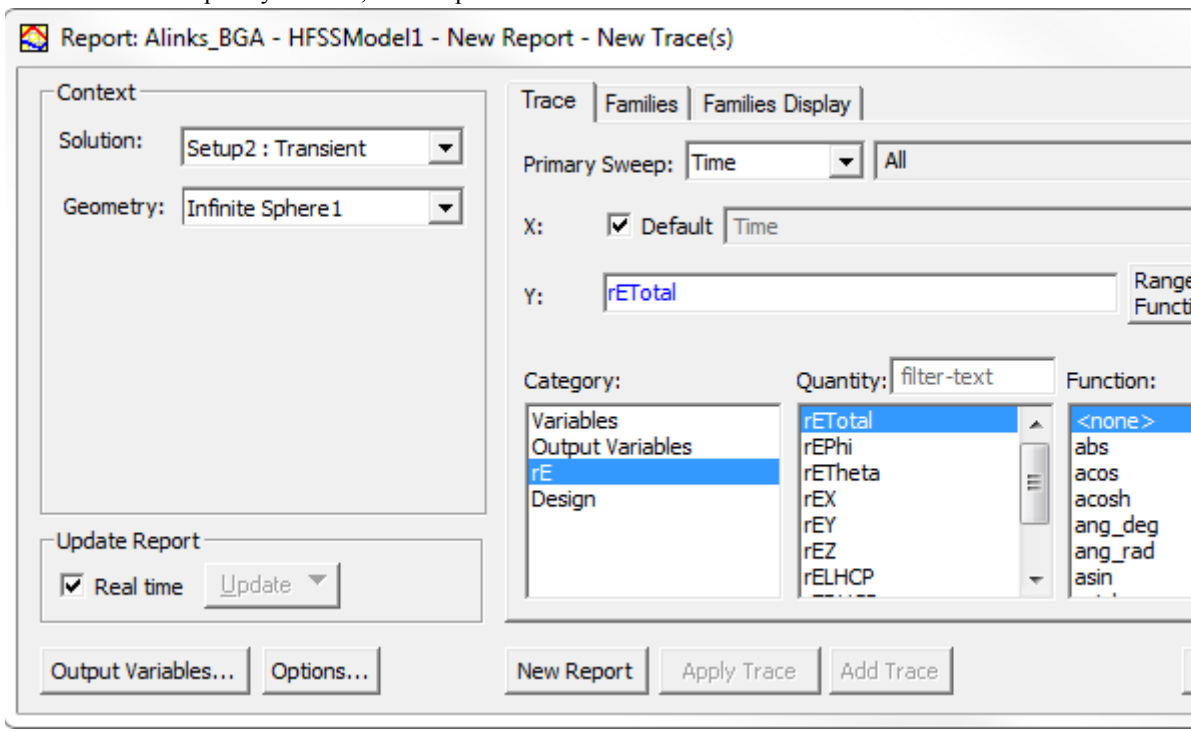
4. Once you have created a far field setup AND at least one setup has "Save radiated fields" selected, the **Results** menu will include **Create Far Fields Report**, with all submenus as in Frequency domain.

   For Rectangular Plot, Rectangular Stacked Plot, and Data Table, the default is "Time" as the primary sweep, with Theta and Phi in Families set to single values corresponding to the first sample point.

   For all other plots, the primary and secondary sweeps will be Theta and/or Phi, as in Frequency domain, and the Time is set to a single value - the start time.

5. In the **Report** dialog, the **Solution** selection includes only setups with "Save radiated fields" checked. The **Geometry** selection will include all far field Infinite Sphere setups. The **Categories** include rE, Variables, Output Variables, and Design. The rE quantities are as for fre-

quency domain, but all quantities will be real.



No matter what type of plot is generated, you can access the Time sweep and change the sampling, as with Field reports in Transient.

For 3D patterns, you can overlay the pattern on the geometry, and to animate versus time, as is done in frequency domain.

Once plots have been created, the reporter caches the base radiation field calculation. This means that subsequent plots will be generated more quickly. If you change the radiation setup, or invalidate solutions, the cache is cleared and the next plot takes longer.

For Transient Network Analysis, the radiated fields are based on the setup in Edit Sources. If you change the source excitations that forces recomputation of the radiated fields.

Output variables are supported, as for frequency domain.

Related Topics

HFSS Transient Getting Started Guides

Creating Reports

Plotting Field Overlays

Transient GPU Acceleration

Starting from HFSS 2014, the transient solver can be accelerated by Nvidia GPUs. All materials, boundary conditions, and excitations in previous versions of HFSS Transient are supported for GPU acceleration.

Hardware Requirement

GPU acceleration in HFSS Transient has been developed for Nvidia cards and is officially supported with the Tesla series. We highly recommend Nvidia Tesla cards for the best performance when using several cards on one machine to solve either multiple variations (DSO) or excitations (HPC) in parallel which is referred to in this document collectively as distributed.

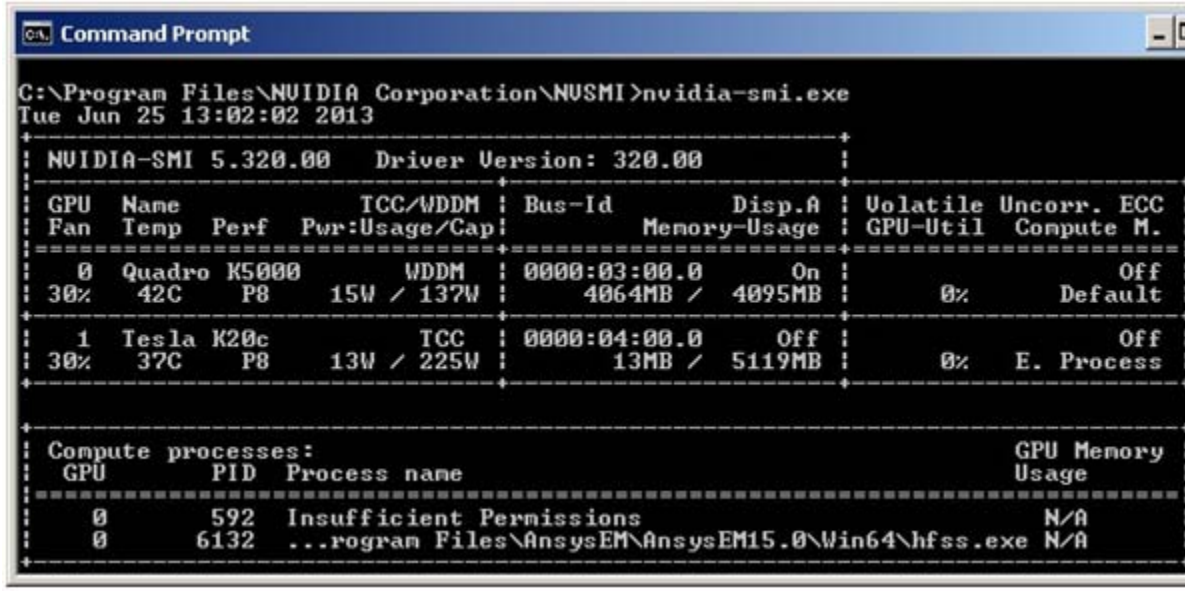
To get the best performance, the GPU used for running simulation jobs should not be attached to any display. Only GPU cards with CUDA Compute Compatibility 2.0 and above should be used. To improve the speedup of transient field visualization, you should install GPU cards on a system with PCI-E 3.0 slots. A mixture of interface cards with lower PCI-E versions may result in the data not being transferred from GPU to CPU at the highest speed.

Setup for Windows

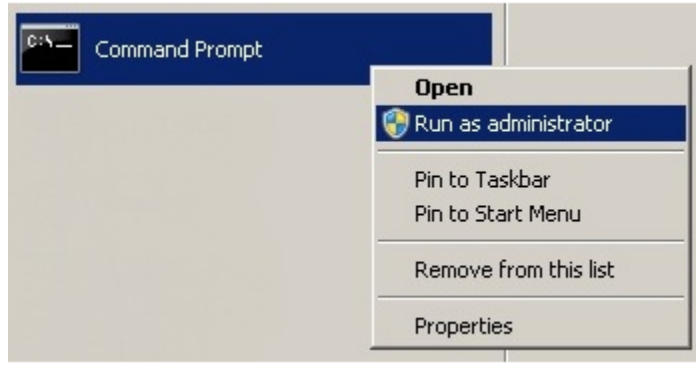
1. After you install GPU cards and Nvidia graphics drivers, you should be able to find the cards in Windows Display Manager.

2. You should run nvidia-smi.exe at C:\Program Files\NVIDIA Corporation\NVSMI to check if GPU cards are installed successfully. The executable nvidia-smi.exe should

exist after the display driver is installed..

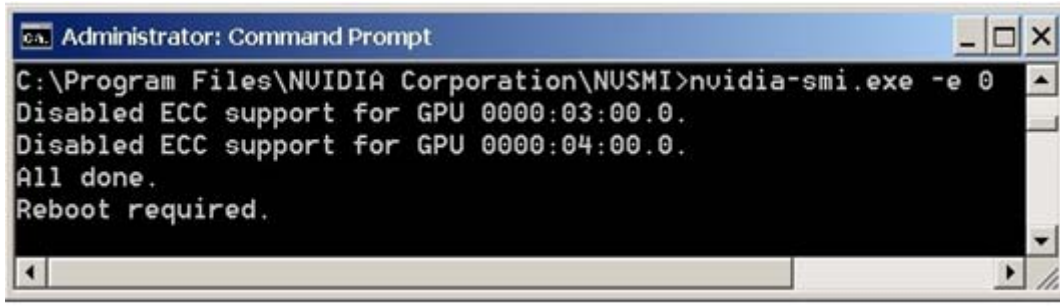


3. To further setup the configuration of GPU cards, open a command window as an administrator.

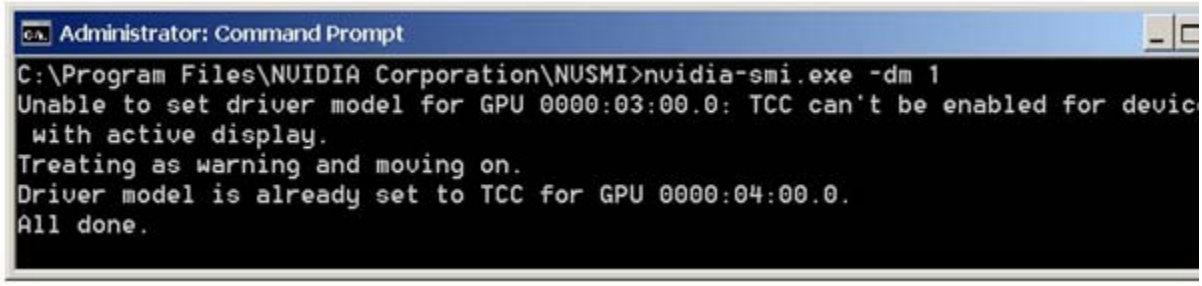


To improve the performance of GPU acceleration, it is recommended that you turn off the Error Correction Code (ECC) support by the -e 0 option of nvidia-smi. New ECC settings will

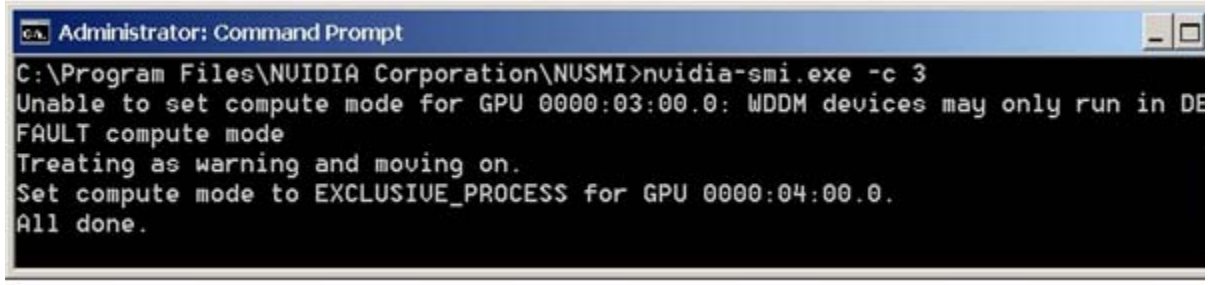
be effective only after system reboot.



4. (Optional) For remote execution of GPU accelerated jobs (e.g. through Windows Remote Desktop Connection or RSM options in HFSS), it is necessary to turn on the Tesla Compute Cluster (TCC) mode by the -dm 1 option of nvidia-smi. New TCC settings will be effective only after system reboot. This step is unnecessary if you run HFSS Transient from a local machine. Please note that only Tesla cards support TCC. You cannot run GPU accelerated jobs on remote GeForce and Quadro cards.



5. (Optional) For users who want to run multiple GPU-accelerated jobs on one machine through distributed mode, it is required to install Nvidia Tesla cards with EXCLUSIVE_PROCESS support. Using the -c 3 option of nvidia-smi, one can set GPUs in a system to be Exclusive_Process. HFSS Transient relies on this compute mode to assign each simulation job to a dedicated GPU card. Please note that GeForce and Quadro cards do not support EXCLUSIVE_PROCESS. Therefore, they should not be used for GPU-acceleration of HFSS Transient in distributed mode.



6. . (Optional) Using -q option of nvidia-smi, one can check if the compute mode is set properly.

7. (Optional) For HFSS Transient to run in GPU-distributed, it is necessary to check if the dynamically linked library nvml.dll exists in its default directory C:\Program Files\NVIDIA Corporation\NVSMI. If not, its path should be added to the Windows environment variable Path.
8. The setup for a Windows GPU system is complete through Steps 1 to 7.

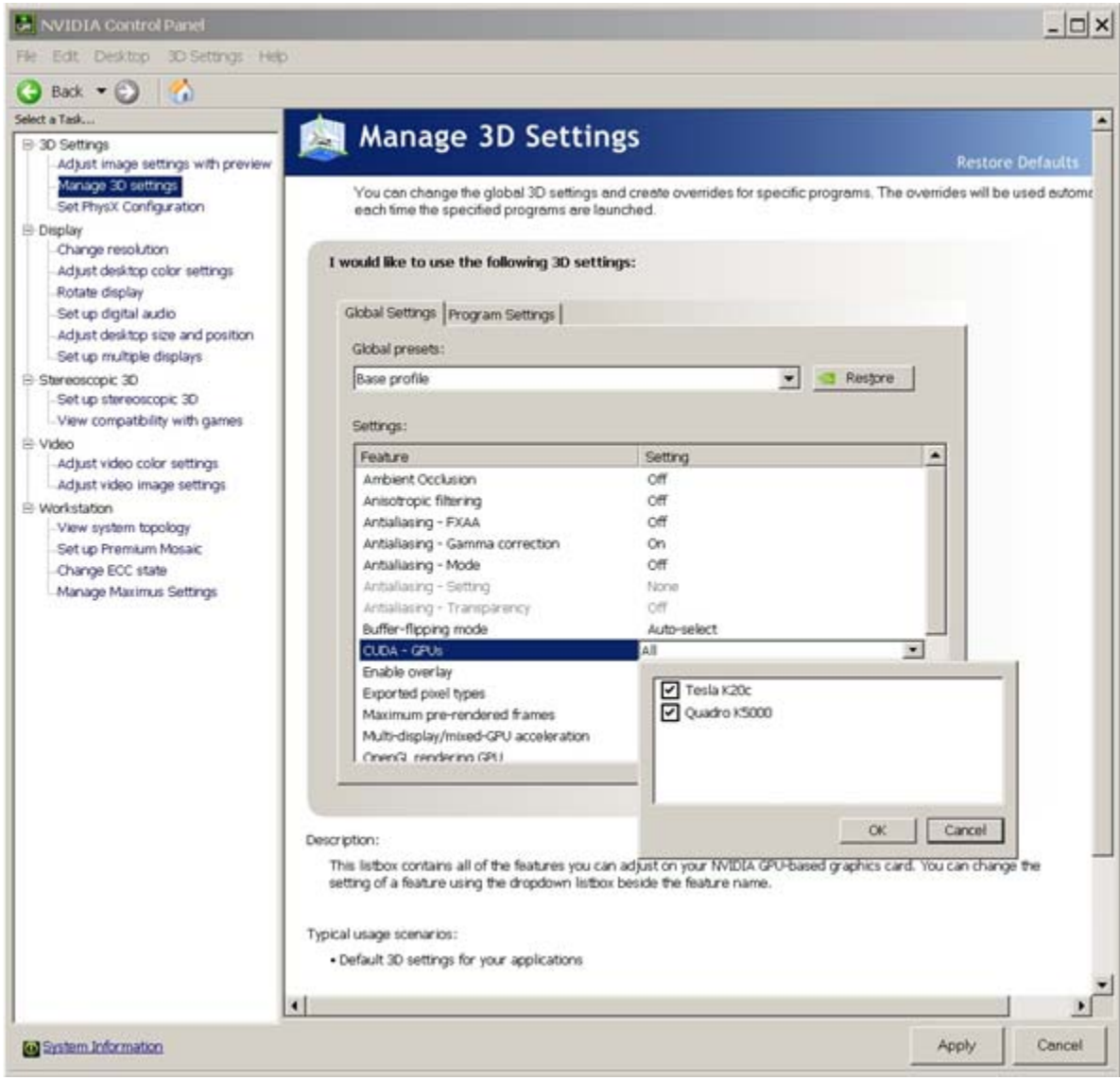
   If the computer system has both Quadro and Tesla cards, please read carefully the next section for setting up a Maximus platform.

Setup for an Nvidia Maximus Platform

When both Quadro (600, 2000, 4000, 5000, or 6000) and Tesla (C2075 and above) cards exist in a machine, the system is an Nvidia Maximus system and GPUs can be dedicated to graphics or compute tasks.

1. Once Nvidia graphics drivers are installed for a Maximus System, the icon of Nvidia Control Panel will be available on the Windows Taskbar. Click the icon to launch Nvidia Control Panel.

2. It is necessary to ensure there are GPUs visible for HFSS Transient. After the following setting to have both cards visible, HFSS Transient will be able to automatically grab Tesla K20c for GPU acceleration instead of Quadro K5000, because the latter is attached to a display and other processes in the system may use it for graphics acceleration. If only Tesla K20c is visible (Quadro K5000 unchecked), it will be used for GPU acceleration. If only Quadro K5000 is visible (Tesla K20c unchecked), it will still be used for GPU acceleration even with a display attached.



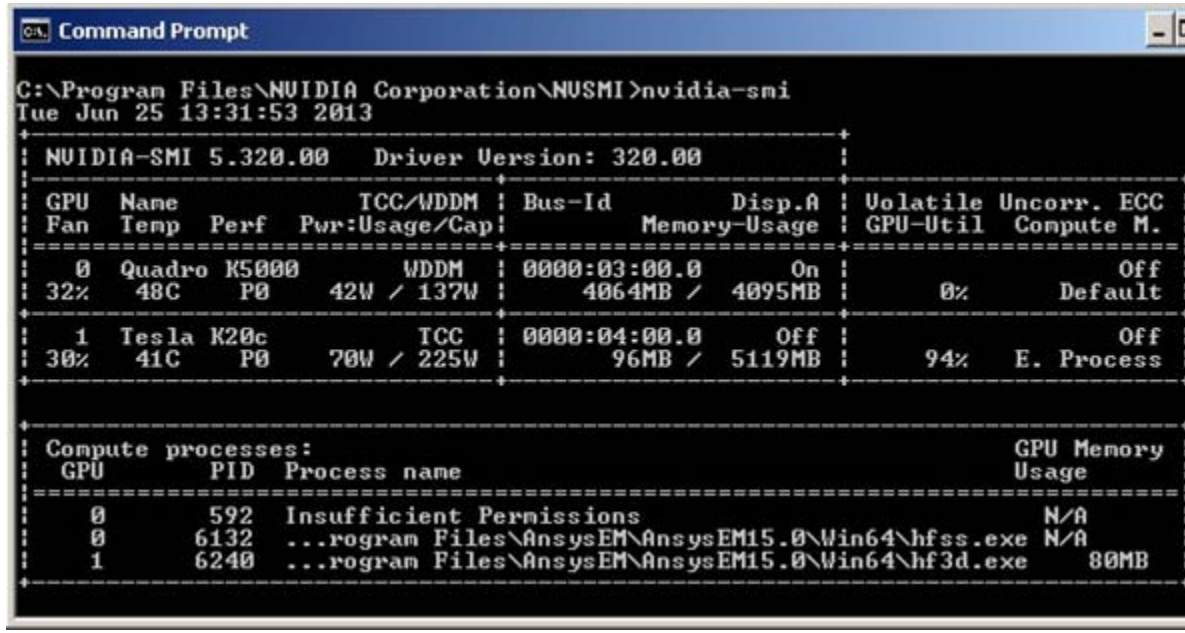
3. Furthermore, you must check to see if an environment variable CUDA_VISIBLE_DEVICES exists in Windows. To be consistent with the above Maximus settings, the variable should be set to make both GPUs visible.

   CUDA_VISIBLE_DEVICES=0,1

   If the variable does not exist, all CUDA devices are visible in a system by default. HFSS Transient will only run on visible GPUs not attached to displays. However, if only one GPU exists

in a system and it is used for both compute and display, HFSS Transient will still grab it for acceleration.

4. The Maximus setup is complete through Steps 1 to 3. As an example, the following figure illustrates one hf3d process running on Tesla K20c and one hfss process using Quadro K5000 for graphics acceleration.



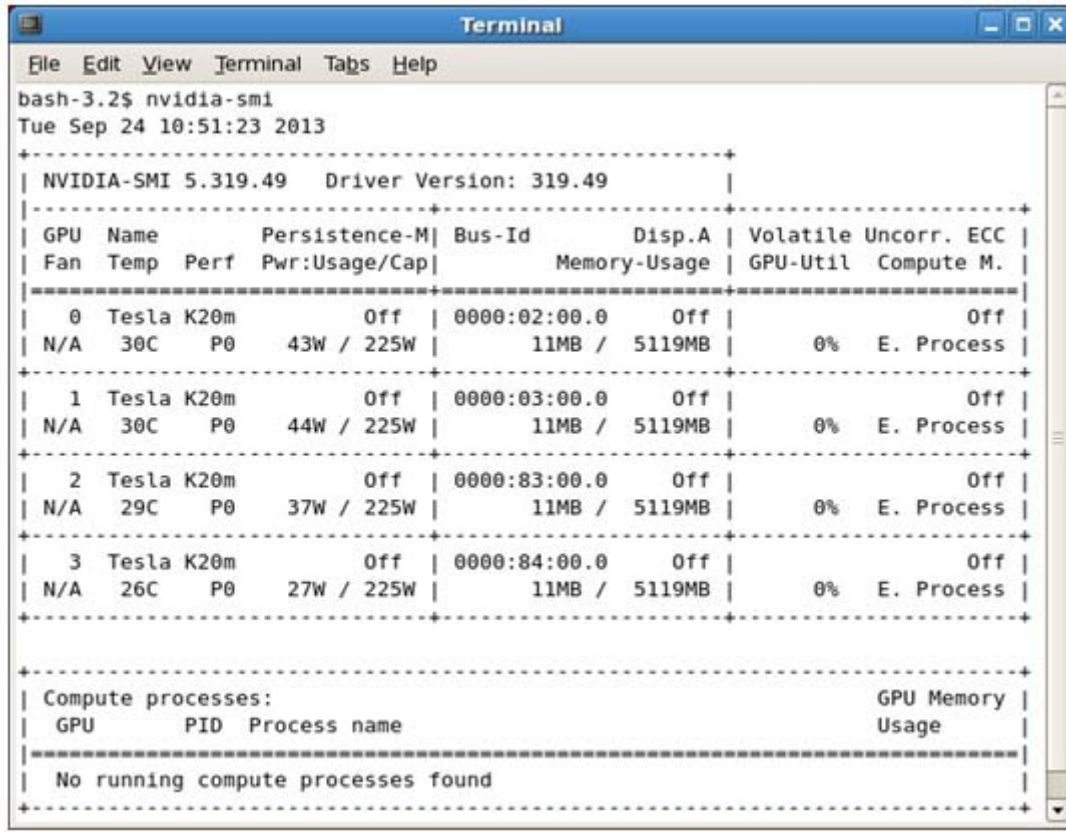
Setup for Linux

1. After you install Nvidia GPU cards and graphics drivers, you should be able to find the cards by the command.

   ```
   /sbin/lspci | grep -i nvidia
   ```

   You can also use the following command to check if GPU cards can be recognized by the system.

   ```
   /usr/bin/nvidia-smi
   ```

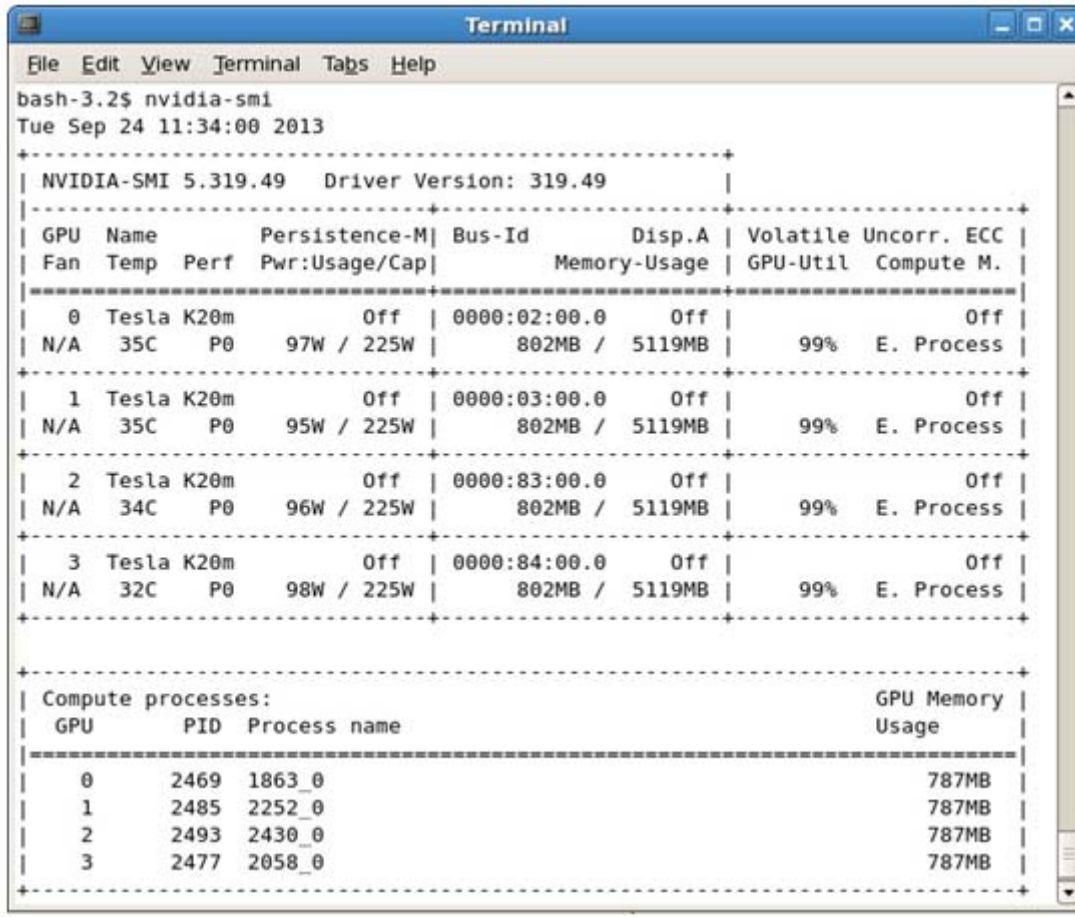


2. (Optional) The setting of GPUs to disable ECC (for performance), enable TCC (for remote execution), and enable Exclusive_Process (for GPU-distributed) are similar to Windows. You need the administrative right to make such changes.

   ```
   sudo nvidia-smi  -e 0
   sudo nvidia-smi -dm 1
   sudo nvidia-smi -c 3
   ```

3. (Optional) On Linux Maximus platforms, the environment variable CUDA_VISIBLE_DEVICES can be set in the shell file ~/.bash_profile to toggle the visibility of GPU cards as CUDA devices. If the variable does not exist, all CUDA devices are visible in a system by default.
4. (Optional) In order to let HFSS Transient access the dynamically linked library libnvidia-ml.so in GPU-distributed, one must check if the library exists in its default directory /usr/lib64. If not, its path should be added to the Linux environment variable PATH through a shell startup file.

5. The setup for a Linux GPU system is complete through Steps 1 to 4. As an example, the following figure illustrates four distributed hf3d processes running on four Tesla K20m cards.



License Options

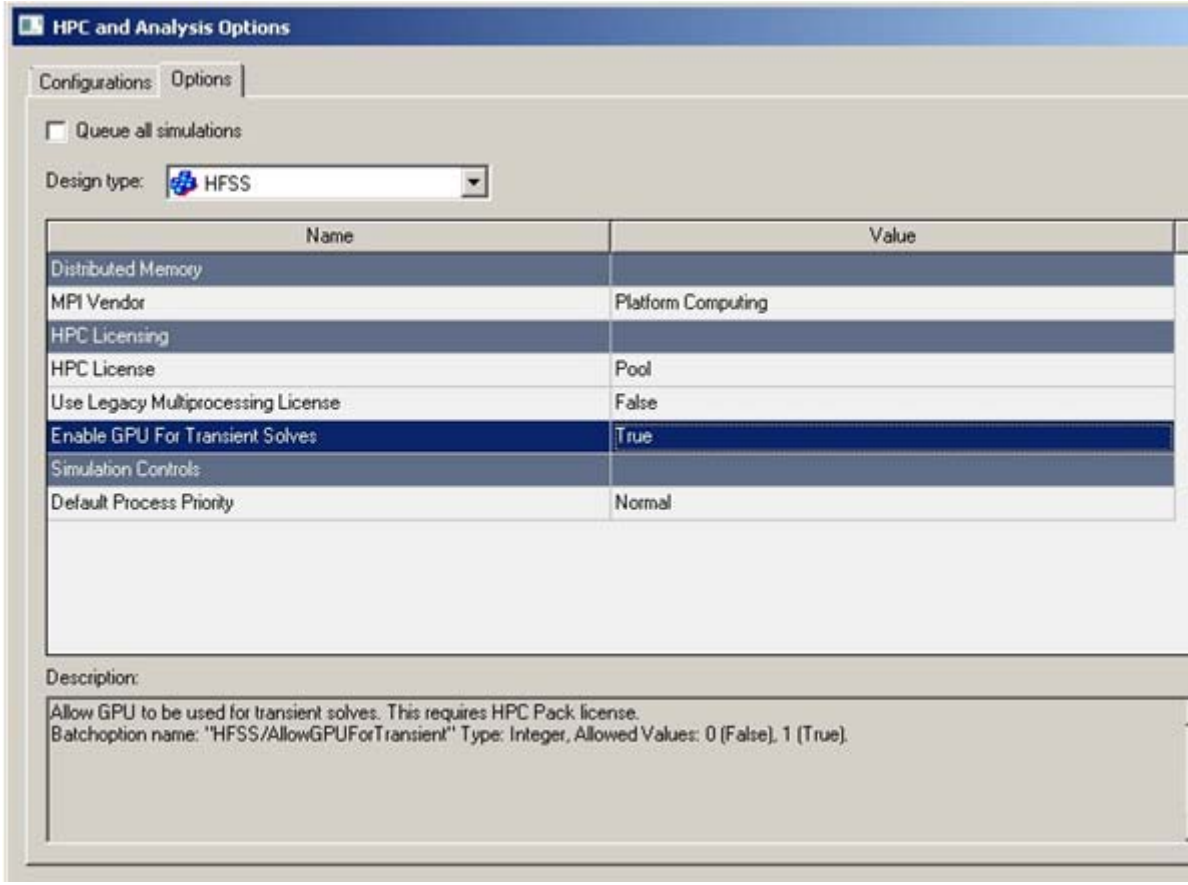
Users with HPC packs can use GPU acceleration for HFSS Transient. The maximum number of GPUs to be used on a standalone machine or for all machines in a cluster is limited by the number of HPC packs.

- 1 HPC pack = 1 GPU
- 2 HPC packs = 4 GPUs
- 3 HPC packs = 16 GPUs
- 4 HPC packs = 64 GPUs

If the number of simulation jobs exceeds the number of GPUs in a system, the excessive jobs will fall back to CPUs and will be accelerated by up to 8 CPU cores for each job.

Enable/Disable GPU Acceleration from the HFSS User Interface

To turn on or off GPU acceleration, click **Tools>Options>HPC and Analysis Options** and select the **Options** tab. Click on the value of Enable GPU For Transient Solves and toggle it to either True or False.



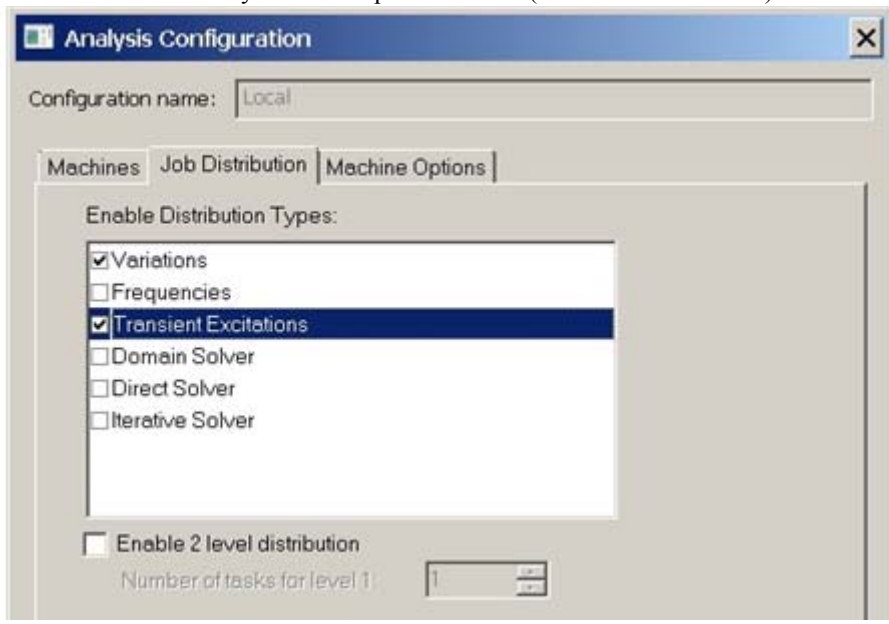
Enabling GPU-Distributed

There are two steps for setting up GPU-distributed.

1. Go to **Tools>Options>HPC and Analysis Options** to edit Analysis Configurations.

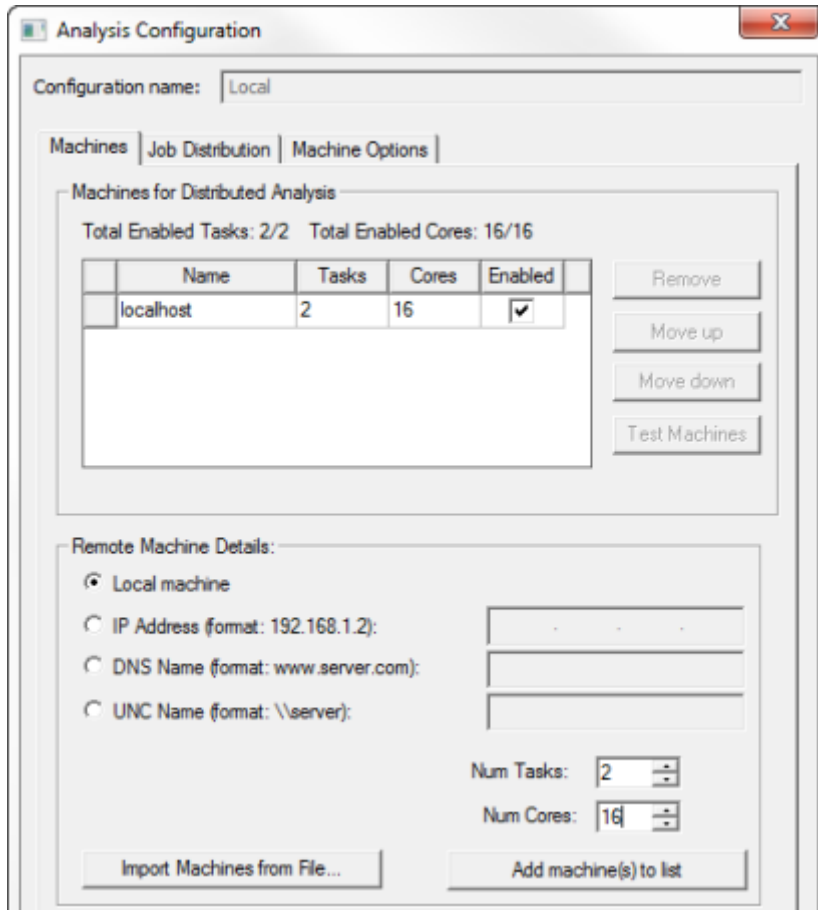
   GPU-DSO in HFSS Transient can be used either for parametric sweep (Variations) or Tran-

sient Network analysis of multiport networks (Transient Excitations).



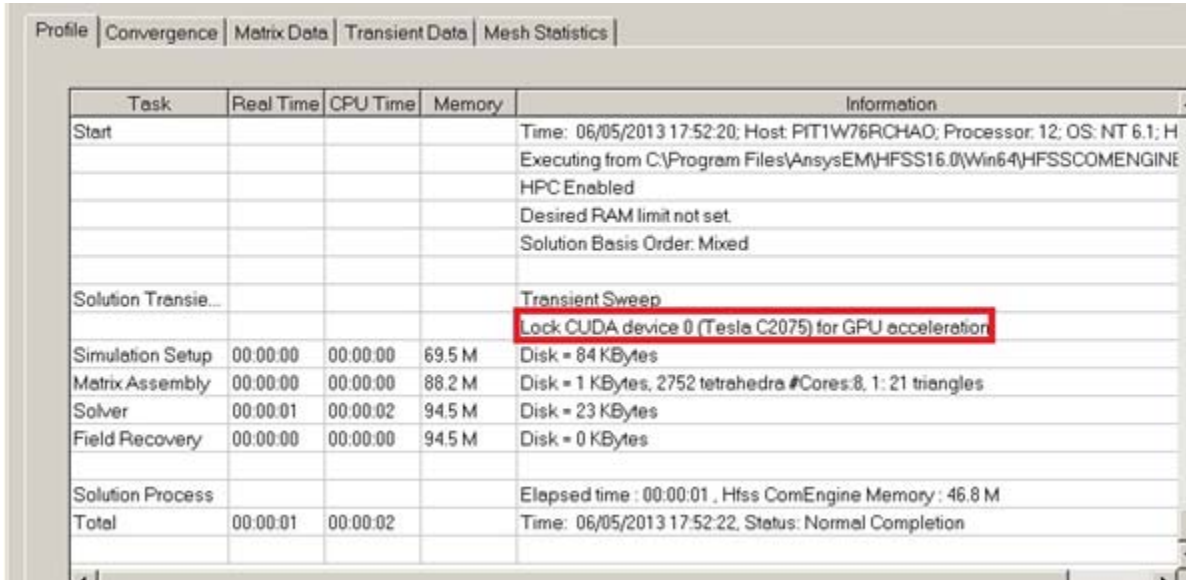
2. Set up the Machines for GPU-Distributed.

   In the following figure two simulation jobs are allowed to run on the local host, or up to two GPUs can be used for acceleration depending on the availability of licenses and GPU cards.

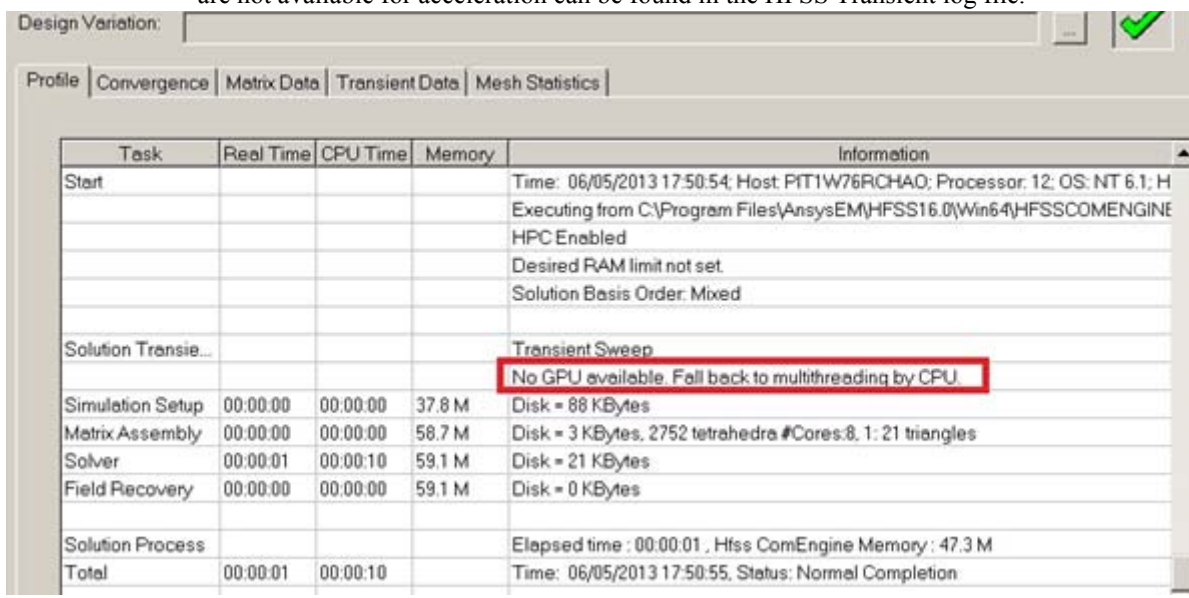


Whether GPU is used for acceleration can be checked by viewing the **Solutions** dialog, **Profile** tab. If a GPU is successfully locked for the use by an hf3d process, the profile will show the

GPU's CUDA device ID and its name.



Otherwise, the profile will indicate the fallback to CPUs. More information about why GPUs are not available for acceleration can be found in the HFSS Transient log file.



Enable/Disable GPU Acceleration from the Command Line

GPU acceleration of HFSS Transient can be toggled by the -batchoptions command line argument:

`EnableGPU=[0/1].`

For example, the following command turns on GPU acceleration.

```
hfss.exe -batchsolve -batchoptions "'EnableGPU'=1"
projectname.hfss
```

6

HFSS-IE

HFSS-IE is a fullwave Integral Equation solver that calculates the "currents" on the surfaces of the objects in the model - both finite conducting and lossy dielectric objects are allowed. HFSS-IE is designed for large open problems. Application areas include:

- Radar cross Section (RCS)
- Antenna placement (for example, antenna on a vehicle)
- Stand alone antennas
- Coupling. EMI/EMC

HFSS-IE features:

- Works from within the standard HFSS desktop, sharing the GUI and the same 3D modeler and reporting features.
- Naturally open - no air volume or ABC needed
- Support for infinite ground plane
- Supports ground plane apertures
- Supports lumped gap and incident wave excitations.
- Supports discrete and/or interpolating frequency sweeps
- Supports mesh link which means you can use the current mesh of different design or project.
- Supports curvilinear mesh elements without restriction.
- Near and far field calculations
- For larger models, HFSS-IE uses automated advanced matrix based compression techniques.
- Data link to and from HFSS and HFSS-IE available, which means that you divide appropriate models based on each solver's advantages.
- Physical Optics solver option for plane incident wave and far field incident wave models.
- Use Distributed memory model option, whose features are discussed in Distributed Memory

Solutions with HFSS-IE.

How HFSS-IE is different than HFSS

HFSS uses the finite element method (FEM) to solve for the electromagnetic fields in the solution region. It meshes over the entire solution volume and solves for the electric field throughout that volume.

HFSS-IE uses an integral equation (sometimes called method of moments = MoM) and solves for the currents on surfaces of objects. It creates a triangular surface mesh on all objects - it solves for the currents or equivalent currents on conducting and dielectric objects. The IE technique is by default an "open model" technique so no ABCs are needed. It can handle closed spaces, but that is typically not where it will be used. In addition HFSS-IE includes a true infinite ground plane.

Related Topics

HFSS-IE Options

Inserting an HFSS or HFSS-IE Design

Setting up an HFSS or HFSS-IE Design

Assigning Excitations

Assigning HFSS-IE Boundaries

Adding a Solution Setup to an HFSS-IE Design

HFSS-IE Getting Started Guides

Distributed Memory Solutions within HFSS-IE

Technical Notes: Integral Equation Method Used in HFSS-IE

Inserting an HFSS-IE Design

The first step is to insert an HFSS-IE design to the active project.

To insert an HFSS-IE design:

- Click **Insert HFSS-IE Design**  .

  The new design is listed in the project tree. It is named HFSS-IEDesign*n* by default, where *n* is the order in which the design was added to the project.

  The **3D Modeler** window appears to the right of the Project Manager. You can now create the model geometry.

Note Click the plus sign to the left of the design icon in the project tree to expand the project tree and view specific data about the model, such as its boundary assignments.

Related Topics

Setting up an HFSS or HFSS-IE Design

Setting the Project Tree to Expand Automatically

Assigning HFSS-IE Boundaries

HFSS-IE designs can include the boundaries listed in the table. HFSS-IE designs assume that the model is surrounded by some background material, so you do not have to draw an enclosing region with an absorbing boundary condition, as in HFSS. HFSS-IE assumes that the background material is vacuum.

Perfect E	Represents a perfectly conducting surface. This resembles the HFSS Perfect E boundary, but does offer selecting an infinite ground plane.
Finite Conductivity	Represents an imperfect conductor.
Infinite Ground Plane	Represents the effects of an infinite ground plane
Aperture	Represents holes in the design.
Impedance	Represents a resistive surface.
Lumped RLC	Represents any combination of lumped resistor, inductor, and/or capacitor in parallel on a surface.
Layered Impedance	Represents a structure with multiple layers as one impedance surface.

Related Topics

Assigning Boundaries

Zoom to Selected Boundary

Setting Default Boundary Base Names

Modifying Boundaries

Deleting Boundaries

Reassigning Boundaries

Reprioritizing Boundaries

Duplicating Boundaries and Excitations with Geometry

Showing and Hiding Boundaries and Excitations

Reviewing Boundaries and Excitations in the Solver View

Setting Default Values for Boundaries and Excitations

Assigning Excitations in HFSS-IE

Excitations in HFSS-IE are used to specify the sources of electromagnetic fields and charges, currents, or voltages on objects or surfaces in the design. You can assign the following types of excitation in an HFSS-IE design:

Lumped Port	Represents an internal surface through which a signal enters or exits the geometry.
Terminal	Represents a terminal. You can assign terminals manually or automatically.
Plane Incident Wave	Represents a wave that propagates in one direction and is uniform in the directions perpendicular to its direction of propagation. HFSS-IE supports regular propagating wave, evanescent, and elliptically polarized plane waves.
Far Field Wave	A Far field wave is sufficiently far (that is, usually more than a wave length distance) from an antenna to approximate as a plane wave. Far field waves are mostly homogeneous.
Near Field Wave	A Near Field wave is close enough to the antenna source for near field effects to occur, typically within a wave length. Near field waves tend to be evanescent, that is, non-homogeneous.

After assigning an excitation, you can modify it in some of the following ways, if applicable to the excitation type:

- Change its properties.
- Delete it.
- Reassign it to another surface.
- Hide it from view.
- Modify the impedance multiplier.

Adding a Solution Setup to an HFSS-IE Design

To add a new solution setup to a design:

1. Select a design in the project tree.
2. Click **HFSS-IE**>**Analysis Setup**>**Add Solution Setup**  .
   - Alternatively, right click **Analysis** in the project tree, and then click **Add Solution Setup** on the shortcut menu.
   - If you have an existing setup, you can Copy and Paste it, and then edit parameters.

   The **Solution Setup** dialog box appears. It is divided among the following tabs:

General	Includes general solution settings.
Options	Includes settings for lambda refinement, adaptive analysis and solution options.
Advanced	Includes settings for Initial Mesh Options. You can choose to use a mesh from a different design or project that has an identical geometry.
Expression Cache	Includes a list of expressions and output variables that you can use for convergence for adaptive analysis.
Defaults	Enables you to save the current settings as the defaults for future solution setups or revert the current settings to HFSS's standard settings.

3. Click the **General** tab.
4. Enter a **Setup Name** or accept the default.

   The **Enabled** checkbox on **General** tab permits to you to disable a setup so that it does not run when you select Analyze All.
5. Enter the **Solution Frequency** and select the frequency units from the pull down list.
6. If you are performing an adaptive analysis, enter **2** or more passes in the **Maximum Number of Passes** box.

   For driven problems HFSS-IE always requires at least one adaptive pass. Entering **1** will also bypass adaptive analysis, generating a solution only at the solution frequency you specified.
7. If ports exist, accept or set the Maximum Delta S per pass. If ports do not exist accept or set the **Maximum Residual Error** for convergence per pass.

   Maximum Residual Error is a stopping criterion for the adaptive solution. If the residual error is less than this value from one iteration to the next, the adaptive analysis stops. Otherwise, it continues until the requested number of passes is completed. In HFSS-IE this is an absolute value that functions like the Maximum Delta Energy, a relative value in HFSS that do not have ports. The default is 0.001.
8. The lower right corner also contains a button for HPC and Analysis options. Here you can select or create an analysis configuration.
9. Click the **Options** tab.

Under the **Options** tab of the **Solution Setup** dialog box, you can edit the following settings:

Do Lambda Refinement

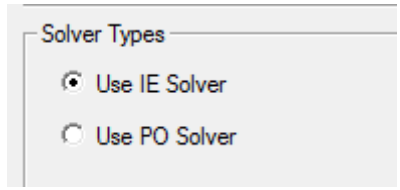
Use Free Space Lamda

Maximum Refinement Per Pass

Minimum Number of Passes

Minimum Converged Passes

Solver Types lets you select either one of the following (not both at once):



Use PO Solver - for designs with a plane incident wave, far field wave, or near field wave.

Use IE Solver optionally uses a distributed memory paradigm to solve large problems by spreading the memory use across multiple machines in a cluster. Note that shared memory parallel in combination with distributed IE is supported. That means a distributed solve can use multiple threads, based on the HFSS-IE solver option, Number of Processors Distributed. For Linux, you can set the Remote Spawn command to use, RSH, or SSH (the default), on the HFSS-IE Solver Options.

For details see Distributed Memory Solutions with HFSS-IE.

10. Under the **Advanced** tab you can specify whether to Import the mesh from another design or project that has an identical geometry.
11. Under the **Expression Cache** tab of the **Solution Setup**, you can edit the following settings:

    Adaptive Options: whether to use Output Variable Convergence (output variables must be defined for this to be enabled.)

    Add, Edit, Remove, or Remove all expressions.

    Also use selected expressions for convergence. Checking this enables the radio button and field for either Max Delta or Max Percent Delta.

    For expressions in the cache, you can directly edit the name, and, by clicking the Intrinsics field for an expression, you can edit the sweep values to which the expression applies.
12. Click **OK**.
13. Optionally, add a frequency sweep to the solution setup.

Related Topics

Setting Adaptive Analysis Parameters

Technical Notes: The HFSS Solution Process

Copying a Solution Setup

Renaming a Solution Setup

HFSS-IE Feature

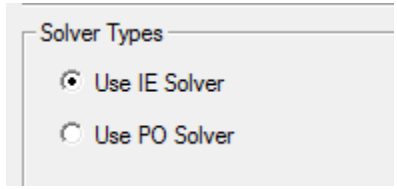
Distributed Memory Solutions within HFSS-IE

Use PO Solver for HFSS-IE Solve Setup

HFSS-IE includes a Physical Optics (PO) solver for use with metallic designs excited by an incident plane wave, a far field wave, or a near field wave. This solver provides first-order scattering information, such as an approximate RCS. If that result is not accurate enough (for example, for a back lobe calculation), you can use the regular HFSS-IE solver. The types of reports are the same; it is the accuracy that changes. When selected, it solves for only one pass, performing no adaptive refinement.

Dielectric materials are not allowed with the PO solver. PEC and finite conducting objects yields the same answers. The design cannot contain an Infinite Ground Plane or an Aperture Boundary condition. If the design contains any conducting boundary other than PEC, for example: Finite Conductivity, Impedance, Lumped RLC or Layered Impedance, then a warning message says that "The PO solver only handles PEC boundaries; all other boundary conditions, including finite conductivity, will be treated as PEC."

You control the option by using the **Use PO Solver** radio button on the **Solve Options** tab of the HFSS-IE Solution Setup. You can select either **Use IE Solver** or the **Use PO Solver** option.



Related Topics

Adding a Solution Setup to an HFSS-IE Design

Distributed Memory Solutions with HFSS-IE

The HFSS-IE solver in HFSS uses the industry standard Message Passing Interface ("MPI") and can perform solutions that distribute memory use across machines in a cluster or network. Memory used by the MPI-enabled HFSS-IE solver is therefore limited by the set of machines that are available rather than the shared memory available on any single machine. This allows you to simulate larger structures than before and to optimally reconfigure the cluster of machines for the problem at hand.

To use the distributed memory solution in HFSS-IE you will need to install HFSS and MPI software from one of the supported third party vendors on all the machines you intend to use. You may need to set passwords depending on the MPI vendor for authentication on the machines. Settings within HFSS are used to turn on distributed memory solutions and define the list of machines you intend to use. Detailed instructions about how to get distributed memory HFSS-IE solutions up and running are outlined in the following sections.

- Installation Requirements for Distributed Memory Solutions with HFSS-IE
- Setting up HFSS-IE and Running Distributed Memory Solutions
- Select the MPI Vendor for HFSS-IE
- Running Distributed Memory Solutions from the Command line
- Discussion of HFSS-IE Distributed Memory Solution
- Interconnects for HFSS-IE Distributed Memory Simulation
- Authentication on Linux (RSH & SSH) for HFSS-IE
- Troubleshooting for HFSS-IE Distributed Memory Solutions

Installation Requirements for Distributed Memory Solutions with HFSS-IE

You must install HFSS on all the machines you intend to use during the HFSS-IE distributed memory solution process. The installation locations for HFSS must be identical and the machines must be uniform: all 64 bit or all 32 bit; all Windows or all Linux. The machines must all use the same interconnect.

In addition, on Windows, you must install one of the supported versions of MPI from either Platform Computing (default) or Intel. Be sure to install the same version of MPI on all machines in your cluster. (Solving on a single Windows machine does not require MPI installation. And users running on Linux do not need to install MPI manually).

Platform	MPI Software
Linux Intel & AMD	Platform MPI 8.1 Intel MPI 4.0.1
Windows Vista, Windows 7 (32 and 64 bit)	Platform MPI 8.1 Intel MPI 4.0.1

You will need to set the password you want to use for your MPI runs on all the machines in the cluster. You can either use the batch command provided by ANSYS to set your password or refer to the MPI vendor's documentation.

Related Topics

Distributed Memory Solutions with HFSS-IE

Setting up HFSS-IE and Running Distributed Memory Solutions

Select the MPI Vendor

Running Distributed Memory Solutions from the Command line

Discussion of HFSS-IE Distributed Memory Solution

Interconnects for HFSS-IE Distributed Memory Simulation

Authentication on Linux (RSH & SSH)

Troubleshooting for HFSS-IE Distributed Memory Solutions

Setting up HFSS-IE and Running Distributed Memory Solutions

After setting up your HFSS-IE project normally you will need to set the list of machines used for the distributed memory solution process and turn on the distributed memory option.

To create the distributed machine list see Configuring Distributed Analysis.

Related Topics

Distributed Memory Solutions with HFSS-IE

Installation Requirements for Distributed Memory Solutions with HFSS-IE

Select the MPI Vendor

Running Distributed Memory Solutions from the Command line

Discussion of HFSS-IE Distributed Memory Solution

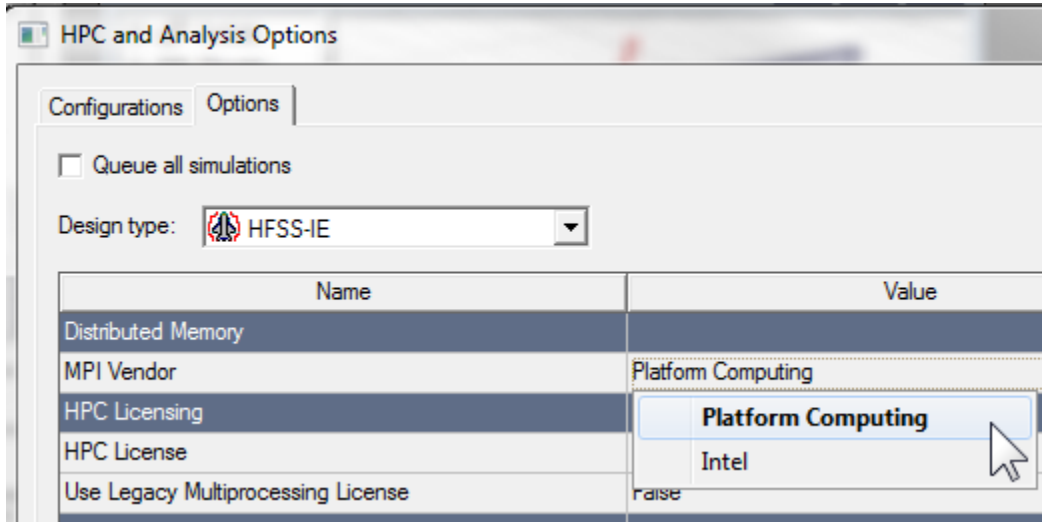
Interconnects for HFSS-IE Distributed Memory Simulation

Authentication on Linux (RSH & SSH)

Troubleshooting for HFSS-IE Distributed Memory Solutions

Select the MPI Vendor for HFSS-IE

After installing MPI on your machine from a particular vendor such as Platform Computing or Intel you need to set which type of MPI you are using in HFSS. Go to the **Options** tab of the Tools>Options>HPC and Analysis Options dialog to set the MPI Vendor type.



Related Topics

Distributed Memory Solutions with HFSS-IE

Installation Requirements for Distributed Memory Solutions with HFSS-IE

Setting up HFSS-IE and Running Distributed Memory Solutions

Running Distributed Memory Solutions from the Command line

Discussion of HFSS-IE Distributed Memory Solution

Interconnects for HFSS-IE Distributed Memory Simulation

Authentication on Linux (RSH & SSH)

Troubleshooting for HFSS-IE Distributed Memory Solutions

Running Distributed Memory Solutions from the Command line for HFSS-IE

You can run distributed memory HFSS-IE solutions from the command line using the -BatchSolve option. Set the distributed memory solve setup option before running the simulation and use the "BatchSolve" flag with the "Distributed" and MachineList" options. For example:

```
hfss -BatchSolve -Distributed -MachineList list="machine1,_
machine2" TheProject.hfss
```

This simulates "TheProject.hfss" as a distributed memory solution on machines "machine1" and "machine2."

Note the distributed memory solve setup option can be turned on via scripting if desired.

Related Topics

Distributed Memory Solutions with HFSS-IE

Installation Requirements for Distributed Memory Solutions with HFSS-IE

Setting up HFSS-IE and Running Distributed Memory Solutions

Select the MPI Vendor

Discussion of HFSS-IE Distributed Memory Solution

Interconnects for HFSS-IE Distributed Memory Simulation

Authentication on Linux (RSH & SSH)

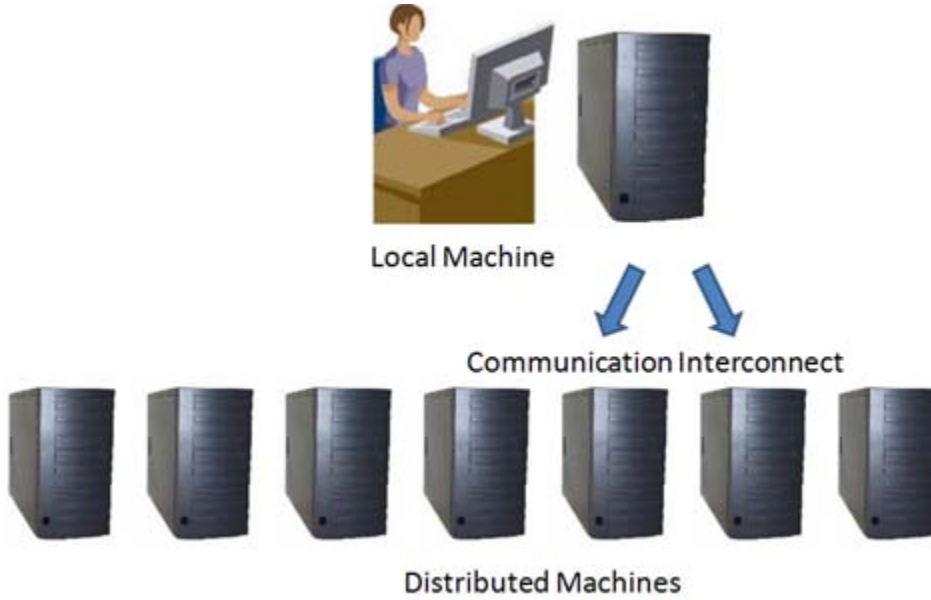
Troubleshooting for HFSS-IE Distributed Memory Solutions

Discussion of HFSS-IE Distributed Memory Solutions

Each machine or "compute node" is connected to the other nodes via a "communication interconnect" and relies on the message passing library (MPI) to exchange data and synchronize computational tasks. Ethernet, Myrinet and Infiniband are common communication interconnects. Each node is identified by a unique integer ID or rank number. The local machine is known as the "Rank-0 node" and is the master. The Rank-0 machine has many tasks including:

- Management of all communication with the HFSS user interface. None of the distributed machines communicate directly with the user interface but pass all information through the Rank-0 machine.
- Mesh generation. The mesh is generated only on the Rank-0 machine.
- Disk access. None of the distributed machines access their local discs. The simulation mesh, intermediate and solution data are passed to and from the distributed machines using MPI.
- Distribution and control of computational tasks on the distributed machines. The simulation process is dynamic and the Rank-0 machine will determine which of the distributed machines has memory available to distribute tasks accordingly.

- Post-processing of the HFSS-IE Distributed Memory Solution



The algorithms used in the distributed memory version of the HFSS-IE solver engine resemble those used in the non-distributed memory version. The matrix solution algorithms in the distributed memory version of HFSS-IE have been adjusted to use slightly more memory so that larger problems can be simulated in less time.

The MPI enabled HFSS-IE solver engine is now multi-threaded. That means a distributed solve can use multiple threads, based on the HFSS-IE solver option, Number of Processors Distributed. For Linux, you can set the Remote Spawn command to use, RSH, or SSH (the default), on the HPC and Analysis Options dialog..If a particular machine has multiple cores and enough memory you can define this machine several times in the distributed machine list ("doubling up") to take advantage of the extra cores.

During the "Matrix Assembly" and "Matrix Solve" steps of the solution process the HFSS-IE engine attempts to distribute memory use evenly. At various points in the matrix solution process the software will poll the machines in the cluster and determine which machine has the most memory available and then reserve a block of memory on that machine. If a particular machine does not have a large block of memory available the memory use on that machine will grow only slowly. If none of the machines in the cluster have sufficient memory the solution process will terminate and an error message will be posted to the HFSS message window.

Note If you list the same machine several times in the Distributed Machine Configuration you can easily overload that machine. For example, consider a setup with the following machines in the Distributed Machine Configuration

MachineA

MachineB

MachineA

MachineA

If you set "Number of Processors, Distributed = 8" this means that you request 24 (=3*8) processors on MachineA and 8 on MachineB. If MachineA does not have 24 processors the simulation will be inefficient.

Many factors affect solution time. In general, the solution time will decrease as the number of compute nodes increases. However, parallel efficiency decreases as the ratio of communication to computation increases so to some extent you need to match the size of the problem to the size of parallel machine. Simulating small structures on a large cluster will not be efficient and may take longer and use significantly more memory than if the structure was simulated on a single machine.

Network interconnect speed and topology can affect performance significantly in homogeneous clusters. Performance can degrade if machines are "doubled up" to the point of causing memory bus contention or if the cluster is significantly inhomogeneous and certain faster machines need to wait for slower machines to catch up to synchronization points in the solution process.

Related Topics

Distributed Memory Solutions with HFSS-IE

Installation Requirements for Distributed Memory Solutions with HFSS-IE

Setting up HFSS-IE and Running Distributed Memory Solutions

Select the MPI Vendor

Running Distributed Memory Solutions from the Command line

Interconnects for HFSS-IE Distributed Memory Simulation

Authentication on Linux (RSH & SSH)

Troubleshooting for HFSS-IE Distributed Memory Solutions

Interconnects for HFSS-IE Distributed Memory Simulation

To obtain the best possible performance we recommend the use of a network interconnect that supports communication speeds greater than 1000MB/sec or higher. Some high performance interconnects plug into a PCI (Peripheral Component Interconnect), PCI-X (extended), or PCIe (PCI Express) slot on the system.

HFSS-IE 14 supports the following network interconnects:

Platform	Interconnects
Win32	Ethernet/GiGE
Win64	Ethernet/GiGE (default), Myrinet, Infiniband
Linux	Ethernet/GiGE (default), Myrinet, Infiniband

Ethernet/GiGE is the default interconnect on all platforms. You can choose one of the alternate interconnects by setting the ANSOFT_MPI_INTERCONNECT environment variable to "myri" for Myrinet and "ib" for Infiniband.

Interconnect variants are supported on Linux. Set the ANSOFT_MPI_INTERCONNECT_VARIANT to the desired interconnect variant. For example, set "ANSOFT_MPI_INTERCONNECT_VARIANT=silverstorm" to use the silverstorm variant.

Related Topics

Distributed Memory Solutions with HFSS-IE

Installation Requirements for Distributed Memory Solutions with HFSS-IE

Setting up HFSS-IE and Running Distributed Memory Solutions

Select the MPI Vendor

Running Distributed Memory Solutions from the Command line

Discussion of HFSS-IE Distributed Memory Solution

Authentication on Linux (RSH & SSH)

Troubleshooting for HFSS-IE Distributed Memory Solutions

Authentication on Linux (RSH & SSH) for HFSS-IE

An important step in using a high performance cluster is setting up authentication across machines in such a way that the machines can be accessed without a password. By default HFSS-IE uses SSH authentication on Linux to spawn commands on the remote machines but also supports RSH. The selection of which to use is made on the **Options** tab of the Tools>Options>HPC and Analysis dialog.

Related Topics

Distributed Memory Solutions with HFSS-IE

Installation Requirements for Distributed Memory Solutions with HFSS-IE

Setting up HFSS-IE and Running Distributed Memory Solutions

Select the MPI Vendor

Running Distributed Memory Solutions from the Command line

Discussion of HFSS-IE Distributed Memory Solution

Interconnects for HFSS-IE Distributed Memory Simulation

Troubleshooting for HFSS-IE Distributed Memory Solutions

Troubleshooting for HFSS-IE Distributed Memory Simulations

A number of things can prevent distributed memory solutions from completing successfully. This section provides suggestions to debug problems.

It is often a good idea to set up and run a small simulation with two processes on a single machine before moving to a large cluster. Using a single machine will allow you to verify that HFSS and MPI are installed correctly while eliminating problems arising from remote installation, authentication and firewall settings.

Many problems occur because the MPI software cannot start and run due to authentication and firewall issues. Please check with your MPI vendor and their end user documentation for information about how to verify that authentication and firewall settings are correct.

If you are using MPI on Windows from Platform Computing you can test whether MPI will run by using the "mpidiag" utility:

- From a command prompt browse to the Platform Computing binaries located at

  `<HFSS_Installation_Directory>\common\fluent_mpi\multiport\mpi\_win64(win32)\pcmpi\bin`
- Enter "mpidiag -s <name_of_machine> -at" to run an authentication test.
- Run the authentication tests in both directions, i.e. both to and from all target machines. This will verify that MPI passwords and firewall settings are correct.

  The tests need to be run in both directions because firewalls may allow communication in one direction but not the other.

Platform Computing's implementation of MPI on Windows requires that you enter a password on each machine in the cluster to run MPI solutions.

- From a command prompt browse to the Platform Computing binaries located at

  `<HFSS_Installation_Directory>\common\fluent_mpi\multiport\mpi\_win64(win32)\pcmpi\bin`
- To set the password run "mpidiag -s <name_of_machine> -cache -at" and enter the password at the prompt.

Recall that HFSS must be installed in the same directory on all machines in the cluster and that the cluster must be uniform (i.e. all Linux machines, all 32 bit Windows machines, or all 64 bit Windows machines)

\Verify that the version of the third party MPI software is identical on all the machines and that it is listed in the table of supported versions above.

Verify that the machine names are correct and that all the machines can be reached on the network.

Related Topics

Distributed Memory Solutions with HFSS-IE

Installation Requirements for Distributed Memory Solutions with HFSS-IE

Setting up HFSS-IE and Running Distributed Memory Solutions

Select the MPI Vendor

7

Drawing a Model

After you insert a design into the current project, you can draw a model of the electromagnetic structure. The general strategy is to build the model as a collection of 3D objects. You can assign any single material to each 3D object.

You can create 3D objects by using the modeler's **Draw** commands or you can draw 1D and 2D objects, and then manipulate them to create 3D objects. Objects are drawn in the **3D Modeler** window. You can also import objects from other systems.

To open a new **3D Modeler** window, do one of the following:

- Insert a new design into the current project.
- Double-click an HFSS design in the project tree.

If a **3D Modeler** window for an existing design is not open, do one of the following:

- Click **HFSS**>**3D Model Editor**.
- Right-click the design name in the project tree, and then click **3D Model Editor** on the shortcut menu.

The model you draw is saved with the current project when you click **File>Save**.

Note If you access your machine via Remote Desktop, if HFSS is running and one or more modeler windows are open, those modeler windows automatically close. The message manager window displays a message indicating that HFSS closed the modeler windows.

When working with multiple projects, or when a project has multiple designs, you may have multiple **Modeler** windows available. To switch to the modeler window associated with a specific design:

1. In the Project Manager window, select the **Design** of interest.
2. Click **HFSS3D Model Editor** to focus the modeling window on the selected design.

If the menu command is unavailable, then the selected design is already in the modeler window.

Related Topics

Setting the Units of Measurement for the Model

Drawing Objects

Model Analysis

Design Settings

Setting the Temperature of Objects

Creating a User Defined Primitive

Creating a 3D Component from an Existing Model

3D Component Library

Modifying Objects

Selecting Objects

Choosing the Movement Mode

Choosing the Snap Settings

Measure Modes for Objects

Setting Coordinate Systems

User Defined Model (UDM)

Setting Units of Measurement for the Model

You can specify the units of measurement for drawing geometric models. After the units of measurement have been specified, they are assigned to the objects in the **3D Modeler** window. You can then choose to display the model's dimensions in the new units, or rescale the model's dimensions to the new units.

To set the model's units of measurement:

1. Click **Modeler>Units**.

   The **Set Model Units** dialog box appears.

2. Select the new units for the model from the **Select units** pull-down list.
3. Specify how the change in units affects the model:
   - Select the **Rescale to new units** option to rescale the dimensions to the new units. For example, selecting centimeters (cm) as the new unit of measurement results in a dimension of 10 millimeters (mm) becoming 10 cm.
   - Clear the **Rescale to new units** option (the default) to convert the dimensions to the new units without changing their scale. For example, selecting cm as the new unit of measurement results in a dimension of 10 mm becoming 1 cm.

Click **OK** to apply the new units to the model.

Drawing Objects

You can draw one-, two-, or three-dimensional objects using the **Draw** commands. You can alter objects individually or together to create the geometry of your structure. In the **Tools>Modeler Options, Drawing** tab, you can set a default to either draw objects directly with the mouse or by invoking a **Properties** dialog in which you can enter the values for the object dimensions. The **Dialog** mode drawing feature works with the equation based curve, equation based surface, and all two and three dimensional objects. You can toggle to **Point** mode via the **F3** function key and to **Dialog** mode via the **F4** function key. When you use the **Dialog** mode for drawing objects the Edit property of new primitives setting is ignored.

One-dimensional (1D) objects in the modeler include straight line, arc line, center-point arc, and spline segments, or a combination of these - called polylines. One-dimensional objects are open objects; their boundaries do not enclose a region, unless you connect their endpoints. They have length, but no surface or volume. Generally they are used as temporary objects from which to create 2D objects.

Two-dimensional (2D) objects in the modeler include objects such as equation based surfaces, rectangles, ellipses, circles, and regular polygons. Two-dimensional objects are closed sheet objects; their boundaries enclose a region. You can create 2D sheet objects by covering the enclosed region. In many applications (FSS, antennas) it is essential to calculate net power flow through a surface.

You can also edit the properties of a polyline from the history tree to assign it a Cross Section property as line or rectangular. If you then assign it either a height or a width, the polyline becomes a sheet object.

By default, the history tree organizes sheet objects according to their boundary assignments. To change this, select the **Sheets** icon, and right-click to display the **Group Sheets by Assignment** checkbox. Within the calculator sheet objects are listed under surface.

Three-dimensional (3D) objects in the modeler include objects such as boxes, cylinders, regular polyhedra, cones, spheres, torii, and helices. These objects have boundaries that enclose a region with volume.

You can create 3D objects by manipulating 2D objects along a plane or by using the appropriate **Draw** commands. You can also edit the properties of a polyline from the history tree to assign it a Cross Section property as circle rectangular. If you then assign it an appropriate diameter or both height or a width, the polyline becomes a 3D object.

By default, the history tree groups 3D objects by material. To change this, select the **Objects** icon, and right click to display the **Group Objects by Material** checkbox.

While you draw objects you can also:

- Select Movement Mode as 3D, In Plane, Out of Plane, Along X, Y or Z axis.
- Select Grid Plane as XY, YZ, or XZ.
- Set the Drawing Plane
- Set Snap Mode
- Set Reference Point for the movement mode
- Adjust the View

After you draw an object in the **3D Modeler** window, you can modify the object's properties, such as its position, dimensions, or color, in the **Properties** dialog box. Most model object properties can be assigned as Design variables when can then be manipulated during the solve to test their effect on the solution. For non-model objects, you can use Post Processing variables (default and Design variables,.

Note If you access your machine via Remote Desktop, if HFSS is running and one or more modeler windows are open, those modeler windows automatically close. The message manager window displays a message indicating that HFSS closed the modeler windows.

Related Topics

Assigning a Cross Section and Dimension to a Polyline

Modifying Objects

Drawing a Region

Setting the Temperature of Objects

User Defined Model (UDM) for ANSYS WB Integration

Drawing a Straight Line Segment

To create an object with one or more straight line segments, use the **Draw>Line** command.

1. Click **Draw**>**Line**  .
2. Select the first point of the line in one of the following ways:
   - Click the point.
   - You can accept the point or change it by editing its coordinates in the X, Y, and Z boxes in the status bar.

   To delete the last point that was entered, click **Undo Previous Segment** on the shortcut menu. After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.
3. Select the endpoint of the line by clicking the point or typing the coordinates in the text boxes in the status bar.

   The endpoint serves as the start point for a subsequent line segment.

   To delete all points and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.
4. Complete the line in one of the following ways:
   - Double-click the endpoint.
   - Click **Done** on the context (right-click) menu.
   - Press **Enter**.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes by editing the Properties. Those listed under the Command tab describe the commands used to create the object. These commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transpar-

ency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.

5. Click **OK** to close the Properties dialog.

Note While drawing a polyline, you can switch between straight line, arc line, or spline segments using the **Set Edge Type** commands on the shortcut menu.

Related Topics

Setting the Reference Point

Deleting Polyline Segments

Converting Polyline Segments

Drawing a Three-Point Arc

Drawing a Center-Point Arc Line

Drawing a Spline

Drawing a Polyline

Drawing a Three-Point Arc Line

In the modeler, a three-point arc line segment is an arced line defined by three points on its curve. Use the **Draw>Arc>3 Point** command to create a polyline object with one or more arc line segments.

1. Click **Draw>Arc>3 Point**  .
2. Select the start point of the arc in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the **X**, **Y**, and **Z** text boxes.
3. Select the midpoint of the arc by clicking the point or typing the coordinates in the text boxes in the status bar.

   To delete the last point that was entered, click **Undo Previous Segment** on the shortcut menu. After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.

   To delete all points and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.
4. Select the endpoint of the arc by clicking the point or typing the coordinates in the text boxes in the status bar.

   The endpoint serves as the start point for a subsequent arc line segment.
5. If the endpoint is the last point of the polyline object, double-click the point to complete the polyline or click **Done** on the shortcut menu.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes by editing the Properties. Those listed under the Command tab describe the commands used to create the object. These

commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.

6. Click **OK**.

   Based on the three points you specified, the modeler calculates the center point and radius of the arc and draws an arced line through the three points.

Note While drawing a polyline, you can switch between arc line, straight line, or spline segments using the **Set Edge Type** commands on the shortcut menu.

Related Topics

Setting the Reference Point

Drawing a Straight Line

Drawing a Spline

Drawing a Center-Point Arc Line

Drawing a Polyline

Deleting Polyline Segments

Converting Polyline Segments

Drawing a Center-Point Arc Line

In the modeler, a center-point arc line segment is an arced line defined by a center point, start point and angle. Use the **Draw>Arc>Center Point** command to create a polyline object with one or more center-point arc line segments.

1. Click **Draw**>**Arc**>**Center Point**  .
2. Select the center point of the arc in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Select the start point, or radius, of the arc by clicking the point or typing the coordinates in the text boxes in the status bar.

   To delete the last point that was entered, click **Undo Previous Segment** on the shortcut menu. After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.

   To delete all points and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.
4. Sweep the angle, or endpoint, of the arc by clicking the point or typing the coordinates in the text boxes in the status bar.
5. If the endpoint is the last point of the polyline object, double-click the point to complete the polyline or click **Done** on the shortcut menu.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's attributes.

6. Click **OK**.

Note While drawing a polyline, you can switch between arc line, straight line, or spline segments using the **Set Edge Type** commands on the shortcut menu.

Related Topics

Setting the Reference Point

Drawing a Straight Line

Drawing a Spline

Drawing a Polyline

Deleting Polyline Segments

Converting Polyline Segments

Drawing a Three-Point Arc

Drawing a Spline

A spline is a curved line defined by three points. The modeler uses a natural spline type: a piece wise cubic spline with an end condition that has a derivative of zero. Use the **Draw>Spline** command to create a polyline object with one or more spline segments.

1. Click **Draw**>**Spline**  .
2. Select the spline's start point in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar, and then press **Enter**.

   To delete the last point entered, click **Undo Previous Segment** on the shortcut menu.After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.

   To delete all selected points and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.
3. Select the midpoint of the spline by clicking the point or typing the coordinates in the text boxes in the status bar.
4. Select the endpoint of the spline by clicking the point or typing the coordinates in the text boxes in the status bar.

   The endpoint serves as the start point for a subsequent spline segment.
5. Complete the spline in one of the following ways:
   - Double-click the endpoint.
   - Click **Done** on the shortcut menu.
   - Press **Enter**.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes by editing the Properties. Those listed under the Command tab describe the commands used to create the object. These

commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.

6. Click **OK**.

Note While drawing a polyline, you can switch between spline, straight line, or arc line segments using the **Set Edge Type** commands on the shortcut menu.

Related Topics

Setting the Reference Point

Drawing a Polyline

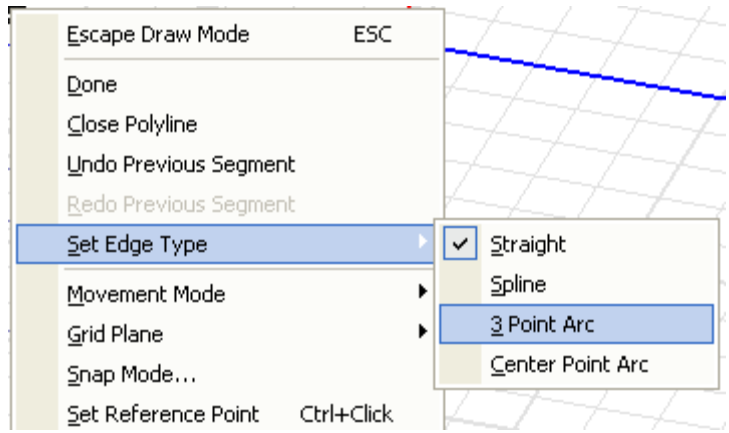
Drawing a Center-Point Arc Line

Deleting Polyline Segments

Converting Polyline Segments

Drawing a Polyline

A polyline is a single object that includes any combination of straight line, arc line, or spline segments. The endpoint of one segment is the start point for the next segment. Use the shortcut menu's **Set Edge Type** commands to switch between straight line, arc line, or spline segments while drawing a polyline.



In the **Polyline** section of **Operation** tab of the **Modeler Options**, select or clear the **Automatically cover closed polylines** check box.

If checked, closed polylines become sheet objects, and are listed as such in the History tree. If unchecked, closed polylines are listed under lines in the History tree.

1. Click **Draw**>**Line**  .
2. Right-click in the **3D Modeler** window to access the shortcut menu, and then point to **Set Edge Type**.
3. Click **Straight**, **Spline**, **3 Point Arc**, or **Center Point Arc** depending on which type of polyline segment you want to draw.
4. If you clicked **Straight**, follow the procedure for drawing a straight line.

   If you clicked **Spline**, follow the procedure for drawing a spline.

   If you clicked **3 Point Arc**, follow the procedure for drawing a three-point arc line.

   If you clicked **Center Point Arc**, follow the procedure for drawing a center-point arc line.
5. Repeat steps 2 and 3 for each segment of the polyline object. The endpoint of the previous segment serves as the start point for the next segment.

   The shortcut menu lets you do the following for each segment:

   Undo Previous Segment or **Redo Previous Segment**.
6. Complete the polyline in one of the following ways:
   - Double-click the endpoint of the final segment.
   - Click **Done** on the shortcut menu.

Note To connect the polyline's start and endpoints, click **Close Polyline** on the shortcut menu.

If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes by editing the Properties. Those listed under the Command tab describe the commands used to create the object. These commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line. Notice that by going to the History tree selecting Create Polyline for that object, you can assign a cross section and dimensions to a polyline.

7. Click **OK**.

If you select a polyline in the History tree, you can use the Measure mode to see the total length.

Related Topics

Setting the Reference Point

Assigning a Cross Section and Dimension to a Polyline

Drawing a Straight Line

Drawing a Three-Point Arc

Drawing a Center-Point Arc Line

Drawing an Equation-Based Curve

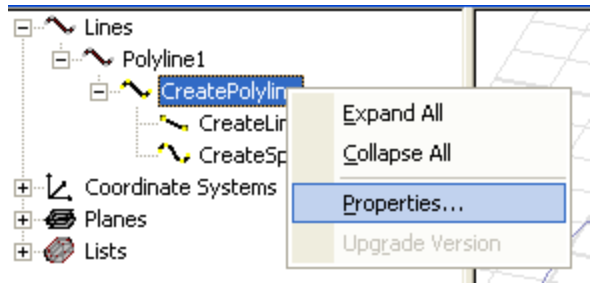
Drawing a Spline
Deleting Polyline Segments
Converting Polyline Segments
Generate History
Setting Modeler Options: Operations Tab

Assigning a Cross Section and Dimensions to a Polyline

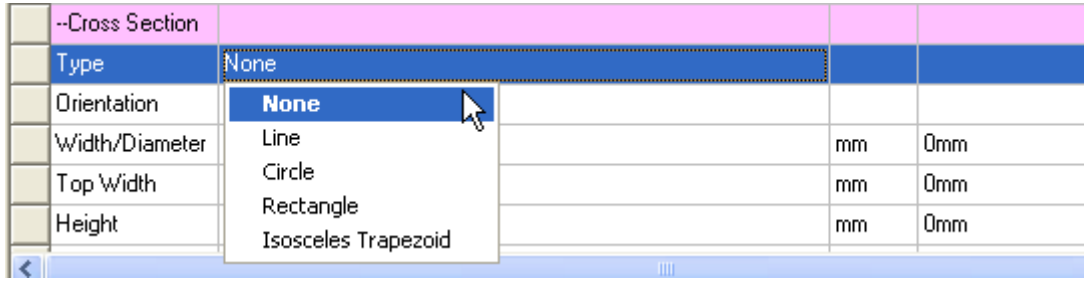
By viewing the History tree property of a polyline, you can assign either a line, circle, rectangle or Isosceles trapezoid cross section to a polyline. This assignment enables editable dimension properties of width for a line, diameter for a circle, and height and width for a rectangle or trapezoid. To assign a cross section to a polyline:

1. In the History tree of the Modeler window, right-click on the polyline that you want to give a cross section.

   This selects the polyline, displays the polyline properties in the docked properties (if you have if displays) and displays a shortcut menu where you can choose **Properties...** to display the undocked Properties window for the polyline.



2. In a Properties window (either docked or undocked) for the selected polyline click on None on the Type line under Cross Section to display the choices for Line, Circle, Rectangle and Isosceles Trapzoid.



3. Select one of Line, Circle, Rectangle, or Isosceles Trapezoid as the cross section.
   - Selecting **Line** causes the Cross Section area of the polyline properties to display editable fields for Orientation and Width.

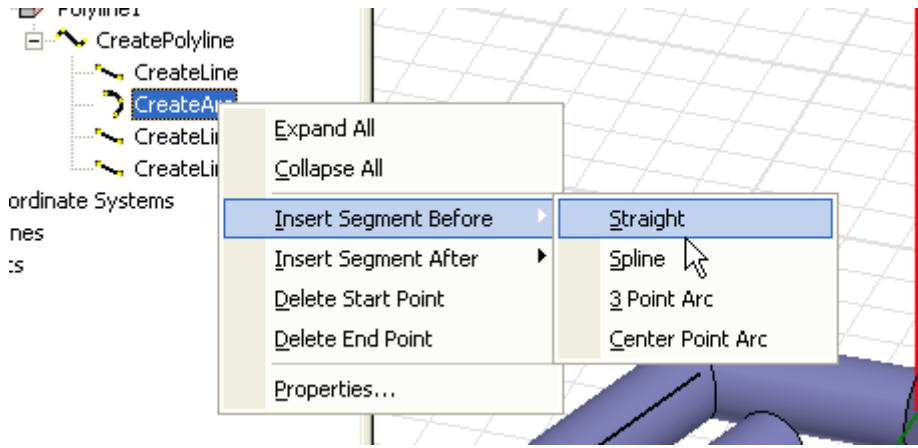
- Selecting **Circle** causes the Cross Section area of the polyline properties to display an editable field for diameter.
- Selecting **Rectangle** causes the Cross Section area of the polyline properties to display editable fields for Orientation, Width and Height.
- Selecting **Isosceles Trapezoid** causes the Cross Section area of the polyline properties to display editable fields for Orientation, Width/Diameter, Top Width, and Height.

4. If you select Line, Rectangle, or Isosceles Trapezoid you can edit the Orientation as Auto, X, Y, or Z. This provides the direction in which the dimension extends.
5. Specify the dimensions and select the units for the Cross section.

   Type a value in the dimension field(s) and select units from the drop down menu.

   The dimensions must be reasonable relative to the specified shape and orientation of the polyline. If the polyline cannot be extended into current Orientation for the given dimension(s), you will receive a warning. If you receive a warning, check the Orientation, dimension and units.

   When the modeler can extend the dimensions legally, it displays the modified object, and lists it in the History tree as either a Sheet object (Line or one dimensional Rectangle) or as a Solid object (Circle or two dimensional Rectangle).
6. You can modify the new polyline either by editing the properties, or by using the History tree to select one of the line objects that make up the polyline, and right clicking to display the popup menu showing commands to **Insert**, **Delete**, or display editable segment **Properties**.



A segment that you select in the Project tree is indicated in the Modeler window by a line in the dimensioned object. If you insert a new segment, it adopts the dimensions you specified for the polyline object.

Related Topics

Setting the Reference Point

Drawing a Polyline

Drawing an Equation-Based Curve

Deleting Polyline Segments

Inserting Line Segments

Drawing a Center-Point Arc Line

Drawing a Straight Line

Drawing a Three-Point Arc

Inserting Line Segments

You can insert line segments of various kinds for existing line objects.

1. Select the line object in the **History** tree (not the modeler window).

   This highlights the object and enables the **Insert Line Segment** commands in the **Draw** menu and short-cut menu.
2. Use the cascade menu from the **Draw>Line Segment** command to or the right-click menu to select whether to **Insert Before Line Segment** or **Insert After Line Segment**.
3. Use the next cascade menu to specify the kind of segment to add. These can be: Straight, Spline. 3 Point Arc, or Center Point Arc.
4. If you clicked **Straight**, follow the procedure for drawing a straight line.

   If you clicked **Spline**, follow the procedure for drawing a spline.

   If you clicked **3 Point Arc**, follow the procedure for drawing a three-point arc line.

   If you clicked **Center Point Arc**, follow the procedure for drawing a center-point arc line.
5. Repeat steps 2 and 3 for each segment of the polyline object. The endpoint of the previous segment serves as the start point for the next segment.
6. Complete the polyline in one of the following ways:
   - Double-click the endpoint of the final segment.
   - Click **Done** on the shortcut menu.

Note To connect the polyline's start and endpoints, click **Close Polyline** on the shortcut menu.

   The **Properties** dialog box appears, in which you can modify the object's attributes by editing the Properties. Those listed under the Command tab describe the commands used to create the object. These commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.
7. Click **OK**.

Related Topics

Drawing a Center-Point Arc Line

Deleting Polyline Segments

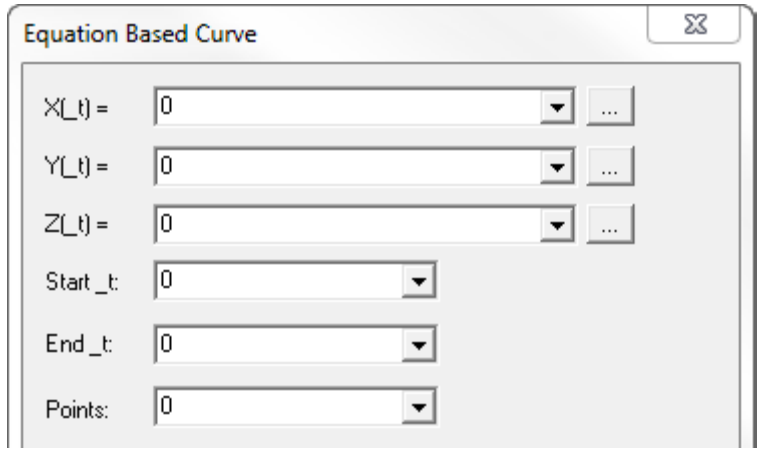
Converting Polyline Segments

Drawing an Equation-Based Curve

Any line that can be described by an equation in three dimensions can be drawn.

1. Click **Draw**>**Equation Based Curve** .

   The **Equation Based Curve** dialog box opens.

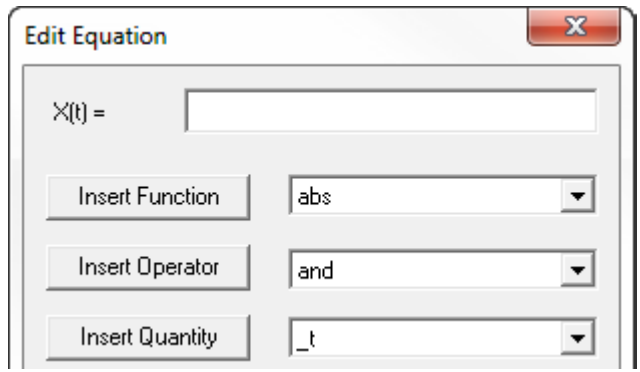


2. Type equations for **X(_t)**, **Y(_t)**, and **Z(_t)**

   You can also define an equation by doing the following:

   Click the **...** button.

   The **Edit Equation** dialog box appears.

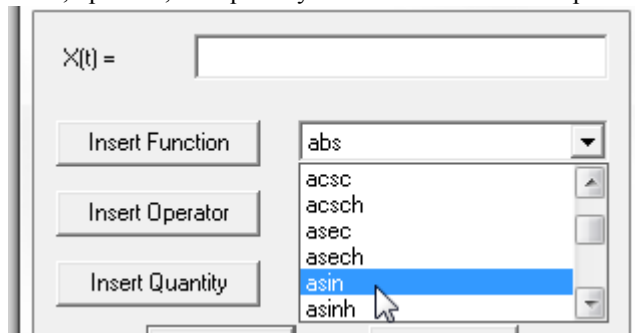


   c. Do one or more of the following to define the equation:

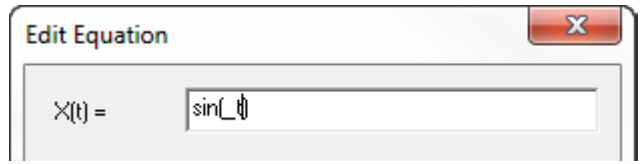
   Type a numerical value, variable, or expression directly in the text box.

   Build an equation for X(_t), Y(_t), or Z(_t) by specifying appropriate trignometric func-

tion, operator, and quantity to insert from the corresponding pull-down lists.



Select the respecting **Insert Quantity**, **Operator** or **Quantity** buttons to place the selections on the equation fields. An inserted Quantity goes inside a currently inserted Function.



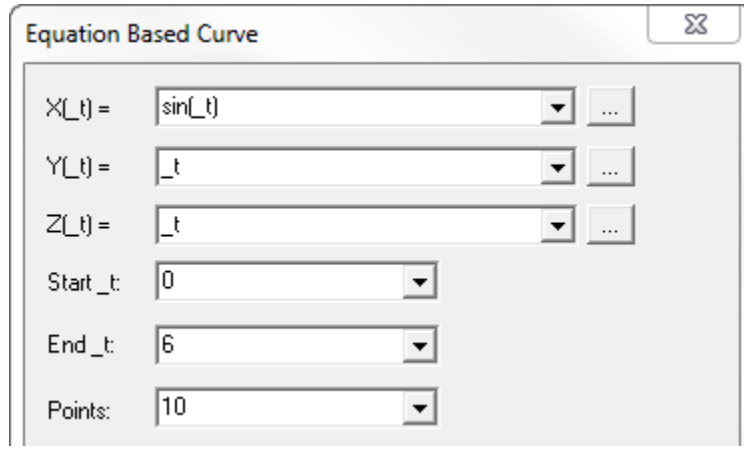
You can also type operators and values to build complex equations.

d. When you are finished defining the equation, click **OK** to close the **Edit Equation** dialog box and return to the **Equation Based Curve** dialog box.

- Any unitless value input in equation based curve is taken as model units. For example, for Y(_t) = 1, the y value is taken as 1 model units (say mm). If a value has units, then it is converted to model units and used. For example, if we specify Y(_t) = 1cm, then y value will be correctly taken as 10mm.
- While parsing expressions, equation based curves convert each variable separately to model units and assume that the resulting expression is in model units.
- Equation based curves depend on the variable value library to correctly evaluate the units of expression.

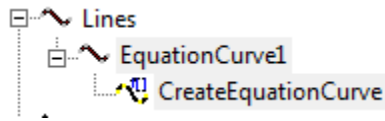
3. Select a start value from the **Start_t** pull-down list.
4. Select an end value from the **End_t** pull-down list.

5. Type in or select the number of points in the curve from the **Points** pull-down list.



6. Click **OK** on dialog to close it and create the curve.

   The curve appears in the modeler window, and the History tree shows the curve object and the command.



   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes by editing the Properties.

   The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.

Related Topics

Setting the Reference Point

Assigning a Cross Section and Dimension to a Polyline

Functions, Operators and Quantities for the Edit Equation Dialog

Functions, Operators and Quantities for the Edit Equation dialog

This table lists the functions, operators and Quantities for the Edit Equation dialog.

Item	Definition
	Functions
abs	Absolute value of the simulation quantity which results in a number that is always positive.
acos	Arc cosine i.e. the inverse function of a cosine.
acosh	Inverse hyperbolic arc cosine.
acot	Inverse cotangent
acoth	Inverse hyperbolic cotangent
acsc	Inverse cosecant
acsch	Inverse hyperbolic cosecant
asec	Inverse secant
asech	Inverse hyperbolic secant
asin	Arc sine i.e. inverse function of sine.
asinh	Inverse hyperbolic sine.
atan	Arc tangent i.e. the inverse function of a tan.
atanh	Inverse hyperbolic tan.
cos	Cosine.
cosh	Hyperbolic cosine.
cot	Cotangent
coth	Hyperbolic cotangent
csc	Cosecant
csch	Hyperbolic cosecant
ln	Natural logarithm.
log	Natural logarithm (same as ln).
sin	Sine.
sinh	Hyperbolic sine.
sqrt	Square root of the selected simulation quantity.
tan	Tangent.
tanh	Hyperbolic tangent.

Item	Definition
Operators	
and	
cross	
division	
dot	
equal	
exp	
exponent	
greater_than	
greater_than_or_equal	
not	
not_equal	
or	
Quantities	
_t	For equation based lines
_u,_v	For equation based surfaces.
PI	

Drawing a Circle

Draw a circle by selecting a center point and a radius. Circles are drawn as true surfaces in the modeler.

1. Click **Draw**>**Circle**  .
2. Select the center point of the circle in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the radius by selecting a point on the circle's circumference in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.
4. Click **OK**.

If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the circle will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

Note The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see:

Technical Notes, "Surface Approximations" and related sections, Rectilinear Elements and Curvilinear Elements, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Surface Approximation

Creating Segmented Geometry

Covering Lines

Drawing an Ellipse

Draw an ellipse by specifying a center point, base radius, and secondary radius.

1. Click **Draw**>**Ellipse**  .
2. Select the center point of the ellipse in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the base radius of the ellipse. If the current drawing plane is xy, then x is the base radius direction. If the drawing plane is yz, then y is the base radius direction. If the drawing plane is xz, then z is the base radius direction. Select the point in one of the following ways:
   - Click the point. HFSS constrains mouse movement to the base radius direction.
   - Type the coordinates of a point relative to the center point in the **dX**, **dY**, or **dZ** box, where **d** is the distance from the previously selected point.
4. Specify the secondary radius of the ellipse. Select the point in one of the following ways:
   - Click the point. HFSS constrains mouse movement to a point on the plane orthogonal to the base radius direction.
   - Type the coordinates of a point relative to the center point in the **dX**, **dY**, or **dZ** box.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.

The **Ratio** value represents the aspect ratio of the secondary radius to the base radius.

5. Click **OK**.

   If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the ellipse will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

If the base radius is larger than the secondary radius, the ellipse's longer axis will lie along the default base radius direction. If the secondary radius is larger than the base radius, the ellipse's longer axis will lie perpendicular to the default base radius direction. To create an ellipse with an arbitrary orientation, rotate or move the ellipse after drawing it.

Note The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see:

Technical Notes, "Surface Approximations" and related sections, Rectilinear Elements and Curvilinear Elements, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Modifying Surface Approximation Settings

Creating Segmented Geometry

Covering Lines

Drawing a Rectangle

Draw a rectangle (or square) by selecting two diagonally opposite corners.

1. Click **Draw**>**Rectangle**  .
2. Select the first diagonal corner in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

   To delete the selected point and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.
3. Select the second corner of the rectangle in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the first diagonal corner in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.

4. Click **OK**.

   If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the rectangle will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

Related Topics

Setting the Reference Point

Covering Lines

Drawing a Regular Polygon

A regular polygon is a 2D object with three or more equal sides. Regular polygons are useful for drawing faceted 2D objects.

1. Click **Draw**>**Regular Polygon**  .
2. Select the center point of the polygon in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the polygon's radius, the distance from the center point to one of the polygon's vertices, in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
4. In the **Segment number** dialog box, enter the **Number of segments** in the polygon, and then click **OK**.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.
5. Click **OK**.

Note The radius is measured from the center point to a corner of the polygon, or the intersection of two edges. It is *not* measured from the center point to the midpoint of an edge.

If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the polygon will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

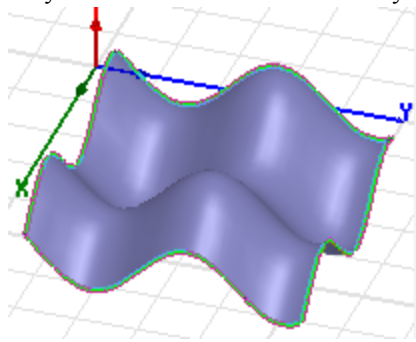
Related Topics

Setting the Reference Point

Covering Lines

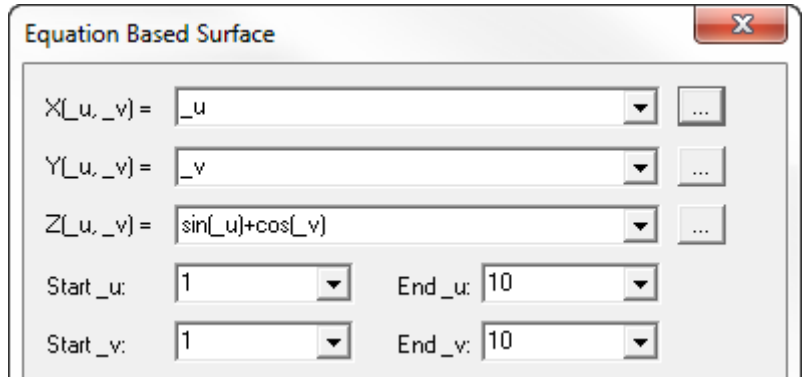
Drawing an Equation-Based Surface

Any surface that can be described by an equation in three dimensions can be drawn.



1. Click **Draw**>**Equation Based Surface** .

   The **Equation Based Surface** dialog box opens.

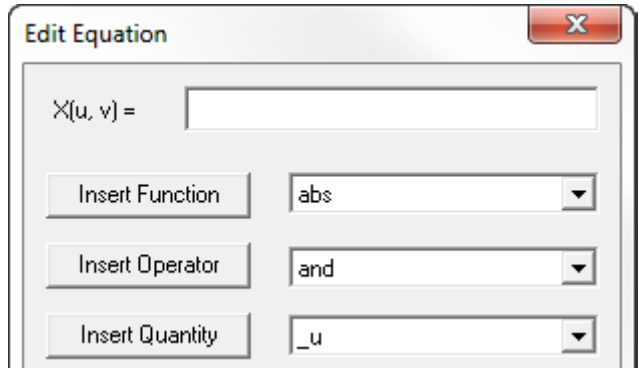


2. Type equations for **X(_u, _v)**, **Y(_u, _v)**, and **Z(_u, _v)**.

   You can also define an equation by doing the following:

   Click the **...** button.

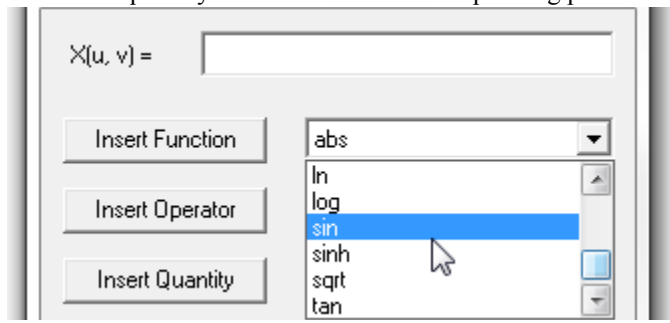
   The **Edit Equation** dialog box appears.



e. Do one or more of the following to define the equation:

Type a numerical value, variable, or expression directly in the text box.

Build an equation for X(u, v), Y(u, v), or Z(u, v), by selecting function, operator, and quantity to insert from the corresponding pull-down lists.



Select the respecting **Insert Quantity**, **Operator** or **Quantity** buttons to place the selections on the equation fields. An inserted Quantity goes inside a currently inserted Function.

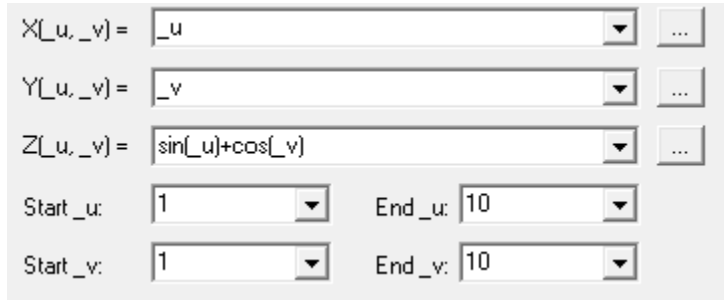
You can also type operators and values to build complex equations.

f. When you are finished defining the equation, click **OK** to close the **Edit Equation** dialog box and return to the **Equation Based Surface** dialog box.

- Any unitless value input in equation based curve is taken as model units. For example, for Y(_t) = 1, the y value is taken as 1 model units (say mm). If a value has units, then it is converted to model units and used. For example, if we specify Y(_t) = 1cm, then y value will be correctly taken as 10mm.
- While parsing expressions, equation based curves convert each variable separately to model units and assume that the resulting expression is in model units.
- Equation based curves depend on the variable value library to correctly evaluate the units of expression.

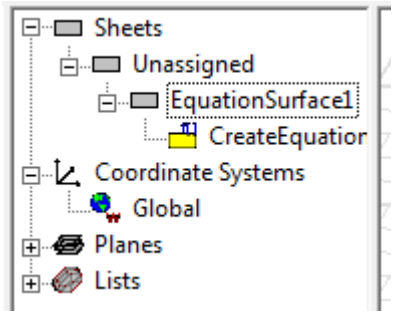
3. Specify start values for the **Start_u** and **Start_v** fields.

Select end values for **End_u** and **End_v** fields.



4. Click **OK** on the **Properties** dialog box.

   The surface is drawn in the Modeler window. The History tree contains the Equation Surface object and the command properties.



   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's attributes.

Related Topics

Setting the Reference Point

Assigning a Cross Section and Dimension to a Polyline

Drawing an Equation Based Curve

Functions, Operators and Quantities for the Edit Equation Dialog

Drawing a Sphere

Draw a sphere, a 3D circle, by selecting a center point and a radius. Spheres are drawn as true surfaces in the modeler.

1. Click **Draw**>**Sphere**  .
2. Select the center point of the sphere in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

3. Specify the radius by selecting a point on the sphere's circumference in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.
4. Click **OK**.

Note The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values.

For details about these commands see:

Technical Notes, "Surface Approximations" and related sections, Rectilinear Elements and Curvilinear Elements, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Drawing a Cylinder

Draw a cylinder by selecting a center point, radius, and height. Cylinders are drawn as true surfaces in the modeler.

1. Click **Draw**>**Cylinder**  .
2. Select the center point of the cylinder's base circle in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the radius by selecting a point on the base circle's circumference in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
4. Specify the cylinder's height by selecting a point on the axis perpendicular to the base circle's plane. Select the point by clicking the point or typing the coordinates in the **dX**, **dY**, and **dZ** boxes.

Note If you create a cylinder with a height of zero, HFSS draws a circular sheet object.

If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.

5. Click **OK**.

Note The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see:

Technical Notes, "Surface Approximations" and related sections, Rectilinear Elements and Curvilinear Elements, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Assigning a Cross Section and Dimension to a Polyline

Modifying Surface Approximation Settings

Creating Segmented Geometry

Drawing a Box

Draw a box by selecting two diagonally opposite corners of the base rectangle, then specifying the height.

1. Click **Draw**>**Box**  .
2. Select the first diagonal corner of the base rectangle in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

   To delete the selected point and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.
3. Select the second corner of the base rectangle in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the first diagonal corner in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
4. Specify the height of the box by selecting a point on the axis perpendicular to the base rectangle. Select the point by clicking the point or typing the coordinates in the **dX**, **dY**, and **dZ** boxes.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.
5. Click **OK**.

Related Topics

Setting the Reference Point

Drawing a Regular Polyhedron

In the modeler, regular polyhedrons are 3D objects with regular polygon faces; each face has three or more equal sides. Regular polyhedrons are useful for drawing faceted 3D objects.

1. Click **Draw**>**Regular Polyhedron**  .
2. Select the center point of the polyhedron in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Select the radius of the polyhedron, the distance from the center point to one of the polyhedron's vertices, in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
4. In the **Segment number** dialog box, enter the **Number of segments** in the polyhedron, and then click **OK**.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.
5. Click **OK**.

Note The radius is measured from the center point to a corner of the polygon, or the intersection of two edges. It is *not* measured from the center point to the midpoint of an edge.

If the **Automatically cover closed polyline** option is selected in the **Modeler Options** dialog box, the polygon will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

Related Topics

Setting the Reference Point

Covering Lines

Drawing a Cone

Draw a cone by selecting the center point and radius of the cone's base circle, then specifying the radius of the cone's top circle and the cone's height. Cones are drawn as true surfaces in the modeler.

1. Click **Draw**>**Cone**  .
2. Select the center point of the cone's base circle in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

3. Specify the radius of the cone's base circle by selecting a point on the base circle's circumference. Select the point in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
4. Specify the radius of the cone's top circle by selecting a point on its circumference. Select the point by clicking it or typing its coordinates in the **dX**, **dY**, and **dZ** boxes.

   To create an apex, select the same center point as the cone's base circle.
5. Specify the height of the cone by selecting a point on the axis perpendicular to the base circle's plane. Select the point by clicking the point or typing the coordinates in the **dX**, **dY**, and **dZ** boxes.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.
6. Click **OK**.

Note The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see:

Technical Notes, "Surface Approximations" and related sections, Rectilinear Elements and Curvilinear Elements, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Drawing a Torus

Draw a torus by selecting its center point, major radius, and minor radius. The modeler then sweeps a circle around a circular path. Toruses are drawn as true surfaces in the modeler.

1. Click **Draw>Torus**  .
2. Select the center point of the torus in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the major radius by selecting a point in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes,

where **d** is the distance from the previously selected point.

The major radius determines the diameter of the torus.

4. Specify the minor radius by selecting a point relative to the major radius point.

   The minor radius determines the diameter of the "donut hole".

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.

5. Click **OK**.

Note The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see:

Technical Notes, "Surface Approximations" and related sections, Rectilinear Elements and Curvilinear Elements, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Drawing a Helix

A helix is a 3D spiral object created by sweeping a 1D or 2D object along a vector. Sweeping a 1D object results in a hollow 3D object. Sweeping a 2D sheet object results in a 3D solid object.

1. Select the 1D or 2D object you want to sweep to form a helix.
2. Click **Draw**>**Helix**  .
3. Draw the vector you want to sweep the object along. The two points that describe the vector affect axis direction only and not the helix length. The helix length is determined when you enter the pitch and number of turns in the **Pitch** and **Turns** text boxes. The initial radius of the helix is determined by the axis position relative to the object being swept.
   a. Select the start point by clicking the point or typing its coordinates in the **X**, **Y**, and **Z** text boxes.
   b. Select the endpoint by clicking the point or typing its coordinates relative to the start point in the **dX**, **dY**, and **dZ** boxes.

      The **Helix** dialog box appears.
4. For **Turn Direction**, select **Right hand** if the turn direction is clockwise and **Left hand** if the turn direction is counter-clockwise.
5. In the **Pitch** text box, type the distance between each turn in the helix, and click a unit in the pull-down list.
6. In the **Turns** text box, type the number of complete revolutions the object will make along the vector.

7. In the **Radius Change per Turn** text box, type a number for the increase in the radius and select the units from the pull-down list.
8. After you set these values, the selected object is swept along the vector to form a helix. The original object you swept is deleted.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.
9. Click **OK**.

Note The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see:

Technical Notes, "Surface Approximations" and related sections, Rectilinear Elements and Curvilinear Elements, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Drawing a Segmented Helix with Polygon Cross-Section using a User-Defined Primitive

Drawing a Segmented Helix with Rectangular Cross-Section using a User Defined Primitive.

Drawing a Segmented Helix with Polygon Cross-Section Using a User Defined Primitive

ANSYS provides you with a DLL to define the parameters of a segmented helix with a polygon cross-section.

1. Click **Draw>User Defined Primitive>SegmentedHelix>PolygonHelix**.

   The **Create User Defined Part** dialog box appears. The **Parameters** tab permits you to edit the parameters. An **Info** tab contains information about the user defined primitive, its purpose, the company/author who created it, the date created and the version number.
2. Specify the values for the following parameters:

PolygonSegments	Number of segments in the polygon cross-section. Enter zero (0) for true circle
PolygonRadius	Radius of the polygon cross-section.
StartHelixRadius	The radius of a segmented helix is defined from the helix center of rotation to the center of the helix cross-section at segment transitions. The first and last segments of the helix are half segments. See this figure.

RadiusChange	The radius change per turn of the helix.
Pitch	Distance between helix turns.
Turns	The number of turns in the helix.
SegmentsPerTurn	The number of segments constructing each turn. Enter zero (0) for true curve.
RightHanded	Helix winding direction. Enter non-zero value for right-handed helix.

3. Click **OK**.

Related Topics

Setting the Reference Point

Creating a User Defined Primitive

Drawing a Segmented Helix with Rectangular Cross Section Using a User Defined Primitive

Drawing a Segmented Helix with Rectangular Cross-Section Using a User Defined Primitive

ANSYS provides you with a DLL to define the parameters of a segmented helix with a rectangular cross-section.

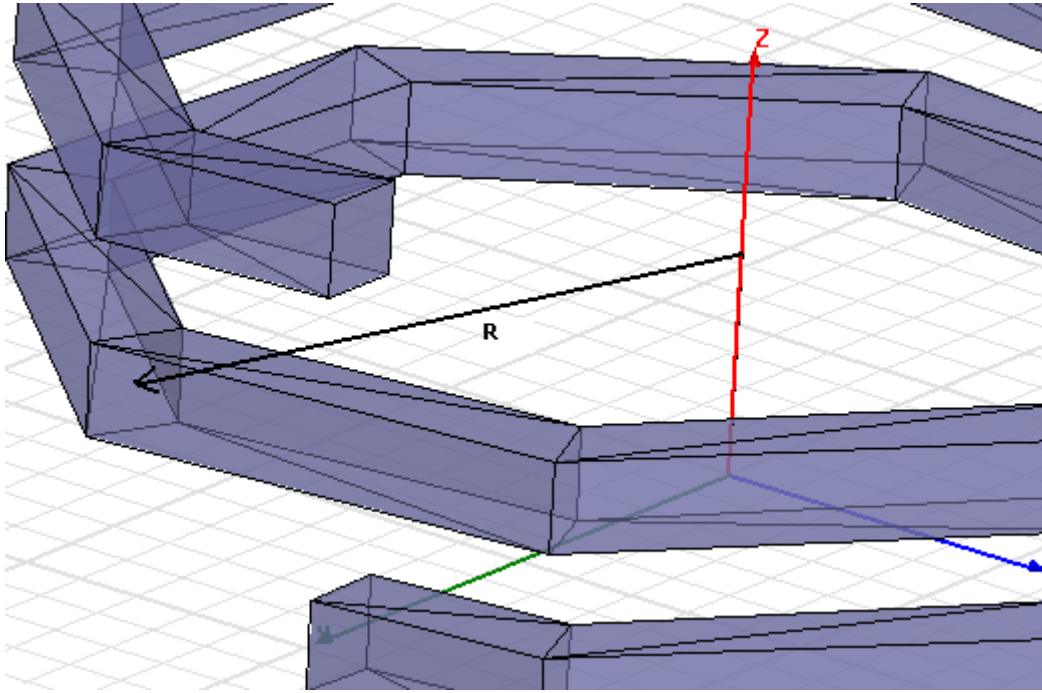
1. Click **Draw>User Defined Primitive>SegmentedHelix>RectHelix**.

   The **Create User Defined Part** dialog box appears. The **Parameters** tab permits you to edit the parameters. An **Info** tab contains information about the user defined primitive, its purpose, the company/author who created it, the date created and the version number.

2. Specify the values for the following parameters:

RectHeight	Height of rectangular cross-section.
RectWidth	Width of rectangular cross-section.
StartHelixRadius	The radius of a segmented helix is defined from the helix center of rotation to the center of the helix cross-section at segment transitions. The first and last segments of the helix are half segments. See this figure.
RadiusChange	The radius change per turn of the helix.
Pitch	Distance between helix turns.
Turns	The number of turns in the helix.
SegmentsPerTurn	The number of segments constructing each turn. Enter zero (0) for true curve.
RightHanded	Helix winding direction. Enter non-zero value for right-handed helix.

3. Click **OK**.



Related Topics

Setting the Reference Point

Creating a User Defined Primitive

Drawing a Segmented Helix with Polygon Cross-Section using a User-Defined Primitive

Drawing a Spiral

A spiral is a 2D or 3D spiral object created by sweeping an object around a vector. Sweeping a 1D object results in a 2D sheet object. Sweeping a 2D sheet object results in a 3D solid object.

1. Select the 1D or 2D object you want to sweep to form a spiral.
2. Click **Draw**>**Spiral**  .
3. Draw the vector you want to sweep the object around:
   a. Select the start point by clicking the point or typing its coordinates in the **X**, **Y**, and **Z** text boxes.
   b. Select the endpoint by clicking the point or typing its coordinates relative to the start point in the **dX**, **dY**, and **dZ** boxes.

      The **Spiral** dialog box appears.
4. Select **Right hand** if the turn direction is clockwise and **Left hand** if the turn direction is

counter-clockwise.

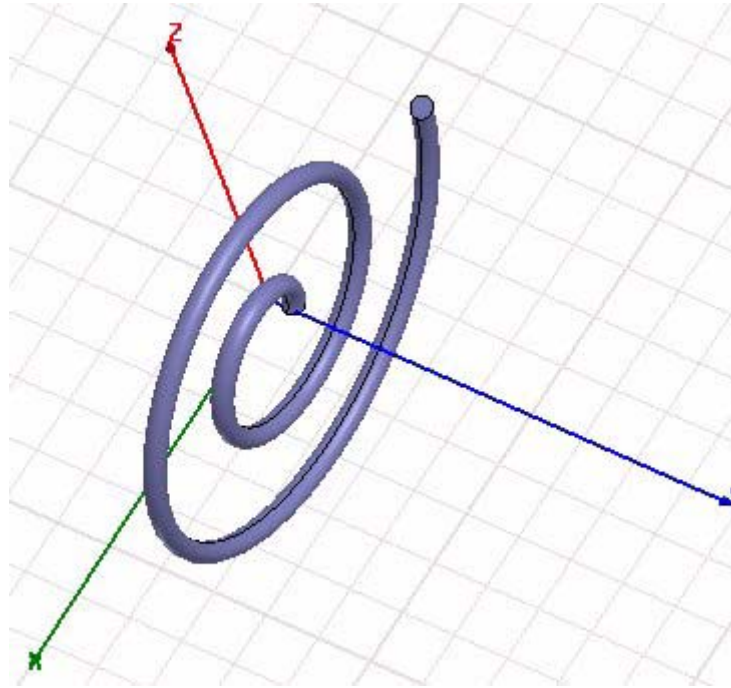
5. In the **Radius Change** text box, type the difference in radius between each turn of the spiral.

   The radius of the first turn is measured from the center point of the 1D or 2D object you are sweeping to the vector you drew.

6. Click a unit for the radius in the pull-down list.
7. In the **Turns** text box, type the number of complete revolutions the object will make around the vector.

   The selected object is swept around the vector to form a spiral. The original object you swept is deleted. If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties.

8. Click **OK**.



This 3D spiral was created from a 2D circle drawn at z = 0. The turn direction was right hand, the radius change was set at 2, and the number of turns was set at 2.

Note The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see:

Technical Notes, "Surface Approximations" and related sections, Rectilinear Elements and Curvilinear Elements, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Drawing a Spiral Using User Defined Primitives

Drawing a Spiral using User Defined Primitives

ANSYS provides you with a DLL and a Python script to define the parameters of a rectangular spiral.

1. Click **Draw>User Defined Primitive>Examples>RectangularSpiral (DLL)** or **Rectangular Spiral (Python)**.

   The **User Defined Primitive Operation** dialog box appears. The **Parameters** tab permits you to see and edit the parameters. An **Info** tab contains information about the user defined primitive, its purpose, the company/author who created it, the date created and the version number.

2. Specify the values for the following parameters:

Xpos	X location of the starting point.
Ypos	Y location of the starting point.
Dist	The separation distance between turns.
Turns	The number of complete revolutions the object will make around the vector
Width	The width of the spiral.
Thickness	The thickness/height of the spiral. If you specify the thickness as zero, the modeler draws a sheet object.

3. Click **OK**.

   This creates the primitive and displays the **Properties** dialog for the new object.

Hint To see newly created DLLs, click **Draw>User Defined Primitive>Update Menu**.

To see the primitives that you have created, click **Draw>User Defined Primitive>UserLib.**

Related Topics

Setting the Reference Point

Creating a User Defined Primitive

Drawing a Spiral

Drawing a Bondwire

A bondwire is a thin metal wire that connects a metal signal trace with a chip. Please see the topic Bondwires in the *Technical Notes* before drawing a bondwire.

1. Click **Draw**>**Bondwire**  .
2. Select the bond pad point in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Select the lead point by clicking the point or typing the coordinates in the text boxes in the status bar.

   The **Bondwires** dialog box appears.
4. In the **Type** list, click the JEDEC modeling standard shape you want the bondwire to have: **JEDEC 4-point**, **JEDEC 5-point**, or **Low**.

   The **Type** selection changes the dialog bondwire graphic, and shows options for that type.
5. Enter the number of facets in the bondwire in the **No. of Facets** text box.

   The minimum value is 3. The value describes the number of faces that make up the circumference of the bondwire.
6. In the diameter field, specify a diameter value and select the units from the pull-down menu.
7. Enter the height between the bond pad and the top of the loop in the **h1** text box. Include the height's unit of length.
8. The value in the **h2** text box is the height between the bond pad and the lead point. It was calculated by HFSS based on the lead point you selected. If you modify the value of *h2*, the lead point will be modified.

   Optionally, type a new value in the **h2** text box. Include the height's unit of length.
9. If you selected **JEDEC 5-point** or **Low** do the following:
   a. Type the angle between the horizontal plane and the wire at the bond pad point in the **alpha** text box.
   b. Type the angle between the horizontal plane and the wire at the lead point in the **beta** text box.
10. Click **OK**.

Related Topics

Setting the Reference Point

Technical Notes: Bondwires

Drawing a Point

Drawing a point object within the problem region enables you to plot fields or perform field computations at that point. Points are always considered non-model objects by the modeler.

1. Click **Draw**>**Point**  .
2. Select the point in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

   The point is listed under **Points** in the history tree.

Related Topics

Setting the Reference Point

Modifying Markers on Point Plots

Drawing Non-Model Objects

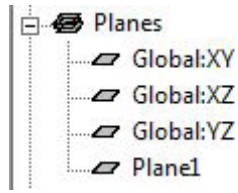
Drawing a Plane

A plane object is a cutplane through the problem region. You can plot fields or perform field computations on its surface. Planes are always considered non-model objects by the modeler.

1. Click **Draw**>**Plane**  .
2. Select the origin in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

   To delete the selected point and start over, press **ESC**.
3. Select a normal point in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the origin in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   The plane is created. Its center point is located at the origin you specified and oriented perpendicular to the normal point you specified. The new plane is listed under **Planes** in the history tree.



Note You only need to draw a plane that does not lie on a pre-defined xy, yz, and xz plane. Default planes are created on the xy, yz, and xz planes of the global coordinate system as well as any new coordinate system you create.

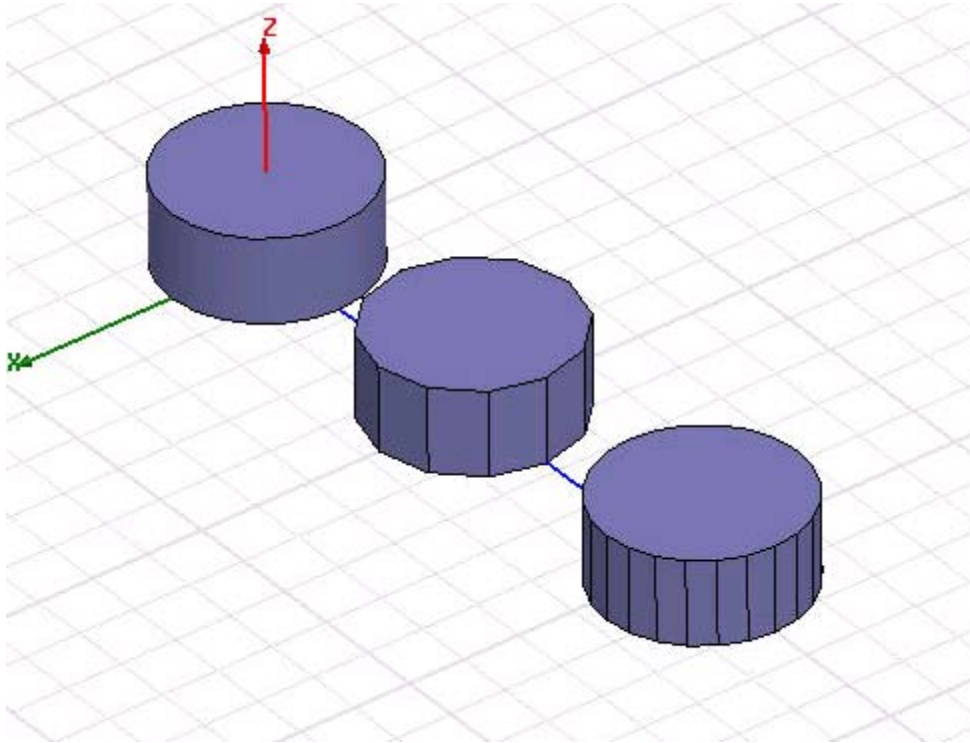
Related Topics

Drawing Non-Model Objects

Plotting Field Overlays

Creating Segmented Geometry

For some structures, you may want to create segmented as opposed to smooth (or True) surfaces. The figure below shows a comparison of a cylinder created with true surfaces and with segmented surfaces.



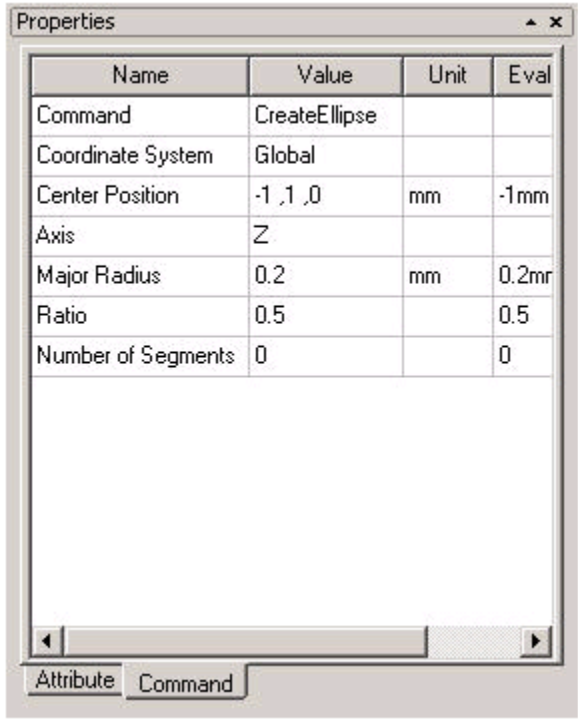
The following model objects can be created as segmented structures:

Circle, Ellipse, Cylinder	See Segmented Objects
Polyline, Arc, Line Segment	See Converting Polyline Segments

Segmented Objects

To create segmented circles, ellipses, and cylinders use the **Number of Segments** parameter on the Command Tab of the **Properties** dialog as shown below. To convert an object from true surface to segmented, do the following:

1. Select the **circle**, **ellipse**, or **cylinder** in the modeler window or in the history tree.
2. In the command tab of the properties window (shown docked below), change the **Number of Segments** to an integer value of three or greater and press **Enter**.



Values of 1 and 2 are not valid values for the circle, ellipse, or cylinder command and will cause an error.

Related Topics

Modifying Surface Approximation Settings

Creating Segmented Geometry

Drawing Non-Model Objects

If you want to create an object that does not affect the geometric model, define the object as *non model*. This ensures that the object is used for analysis only; it will not affect the solution process. After drawing the non-model object, assuming it lies in the problem region, you can use it in the

reporter as a place on which to plot field quantities. For example, draw a non-model line across the design, then (in the reporter) plot Mag_E on every point along that line.

You can assign output variables (default) and design variable as property values for non-model objects.

Following are examples of using non-model objects to analyze a solution:

- Draw a polyline along which to plot fields or perform field computations. Note that when you create a value versus distance plot, by default, the line will be divided into 100 equally spaced points. You can modify the number of points into which the line is divided in the **Edit Sweeps** dialog box.For more information, see Specifying Variable Values for a Sweep Definition.
- Draw a rectangle upon which to plot fields in the reporter or perform field computations.
- Draw a volume box to analyze fields in areas of the problem region that are not occupied by an object or that consist of parts of several objects.
- Draw a plane, which is always a non-model object.
- Draw a point object, which is always a non-model object, in order to plot fields in the Reporter or perform field computations at that point.

What do you want to do?

Switch to non-model drawing mode. Objects you draw in non-model mode will not be included in the solution process.

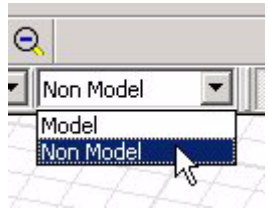
Modify an existing model object to be a non-model object.

Selecting Non-Model Drawing Mode

To switch to non-model drawing mode:

1. Click **Modeler**>**New Object Type**>**Non Model**.

   Alternatively, click **Non Model** on the drawing model pull-down list in the 3D Modeler Draw toolbar:



2. Draw the object.

Related Topics

Changing an Object to Non Model

Drawing Non-Model Objects

Changing an Object to Non Model

To modify an existing object to be a non-model object:

3. Select the object you want to modify.In the **Properties** dialog box, clear the **Model** option.

   The object will not be included in the solution process. If the object lies in the problem region, you can plot solution quantities on it.

Related Topics

Selecting Non-Model Drawing Mode

Drawing Non-Model Objects

Drawing a Region

To draw a region encompassing the objects in the current project:

1. Click **Draw>Region** or click the  icon on the tool bar.

   This displays the **Region** dialog. You can define the region padding as a percentage offset, relative offset, or absolute offset.

2. For the Padding data, click the Padding Data radio button as **Pad all directions similarly** or **Pad individual directions**.

   Selecting **Pad all directions similarly** leaves the Padding Percentage field as requiring a single value that affects all directions. In this case, you can specify the Padding type by selecting **Percentage Offset** or **Absolute Offset** from the drop down menu.

   Selecting **Pad individual directions** displays the Padding Percentage as a table of Positive and Negative X,Y, and Z coordinates, permitting you to specify padding for each direction. In this case, you can specify the Padding type by selecting **Percentage Offset** or **Absolute Offset** or **Absolute Position** from the drop down menu.

3. Specify the Padding values in the fields and select the units from the dropdown list.
4. If desired, click the check box to save the values as Default.
5. Click **OK** to close the dialog and create the region.

   The region is drawn, selected, and displayed in the History tree. It is created using the current coordinate system. The Properties dialog for the region has a Commands tab that shows the coordinate system and Padding values, and the Attributes tab includes properties for Name, Material (Default, vacuum), Solve inside, Orientation, Model, Color, Display Wireframe, and Transparency. You can edit all of these values.

If you try to create a region that does not contain all of the objects in your model, the modeler automatically expands the region to cover all objects. The region also updates automatically as your geometry changes.

Only one region can be created for a single project using the **Draw>Region** command. If you try to create a second region, the **Properties** window appears for the existing region, allowing you to change operation parameters and attributes.

Model Analysis

For some models it may be beneficial to remove unnecessary small entities and to fix object misalignments to avoid potential mesh issues. HFSS includes Model Analysis functions to help you evaluate models you have imported or created. Select **Modeler> Model Analysis** to see the menu options. Depending on the design and the current selection, some features may not be enabled. The menu includes the following commands.

- **Analyze Objects**
- **Analyze InterObject Misalignment**
- **Analyze Surface Mesh**
- **Show Analysis dialog**

Note Before running model analysis, you must remove all command history for the selected object by using the Purge History command. If you need to save the object history, save a separate copy.

1. After import, you typically perform validation check. This lets you focus on objects and object pairs that have errors and or warnings. The objects that fail should be analyzed by using the **Modeler>Model Analysis>Analyze Objects** menu item.
2. Select the objects and invoke **Modeler>Model Analysis>Analyze Objects**.

   This displays the **Analysis Options** dialog to allow you to specify settings for entity check level, and small feature detection.

   When you **OK** this dialog, the initial analysis executes and the **Model Analysis** dialog is displayed.
3. Choose the objects that have "Invalid Entities Found" and **Perform>Heal Objects**.

   In most cases, the objects will be healed and the errors fixed.
4. If errors still persist, choose the edges and faces and click on **Delete**.

   This will replace the selected face/edge object by a tolerant edge/vertex respectively. In some cases the replacement of the face/edge by tolerant edge/vertex will fail.

When models pass the initial validity checks, mesh generation could still fail. The following errors can be present in models: (See Error Detection.)

1. Non-manifold topology. These are non-manifold edges and vertices that are present in the model.
2. Object pair intersection. This detects whether pairs of objects intersect.
3. Small feature detection - small edge length, small face area and sliver face detection.
4. Mis-aligned entities detection - detects pairs of faces from objects that can be aligned to remove object intersections. This improves the probability of mesh success.
5. Mesh failure error display. This is available for single object, object pairs and last simulation

run (all objects in a model). Errors reported by the meshing module are reported to the user. Errors of type 3 and 4 should be resolved before you invoke the meshing for the model.

By default, the **Heal** command is automatically applied to imported objects.

Related Topics

Set Material Override

Analysis Options Dialog

Healing

Technical Notes: Removing Object Intersections

Healing Non-manifold Objects

Healing Options

Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Analysis Options Dialog

To perform analysis on an object according to specified features and tolerance values:

1. Select the object you want to analyze and click **Modeler> Model Analysis>Analyze Objects.**

   This displays the **Analysis Options** dialog, with the **Analysis Options** tab selected. Selecting **Modeler>Model Analysis>Heal** also displays this dialog. If, during **Modeler>Import...** you select **Heal Imported Objects** and **Manual** on the file browser dialog, you also see this dialog.
2. If desired, check the **Perform Entity Check Errors** checkbox.

   This enables the **Check Level** menu. The setting can be **Basic**, **Strict**, or **Comprehensive**. See Modeler Validation Settings for more explanation.
3. If desired, click the check boxes to enable and set the **Detect Feature** settings:
   - **Detect Holes**, and specify the Maximum Radius.
   - **Detect Chamfers**, and specify the Maximum Width.
   - **Detect Blends**, and specify the Maximum Radius.
4. If desired set the **Detect Small Entities** features and tolerance values.
   - **Small Edges**, length less than
   - **Small Faces**, area less than
   - **Sliver Faces**, which enables:

     Object Bounding Box Scale Factor

     Sliver Edge Width
5. Click the **Properties** tab to see a listing of the geometric properties of the selected object.
6. Clicking **OK** on this dialog displays the **Model Analysis** dialog which contains the results of the analysis.

Related Topics

Heal

Analysis Options Dialog

Model Analysis dialog.

Analyzing the Surface Mesh

To set the options to analyze the surface mesh:

1. Select an object of interest.

   This enables the Analyze Surface Mesh command in the menu.

2. Click **Modeler>Model Analysis>Analyze Surface Mesh**.
   The **Surface Mesh Analysis Options** dialog box appears. This dialog box allows you to set parameters to remove.

   - You can also open the **Surface Mesh Analysis Options** dialog box from the **Model Analysis** dialog box via the **Perform** pull-down menu on the **Objects** tab.

3. Select or clear the **Perform Object Pairs Analysis** check box. Selecting this option evaluates the mesh for all combinations of the selected objects.

4. Select or clear the **Ignore Objects Separated by greater than** check box, and enter a value in the text box. Selecting this option means that object pairs are disregarded from analysis if their separation is greater than the specified value.

5. Click **OK** to perform the analysis with the selected options.
   The **Model Analysis** dialog box appears, displaying the results of the analysis.

Related Topics

Heal

Model Analysis dialog.

Model Analysis dialog

This dialog contains results for all model analysis, including diagnostic information relating to mesh issues. To view the analysis options:

1. Click **Modeler> Model Analysis>Show Analysis Dialog**.
   A submenu appears.

2. Select one of the following from the submenu:

   - **Objects**
   - **Objects Misalignment**
   - **Surface Mesh**
   - **Last Simulation Mesh**

   The **Model Analysis** dialog box appears. (This dialog box also appears automatically after clicking **OK** in the **Analysis Options** dialog box.)

3. Select the **Auto zoom to selection** check box to automatically zoom to the item selected on the **Objects** tab.

4. Make the desired changes on each tab in the **Model Analysis** dialog box.
   - **Objects** tab
   - **Objects Misalignment** tab
   - **Surface Mesh (Single/Pairs)** tab
   - **Last Simulation Mesh** tab

Click **Close** to close the **Model Analysis** dialog box.**Related Topics**

Heal

Analysis Options

Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Objects Tab

All results relating to model analysis of specific objects are presented under the **Objects** tab.

1. The results table contains the following information.
   - Name - column listing the objects in the current design.
   - Last Analysis status - column giving the analysis status of the listed objects. Objects can have the following status:
     - Good - the object contains no invalid geometry entities given the tolerance values specified in the **Analysis Options** dialog.
     - Null Body - the object is non-existent.
     - Analysis not performed - the object was not selected for analysis.
     - Invalid entity errors - these are api_check_entity() errors and non-manifold errors which ANSYS EM recommends that you fix before meshing.
     - Small entity errors - small faces, sliver faces and small edges that are optionally detected based on the tolerance limits specified in the Analysis Options dialog.
2. Select any object name in the table which contains errors to display a set of radio buttons in the panel and a list of corresponding faces, edges and vertices.

Note **Auto Zoom to Selection** -- if this option is checked, the modeler automatically zooms to the item selected in the **Model Analysis** dialog box.

3. Select the face, edge or vertex entity from the list to view the error description in the **Description** field.
4. Select the **Delete** button if you want to remove a selected face or edge entity.
5. Select the **Perform** button to list the commands that you can execute on the selected objects in the Results table.
   - **Heal Objects** - repairs invalid geometry entities for the selected objects within the specified tolerance settings. The **Healing Analysis** dialog will appear.
   - **Analyze Objects** - evaluates the object status. Selecting this displays the Analysis Options

dialog.

- **Analyze Surface Mesh** - invokes a mesh for each selected object and reports analysis results under the **Surface Mesh (Single/Pairs)** tab. Selecting this option displays a dialog with radio buttons to select.
  - Perform Object Pairs Analysis - evaluates mesh for all combinations of the selected objects.
  - Ignore objects separated by greater than a specified value - object pairs are disregarded from analysis if their separation is greater than the specified value.
  - Click **OK** to perform the analysis with the selected options.
- **Analyze Interobject Misalignment** - determines any misalignments between two selected objects in the results table. The results are reported under the **Objects Misalignment** tab.

- **Display Healing Log** -- checking this causes the **Model Analysis** dialog to display a healing log which includes information about operations performed on an object during the healing process.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Object Misalignment Tab

The table in this panel displays results of an **Interobject Misalignment** analysis. It contains a list of Alignable Faces, described in a list of Object Sets, and corresponding Misaligned Faces.

All misaligned face pairs corresponding to the analyzed objects are listed in the table.

- **Align Faces** - select a face pair in the table and click the **Align Faces** button to align selected faces.
- **Clear All Analysis Data** - this button removes all information from the tables.
- **Auto Zoom to Selection** -- if this option is checked, the modeler automatically zooms to the item selected in the table.

After validation check is performed, the pairs of objects that intersect are chosen for analysis. Use the analysis results to find whether objects have faces that can be aligned.

Choose all the bodies that intersect with another body.

1. From the **Model Analysis** dialog choose perform/Analyze Interobject misalignment. Or you can run **Modeler>Model Analysis>Analyze Interobject Misalignment**.

   If the analysis finds object pairs that can be aligned, they will be displayed in the **Objects Misalignment** tab.

2. You can select individual or multiple rows and perform Align Faces. In some cases, face alignment will fail if the topology of the body changes by a large factor after alignment.
3. Identify individual bodies and body pairs that fail to mesh.
4. Perform Mesh analysis on individual objects and object pairs.
5. Review the reports and fix the errors.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Technical Notes: Removing Object Intersections

Healing Non-manifold Objects

Healing Options

Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Set Material Override for HFSS

Set Material Override for HFSS-IE

Select by intersection error message.

Surface Mesh (Single/Pairs) Tab

The panel displays the results of a surface mesh analysis.

1. You can display results for:
   - Individual Objects
   - Object Pairs

Note **Auto Zoom to Selection** -- if this option is checked, the modeler automatically zooms to the object or object pair selected.

2. The results table contains the following information:

- Object - column listing object name or a pair of object names.
- Last Analysis Status - column stating the meshing status of the object or object pair.
  - Mesh Success
  - Mesh Failure
- Error Type - this column gives the category of error that caused the mesh failure.
- Error Detail - provide specific geometry information regarding mesh error location.

Display options include:

- **Display Mesh Analysis** log checkbox -checking this displays further details concerning each error to be listed.

- **Auto Zoom to Selection** -- checking this causes the modeler to automatically zoom to objects or faces corresponding to the error.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Last Simulation Mesh Tab

The table in this panel lists all model errors as viewed by the mesher.

- Error Type - this column gives the category of error that caused the mesh failure.
- Error Detail - provide specific geometry information regarding mesh error location.

Display options include:

- **Display Mesh Analysis** log checkbox -checking this displays further details concerning each error to be listed.
- **Auto Zoom to Selection** -- checking this causes the modeler to automatically zoom to objects or faces corresponding to the error.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Design Settings for HFSS

The **HFSS>Design Settings** command displays a dialog with tabs for **Set Material Override,** for automatic **Lossy Dielectrics** and for **DC Extrapolation**. For Transient designs, the dialog contains a Transient tab.

Set Material Override Tab

The **Set Material Override** tab includes text note and a checkbox to **Allow metals to override dielectrics**. The purpose of this feature is to allow you to avoid doing explicit subtraction in the modeler. One example application is a via that passes through many dielectric layers--with the option turned on, the via does not have to be subtracted from the layers.

The **Set Material Override** option allows some intersections to be resolved automatically in the mesh. If metal intersects dielectric, the metal overrides the dielectric in the overlap region. (That is, the metal object is subtracted from the dielectric.) If objects with the same material overlap, the small object overrides the larger. (That is, the small object is subtracted from the larger.) All other intersections are treated as errors. Normally, the modeler considers any intersection between 3D objects to be an error.

To use this feature, check **Enable material override**.

In the meshing process, the dielectrics are locally overwritten by the metals in the intersecting region. That is, the part of the dielectric that is inside the metal is removed, and if the dielectric is completely inside, the whole object disappears.

Lossy Dielectrics Tab

This option applies frequency dependent lossy materials for the solver and post processor. The materials are not modified in the design. Instead, the Djordjevic-Sarkar model is applied before the material is passed to the solver or used for post processing.

- Automatically use causal materials. (Default, unchecked).

If the assigned material is already frequency dependent, automatic creation of frequency dependent lossy materials is ignored.

This feature addresses cases where you only have simple constant material properties available, but want to automatically apply a general-purpose frequency dependence to ensure causal solutions when solving frequency sweeps. Automatic causal material calculations are not performed under the following circumstances:

- If the solution type is eigenmode
- If the material permittivity or loss tangent is anisotropic
- If the material permittivity or loss tangent is spatially dependent
- If the material permittivity or loss tangent is frequency dependent
- If the material itself is not a lossy dielectric

Otherwise, when enabled the Djordjevic-Sarkar model is applied to all constant lossy dielectrics. These are defined as having a constant permittivity that is greater than one and a constant loss tangent that is greater than zero. The inputs to the Djordjevic-Sarkar model are the material's constant

permittivity and loss tangent, plus the standard default values of measurement frequency (1 GHz), DC conductivity (1e-12 S/m), and DC permittivity (none). The outputs from the Djordjevic-Sarkar model are the expressions for permittivity and conductivity. These expressions, plus zero loss tangent, are used in place of the material's constant properties.

When reading legacy designs (HFSS 12 and earlier), this feature is unchecked.

DC Extrapolation Tab

The **DC Extrapolation** tab option affects Interpolation Sweep setup. If you select Standard DC Extrapolation, the software computes values automatically, and the tab for DC Extrapolation does not appear on the Interpolating Sweep setup.

For Advanced DC Extrapolation, the Interpolating Sweep setup includes the DC Extrapolation tab, so you can set a Minimum solved frequency.

The **Design Settings** dialog also contains checkboxes to **Save As Default**.

Note Users must be careful: this setting changes the "ground rules" of the modeler, and may have unexpected results.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Materials

Setting the Temperature of Objects

Design Settings for HFSS Transient

DC Extrapolation options

Design Settings for HFSS-IE

The **HFSS-IE>Design Settings** command brings up a dialog with tabs for **Material Thresholds**, for **Set Material Override** and for **Lossy Dielectrics**.

The **Set Material Override** option allows some intersections to be resolved automatically in the mesh. If metal intersects dialectric, the metal overrides the dialectric in the overlap region. (That is, the metal object is subtracted from the dialectric.) If objects with the same material overlap, the small object overrides the larger. (That is, the small object is subtracted from the larger.) All other intersections are treated as errors. Normally, the modeler considers any intersection between 3D objects to be an error.

To use this feature, check **Enable material override**.

In the meshing process, the dielectrics are locally overwritten by the metals in the intersecting region. That is, the part of the dielectric that is inside the metal is removed, and if the dielectric is completely inside, the whole object disappears.

The purpose of this feature is to allow you to avoid doing explicit subtraction in the modeler. One example application is a via that passes through many dielectric layers--with the option turned on, the via does not have to be subtracted from the layers.

You can use the **Material Thresholds** tab to set the thresholds for:

- Perfect Conductors in Siemens per meter.

  Insulator Conductor in Siemens per meter

You can use the **Lossy Dielectrics** tab to automatically use casual materials. The options causes objects with constant material permittivity greater than one and dielectric loss tangent greater than zero to be treated as frequency dependent. Their actual permittivity and conductivity will be determined by the Djordjevic-Sarkar algorithm, and the loss tangent will be zero.

Automatic casual material calculations are not performed under the following circumstances:

- If the material permittivity or loss tangent is anisotropic
- If the material permittivity or loss tangent is spatially dependent
- If the material permittivity or loss tangent is frequency dependent
- If the material itself is not a lossy dielectric

The **Design Settings** dialog also contains checkboxes on each tab to **Save As Default**.

Note Users must be careful: these settings change the "ground rules" of the modeler, and may have unexpected results.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

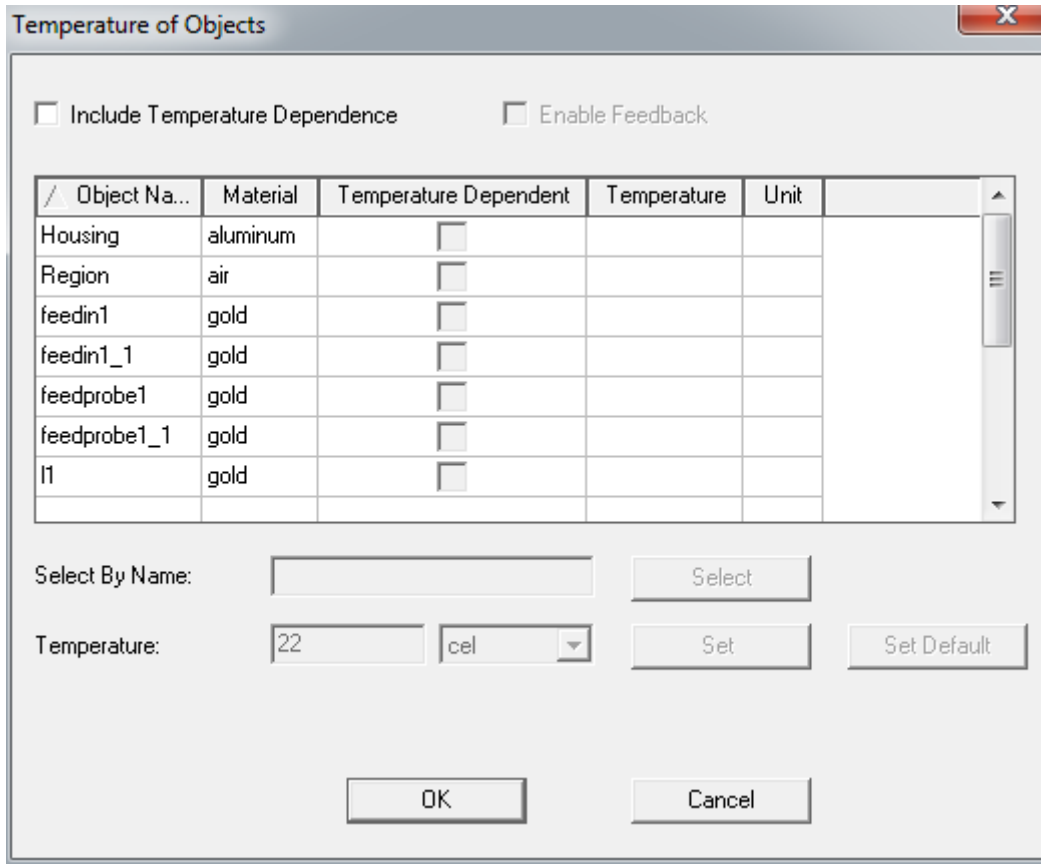
Analyze Surface Mesh
Healing
Materials
Setting the Temperature of Objects

Setting the Temperature of Objects

To set the temperature of objects and to Enable Feedback for use in Workbench:

1. Use the **HFSS** or **HFSS-IE>Set Object Temperature** command to display the **Temperature of Objects** dialog. You can also right-click on the Project and select from the shortcut menu.

   This dialog includes a table of the objects in the design. The first column lists the object name, then the material, then a checkbox to show whether that object has temperature dependent features, and then columns for Temperature and unit.



2. To enable the editing features, check **Include Temperature Dependence**.

   This makes the table objects selectable. The headers for the **Object Name** column and the **Material** column include sort direction arrows. You can invert the sort direction in each column by clicking the header. If the list is longer than the display, you can use a scroll bar on the right of the table.

   With the dialog enabled, you can use the **Select by name** field. Enter the name of the object you want and click **Select**.

Selected objects are highlighted. You can make multiple selections.

3. To set the temperature for a selected object or objects, type the value or an existing variable name in the text field.
4. Select the units from the drop down menu.
5. Click **Set** to apply the value to the selected objects, or click **Set Default** to make the specified values the default.

   If you click Set Default, the row for the selected objects display the Temperature value and units.
6. To edit the material for an object, when the dialog is enabled, you can click on the material for the row, and display a drop down menu listing the material and an **Edit**... button.

   Click the edit button to display the Materials dialog.
7. The Enable Feedback checkbox appears when HFSS is used with the ANSYS Workbench. If checked, you can perform thermal static and transient analysis based on an HFSS high frequency solution.
8. To close the dialog and accept the changes, click **OK**.

Related Topics

Resolving After ANSYS Workbench Thermal Feedback

ANSYS Workbench Integration Overview

Materials

Set Material Override

Specifying Thermal Modifiers

Heal

The **Heal** command provides a way to correct geometric violations and to remove specific kinds of small features. When models are imported, two types of errors can occur - geometry errors and topology errors. Geometry errors are errors in definition of the underlying geometry while topology errors are errors in how the underlying components like faces, edges and vertices are connected. ANSYS Electromagnetics recommends that these be fixed before you invoke mesh generation.

Imported objects which have only one operation on the history tree, can be healed. (Use the Purge History command to remove unwanted history operations before using **Heal**.)

Note If you need to save the object history, save a separate copy for that purpose before you heal the object.

Healing can be invoked in different ways.

- The menu command **Modeler>Model Healing>Heal** command applies to a selected object.
- Some formats permit healing during **Modeler>Import**. These are:

  3D Modeler file (*.sm3),

  SAT file (*.sat),

  STEP file (*.step,*. stp),

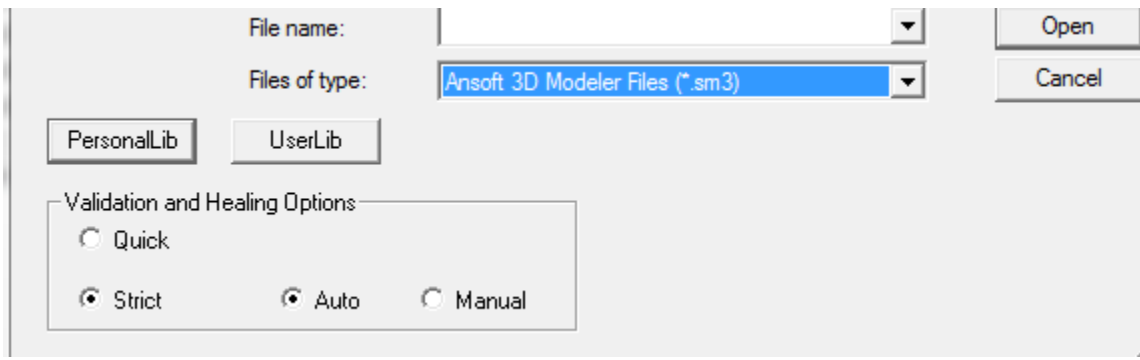
  IGES file (*.iges, *.igs),

  ProE files (*.prt, *.asm),

  CATIA (*.model, *.CATpart), and

  Parasolid file (*.x_t, and *.x_b).

  Selecting these formats enables a checkbox at the bottom of this window, "**Validation and Healing Options**."



  See this table for details. The Quick option allows you to switch off healing to speed up the import process.

- The **Model Analysis** dialog that appears after running **Modeler>Model Analysis>Analyze Objects**, or **Modeler>Model Analysis>Show Analysis** dialog includes a **Perform** action

menu with **Heal Objects** as a selection.

Any of these approaches leads to the same heal process.

Basic Steps in the Heal Process

There are several steps that are performed on selected objects.

1. Entity check, according to the **Analysis Options** settings.
2. Basic healing. This is done for all selected objects. Basic healing consists of fixing surface normals in the object and updating the orientation (to avoid having an object with negative volume).
3. Advanced healing. This is auto-heal. This is invoked on objects that require healing, that is, bodies that have errors, including have non-manifold errors.
4. Feature Removal. If you choose in the **Healing Options** to remove small holes, chamfers, blends, small edges, small faces and/or sliver faces, the actions are performed on all selected objects. There is no guarantee that small feature removal will be successful. (Also see Specifying the Model Resolution for defeaturing through the Auto Simplify and Model Resolution settings there.)

The above actions are performed on the selected objects. If you choose objects for healing which have not been analyzed, analysis is performed to determine its state (that is, whether it has invalid entities, small entities, and so forth). Invalid objects have all the above steps performed. Advanced healing is not performed on objects that do not require it.

While working on analyzing complex bodies, it is sometimes useful to examine faces, edges and vertices. In particular it is useful to find the connected faces for a face or edge or vertex, connected edges for a face/edge/vertex and connected vertices for a face/edge/vertex. The additional selection modes are available under **Edit->Select** and via the toolbar icons.

Related Topics

Align Faces

Remove Faces

Remove Edges

Technical Notes: Removing Object Intersections

Healing Non-manifold Objects

Healing Options

Specifying the Model Resolution

Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Validation and Healing Options for Import File

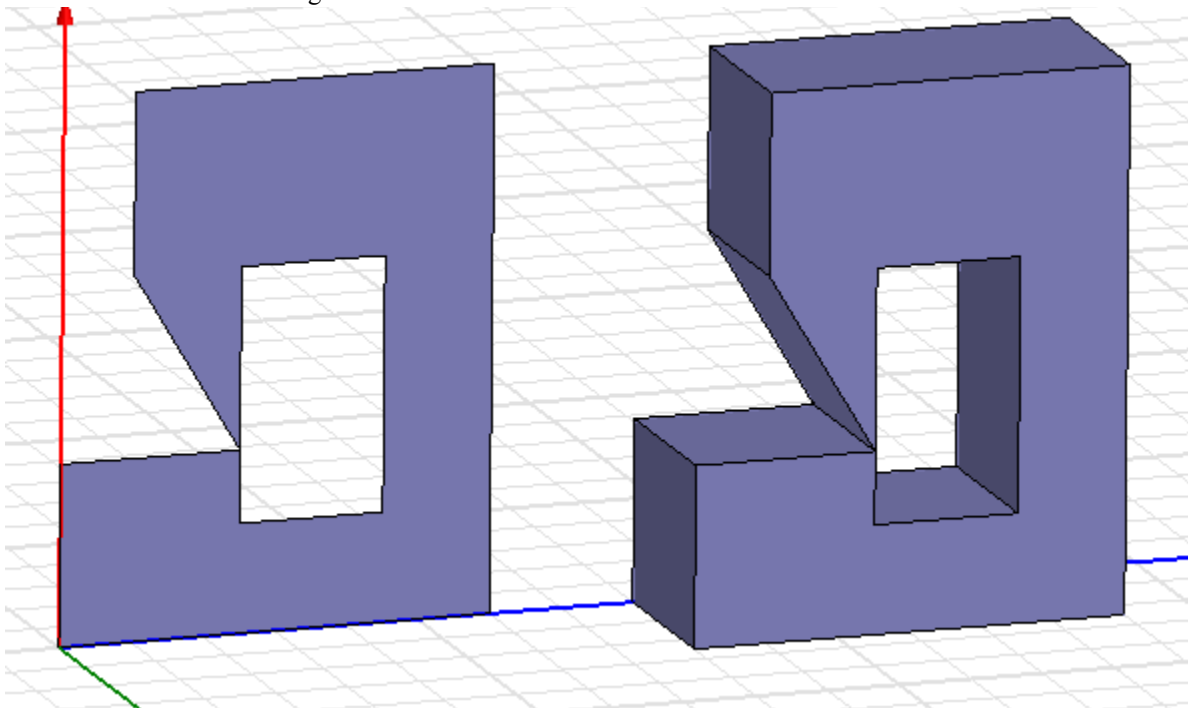
The Import File dialog contains shows different Validation and Healing options for the seven formats listed here. The Quick option is allows you to import these formats without healing.

Format	Options Available			
3D Modeler file (*.sm3),	Quick	Strict		
SAT file (*.sat),	Quick	Strict	Auto or manual	
STEP file (*.step,*. stp),	Quick	Strict	Auto or Manual	Stitch Tolerance
IGES file (*.iges, *.igs)	Quick	Strict	Auto or Manual	Stitch Tolerance
ProE files (*.prt, *.asm)	Quick	Strict	Auto or Manual	Import Free Surfaces. This imports such surfaces as well as parts.
CATIA (*.model, *.CATpart)	Quick	Strict	Auto or Manual	Stitch Tolerance
Parasolid file (*.x_t, and *.x_b)	Quick	Strict	Auto or Manual	

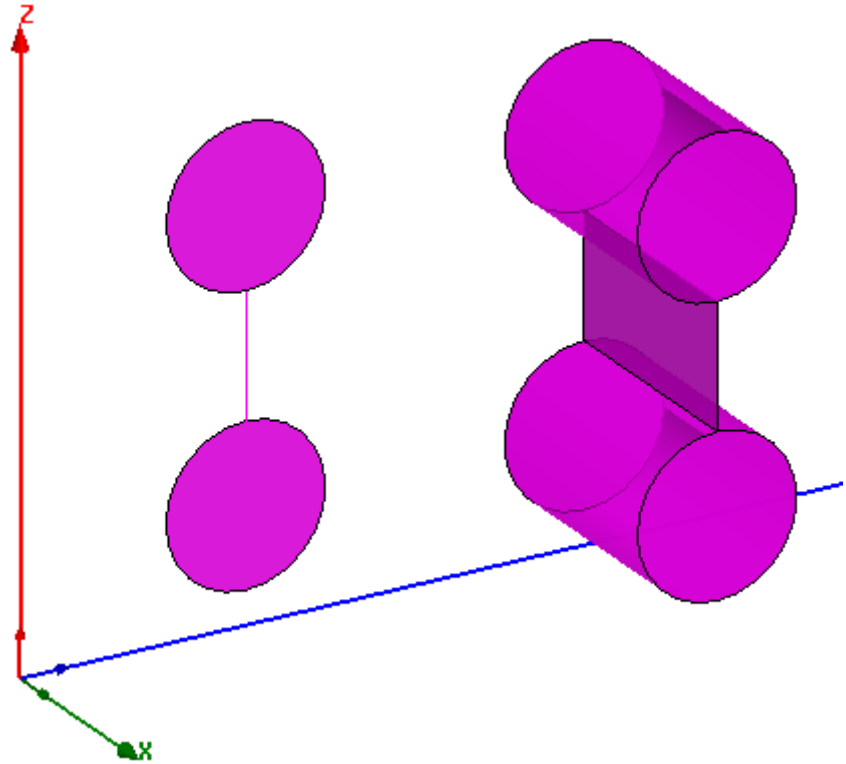
Healing Non-Manifold Objects

Non-manifold objects, in simple terms, are non-physical objects or objects that cannot be manufactured. For example, objects that intersect themselves (like the symbol for infinity in 2d) are clearly non-manifold. In addition objects that touch themselves may be non-manifold such as when a 2D

object touches itself at a vertex, or a 3D object touches itself at a point or edge. These cases are shown in the figure below.



Another type of non-manifold object has mixed dimensionality. For example, a pair of 2D objects connected by a 1D line segment, or a pair of 3D objects connected by a 2D sheet object. These cases are illustrated below.



The criteria for manufacturability is a simple manifestation of a complex mathematical concept that must be adhered to in the solid modeling system. When creating geometry, either directly, or through boolean operations, you should always consider whether or not the resulting operation will result in an object that could not be manufactured. If this is the case, then the object will cause an error in the modeler or in the meshing system.

To heal non-manifold objects:

1. Identify an edge that is non-manifold.
2. Select the connected faces.

   You can use the Face selection toolbar icons.
3. Create a face coordinate system on the planar face.
4. Create a small box to cover the non-manifold edge.
5. Either do a union or a subtraction to remove the faces that contain the non-manifold edge.

   The non-manifold edge is now removed. You may also remove or add a small portion of the model.

6. Do for all the non-manifold edges.

Related Topics

Healing

Technical Notes: Removing Object Intersections

Healing Options

Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Technical Notes: Mixed Dimensionality.

Setting the Healing Options

The **Healing Options** let you control how healing proceeds with respect to a variety of features and issues.

1. Click **Modeler>Model Healing>Heal** to open the **Healing Options** dialog. You can also open the **Healing Options** dialog from the **Model Analysis** dialog via the **Objects** tab drop down menu.

   The **Healing Options** dialog contains three tabs:

   - **Healing Options**
   - **Feature Removal Options**
   - **Properties**, which lists the geometric properties of the currently selected object.

2. Select the **Healing Options** tab on the **Healing Options** dialog to specify the following:

   - **Heal Type** as: **Auto Heal** (default), **Manual Heal**, or **No Heal**.

   Selecting **Manual Heal** enables the **Manual Heal Options**:

   - **Perform Tolerant Stitching** checkbox.

     This enables a field for the **Stitch Tolerance** value, and a checkbox to **Stop After First Error**.

   - **Perform Geometry Simplification**

     This enables fields for **Simplification Tolerance** and **Maximum Generated Radius** values.

     You can also select radio buttons to **Simplify Curves**, **Surfaces**, or **Both**.

   - **Tighten Gaps** settings.

     A checkbox to select **Perform Tighten Gaps**

     A field to specify **Tighten Gaps Within** a given value in mm.

3. Select the **Feature Removal Options** tab to specify the following:

   Here you can specify the following **Feature Removal Options**.

   - **Remove Holes** checkbox and **Maximum Radius** value.
   - **Remove Chamfers** checkbox and **Maximum Width** value.
   - **Remove Blends** checkbox and **Maximum Radius** value.

You can specify the following **Remove Small Entity Options**:

- **Small Edges, Length Less Than**, less than a specified value.
- **Small Faces Area e Less Than**, less than a specified area.
- **Sliver Face Width Less Than**, less than either:
  - **Object Bounding box Scale Factor**, less than a specified scale factor
  - **Sliver Edge Width**, less than a specified value.

  Sliver faces have a maximum distance among the long edges that is smaller than the specified tolerance and have at least one short edge and at most three long edges. A short edge has a length less than the specified tolerance. A long edge has a length greater than the specified tolerance. You can give the tolerance as a absolute value or a factor of the bounding box containing the face.

You can **Control Object Properties Change** according to the following settings:

- **Allowable Change in Surface Area** checkbox, and percent value.
- **Allowable Change in Volume** checkbox, and percent value.

4. Select the **Properties** tab to view the geometric properties of the currently selected object.
5. Click **OK** to apply the specified Healing options and to open the Analysis dialog.

Related Topics

Healing

Stitch Sheets

Technical Notes: Removing Object Intersections

Healing Non-manifold Objects

Specifying the Model Resolution

Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Stitch Sheets

Use the **Modeler>Model Healing>Stitch Sheets** command to stitch selected sheets.

1. Select two or more sheet objects.

   This enables the Stitch Sheets command on the **Modeler>Model Healing** submenu.
2. Click **Modeler>Model Healing>Stitch Sheets**

   This displays a Stitch dialog with a Maximum Stitch Tolerance field. The default value (auto) comes from the **Healing** dialog **Options** tab with Manual Healing selected. You may edit the value in the Stitch dialog or in the Healing Options.
3. Click OK.

   This closes the dialog and attempts to perform stitching on the selected sheets. If the sheets are separated beyond the stitch tolerance, stitching is not performed and a warning is issued.

Related Topics

Healing

Align Selected Faces

Use the **Modeler>Model Healing>Align Faces** command to align the selected faces. You can also use the toolbar icon when you have made an appropriate face selection 

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Remove Selected Faces

Use the **Modeler>Model Healing>Remove Faces** command to remove the selected faces. You can also use the toolbar icon when you have made an appropriate face selection 

If you find object-pair intersections that healing does not fix, or that can be fixed (by alignment), you can correct the problem by one of the following methods.

1. Use the **Remove Faces** command (**Modeler>Model Healing>Remove Faces**) or by performing Boolean subtract.
2. If overlap between objects is too large to be fixed by healing or by face alignment. Boolean intersect shows the common portion between the bodies. In this case, use a subtract operation to remove overlaps.

Related Topics

Align Faces

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Set Material Override

Remove Selected Edges

Use this **Modeler>Model Healing>Remove Edges** command to remove the selected edges. You can also use the toolbar icon when you have made an appropriate edge selection 

Related Topics

Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Technical Notes: Error Types

Technical Notes: Error Detection

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Creating a User Defined Primitive

The modeler allows you to generate user-defined primitives, primitives customized to suit any application. User-defined primitives are accessed using DLLs or IronPython scripts that you build and compile. When user defined primitives exist in your UserLib or Personal directory (given the paths specified in **Tools>General Options** Project Options tab), they appear in the **Draw>User Defined Primitives>** menu.. Newly created UDPs will appear after a restart, or **Draw>User Defined Primitive>Update Menu.**

For Python based primitives, see Creating User Defined Primitives and User Defined Models in Python Scripts

The modeler includes example C++ source and header files that can be used to generate DLLs. The files are located in the **`UserDefinedPrimitives/Examples`** subdirectory under the **`hfss13`** directory.

As an example, create the primitive **myUDP.dll** using Microsoft Visual C++ Developer Studio:

1. Create a directory to store all of the workspace information, call it `UDPDir`.
2. Use the sample workspace **RectangularSpiral.dsw** as a template:
   a. Copy **RectangularSpiral.dsw** and **RectangularSpiral.dsp** from the **`UserDefined-Primitives/Examples`** directory to this new directory.
   b. Make sure the new files have write permissions.
   c. Rename the files to **myUDP.dsw** and **myUDP.dsp** respectively.
   d. Open the **.dsw** and **.dsp** files in a text editor, and replace every occurrence of **RectangularSpiral** with **myDLL**.
   e. Save **myUDP.dsp** and **myUDP.dsw**.
3. In the **`UDPDir`** directory, create a **`Headers`** subdirectory.
4. Copy the **UserDefinedPrimitiveStructures.h** and **UserDefinedPrimitiveDLLInclude.h** files from the **`UserDefinedPrimitives/Headers`** directory.

Note The header files include information on the methods that are available for use in your source code. They must be included when you compile the DLL.

5. In the **`UDPDir`** directory, create a **`Source`** subdirectory.
6. Use the sample source file **RectangularSpiral.cpp** as a template:
   a. Copy **RectangularSpiral.cpp** from the **`UserDefinedPrimitives/Examples`** directory to this new directory.
   b. Make sure the new file has write permission.
   c. Rename the file to **myUDP.cpp**.

   The resulting directory structure will resemble the following:

UDPDir/

myUDP.dsw

myUDP.dsp

Headers/

`UserDefinedPrimitiveDLLInclude.h`

`UserDefinedPrimitiveStructures.h`

Sources/

myUDP.cpp

7. Open **myUDP.dsw** using Microsoft Visual C++ Developer Studio, and edit the source code to create your desired primitive. You may also add additional headers and source files as appropriate.

   The UDP dll contains a data structure called UDPPrimitiveTypeInfo. This contains information about the udp, its purpose, company/author who created it, date created and the version number. When you select a primitive from your library, you see the **Create Primitive** dialog with a **Parameters** tab for setting the parameters, and an **Info** tab with the information from this data structure.
8. Build **myUDP.dll** using the **Win32 Release** configuration.
9. Copy the resulting file **myUDP.dll** to the **`hfss15/userlib/UserDefinedPrimitives`** directory or the **hfss15/personallib/UserDefinedPrimitives** directory.
10. To view your primitives, click **Draw>User Defined Primitive>Update Menu** and then click **Draw>User Defined Primitive>**.

Note

On UNIX, you may use the same example directory structure, source, and header files to build and compile a shared library using C++. The resulting shared library will have a **.so** extension for Solaris and a **.sl** extension for HP-UX, and needs to be placed in the same **`hfss14/userlib/UserDefinedPrimitives`** directory.

As with the Windows DLL, the compiled library will work only on the operating system on which it was built.

Related Topics

Drawing a Spiral Using User Defined Primitives

Drawing a Segmented Helix with Polygon Cross-Section Using a User-Defined Primitive

Drawing a Segmented Helix with Rectangular Cross Section Using a User Defined Primitive

Creating User Defined Primitives and User Defined Models in Python Scripts

User Customization through User Defined Primitives (UDPs)

User Defined Primitives (UDPs) allow users to add customized geometric modeling commands to the HFSS Desktop. UDPs are compiled libraries that can be added to the desktop interface and shared between users with common modeling needs.

To create a UDP, see Creating a User Defined Primitive for requirements and the procedure for building a proper DLL.

In order to share UDPs between users, an existing DLL may be copied into the **userlib>User Defined Primitives** subdirectory which can be given the paths specified in **Tools>General Options** Project Options tab. Placing an appropriately constructed DLL in this subdirectory and executing **Draw>User Defined Primitives>Update Menu** adds a new menu item in the **Draw>User Defined Primitives** menu to allow access to the UDP.

Related Topics

Drawing a Spiral Using User Defined Primitives

Drawing a Segmented Helix with Polygon Cross-Section Using a User-Defined Primitive

Drawing a Segmented Helix with Rectangular Cross Section Using a User Defined Primitive

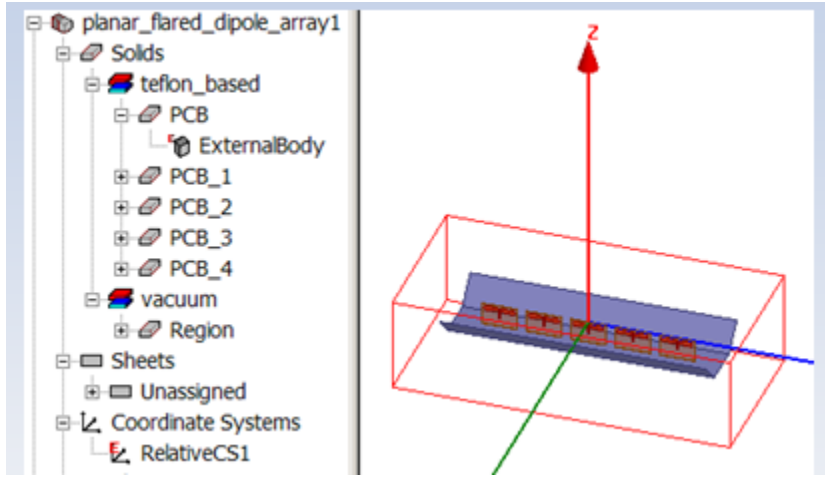
Creating User Defined Primitives and User Defined Models in Python Scripts

User Defined Model (UDM)

A User Defined Model (UDM) is collection of externally defined parts imported into an ANSYS EM 3D Modeler or created using a C or Python script.

- UDM includes part attributes (like name, color etc) and material assignment
- UDM can also have external coordinate systems and corresponding planes
- UDM parts can be parameterized and manipulated in ANSYS EM modeler just like any other

part



UDM can either represent Static geometry models or

- Dynamic links to models of external geometry editors
- Used for supporting CAD integration in WorkBench

  UDM uses same plugin technology as User Defined Part (UDP)

See the following sections:

UDM compared to User Defined Primitives

Insert UDM Command on Draw Menu

UDM Properties

Library of Models for CAD Integration

ANSYS EM to ANSYS Geometry Transfer

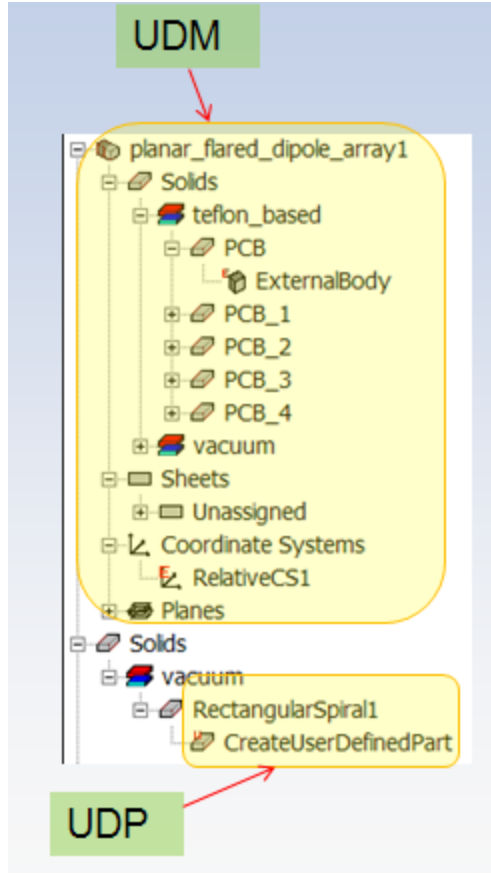
Material Assignment Transfer

Geometry Transfer through ANSYS DesignModeler (DM)

Creating User Defined Primitives and User Defined Models in Python Scripts

UDM compared to User Defined Primitives

User Defined Models (UDM) resemble User Defined Primitives (UDP):



- ANSYS EM products can be extended by users through new UDMs
- UDM plugins are discovered by searching standard directory paths
- Plugins for static UDM can build model using 'callback interfaces' (like create-box, create-cylinder, subtract etc) similar to UDP.
- UDMs run inside ANSYS EM applications
- UDMs provide geometry, topology, persistence and parameters

In contrast to UDP:

- UDM provides multiple Parts/CS/etc.

  UDP provides primitive operation only for a single Part
- UDM provides part attributes and material assignment

  UDP does not define part attributes or material

UDM Properties have four tabs - Definitions, Parameters, Options and Info

- Definition tab has:
  UDM name
  Coordinate system used to position UDM
  May have external reference to file
- Info tab has:
  UDM dll or .py name, dll or .py location, version etc
- Option tab:
  may have options if any

Related Topics

User Defined Model (UDM) for ANSYS WB Integration

Insert UDM Command on Draw Menu

UDM Properties

Library of Models for CAD Integration

ANSYS to ANSYS EM Geometry Transfer

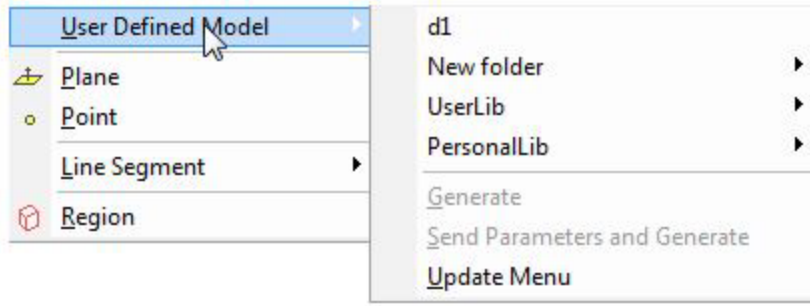
CAD Integration Material Assignment Transfer

Geometry Transfer through ANSYS DesignModeler (DM)

Creating User Defined Primitives and User Defined Models in Python Scripts

Insert UDM Command on Draw Menu

To insert a UDM into a design, use the User Defined Model command on the Draw menu for the Modeler window.



Related Topics

User Defined Model (UDM)

UDM compared to User Defined Primitives

UDM Properties

Library of Models for CAD Integration

ANSYS EM to ANSYS Geometry Transfer

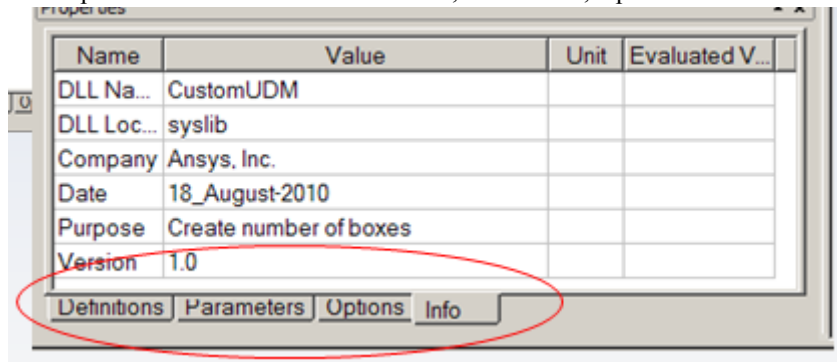
Material Assignment Transfer

Geometry Transfer through ANSYS DesignModeler (DM)

Creating User Defined Primitives and User Defined Models in Python Scripts

UDM Properties

UDM Properties have four tabs - Definitions, Parameters, Options and Info.



Definition tab has

- UDM name
- Coordinate system used to position UDM
- May have external reference to file

Info tab has:

- UDM dll or .py name, dll or .py location, version etc

Option tab:

- may have options if any

UDM Parameters

UDM Part Edits

Related Topics

User Defined Model (UDM)

UDM compared to User Defined Primitives

Insert UDM Command on Draw Menu

Library of Models for CAD Integration

ANSYS EM to ANSYS Geometry Transfer

Material Assignment Transfer

Geometry Transfer through ANSYS DesignModeler (DM)

Creating User Defined Primitives and User Defined Models in Python Scripts

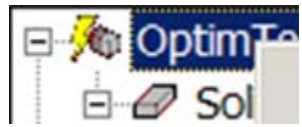
UDM Parameters

UDM geometry can be manipulated through its parameters.

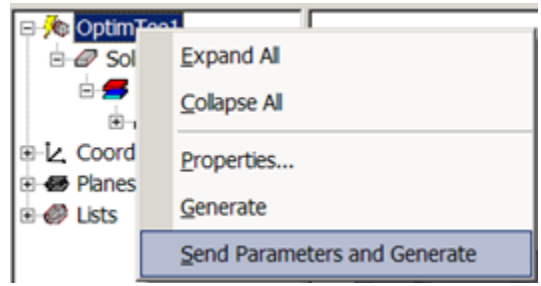
- Can be mapped to design or project variable for animation, parametric analysis.
- IDs are persisted (allowing to retain boundaries) during parameter edits.

UDM geometry in HFSS invoked through Workbench is not 'dynamic updated' upon parameter edits.

- In such cases, UDM shows a lightening bolt icon by the model name when parameters are edited.



- You must run the **Send Parameters and Generate** command to synchronize parameters with geometry



Related Topics

UDM Properties

UDM Part Edits

UDM Part Edits

Several modeling operations are allowed on UDM parts

- Operations will be part of history tree and retained during model refresh

The following operations are not allowed

- Non history tree operations like healing, defeature.
- Operations which use UDM parts as tool, such as sweep or boolean (but allowed when clone tool option is selected)

Following part attributes can be modified for UDM parts

- Model/Non Model flag
- Solve Inside flag
- Part orientation

Related Topics

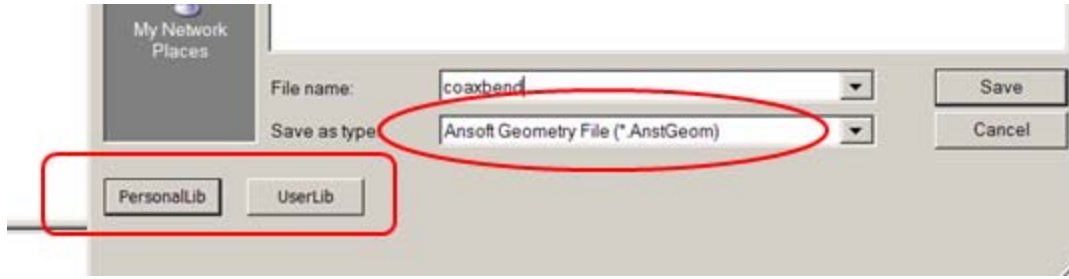
UDM Properties

UDM Parameters

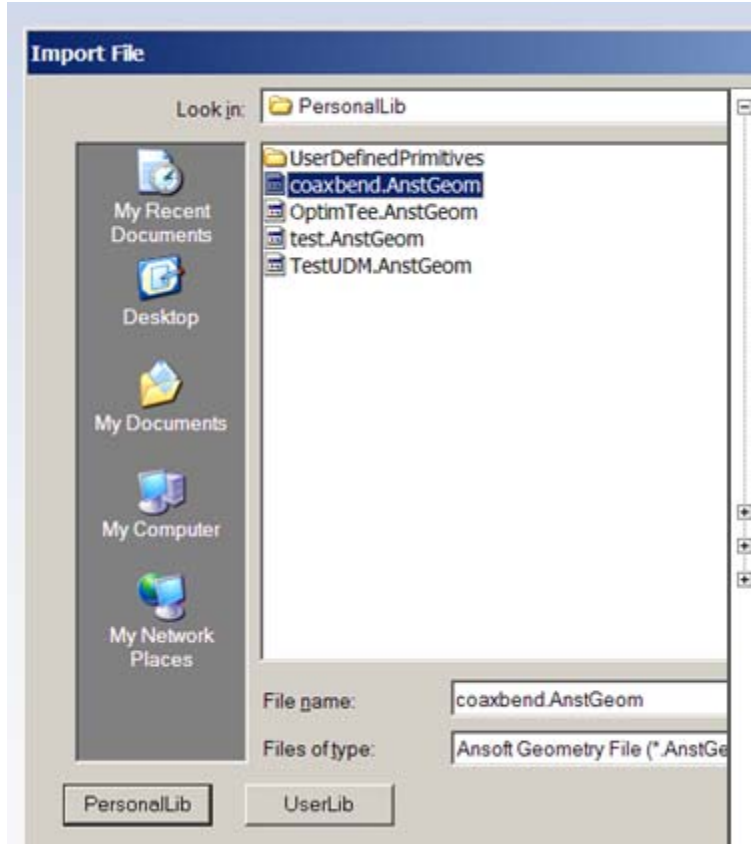
Library of Models for CAD Integration

UDM technology allows a library of models

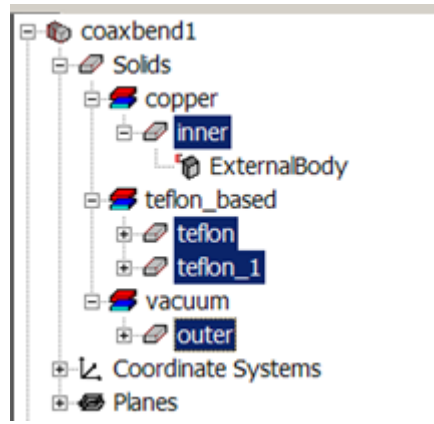
- Any ANSYS EM model can be exported as 'Ansoft Geometry File'



- An Ansoft Geometry File can be imported back as a UDM.



- Any Design/project variables associated with model are brought in as UDM parameters.



Geometry computation for UDM (and also UDP) can be specified in the Modeler options as either done on.

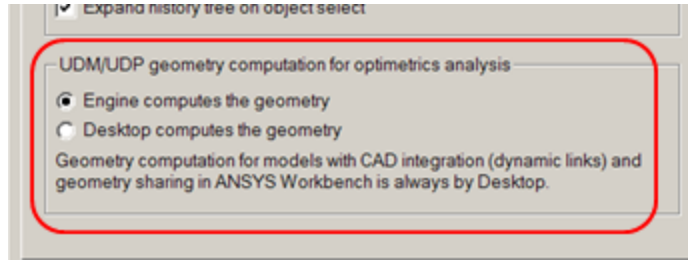
- Engine side (default):

  Requires deployment of UDM on each node

- Desktop side:

  UDM need not be deployed on each engine

  Desktop will be busy during parametric analysis



Related Topics

User Defined Model (UDM)

UDM compared to User Defined Primitives

Insert UDM Command on Draw Menu

UDM Properties

ANSYS EM to ANSYS Geometry Transfer

Material Assignment Transfer

Geometry Transfer through ANSYS DesignModeler (DM)

3D Component Library

The 3D Modeler lets you access predefined 3D component libraries and to create 3D Components with:

- Geometry – objects and coordinate systems
- Material assignment and definition
- Geometry parameters
- Boundaries, excitations and mesh operations

This lets you easily share and reuse components. You can add components to a library and share components with other users. Once a component inserted into the target design, you can directly manipulate whole components using Arrange operations such as **Move**, **Rotate** and **Mirror** or Duplicate operations like **Duplicate Along Line**, **Duplicate Around Axis** and **Duplicate Mirror.**

The documentation includes the following sections:

Creating a 3D Component from an Existing Model

Antenna Library

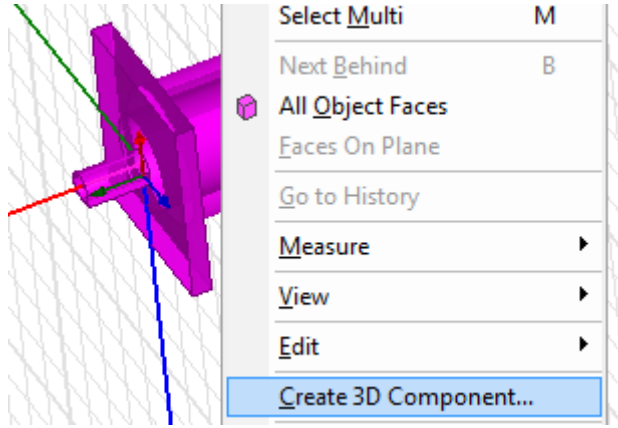
Rectangular Waveguide Library

Inserting a Component into a Design

Editing a Component

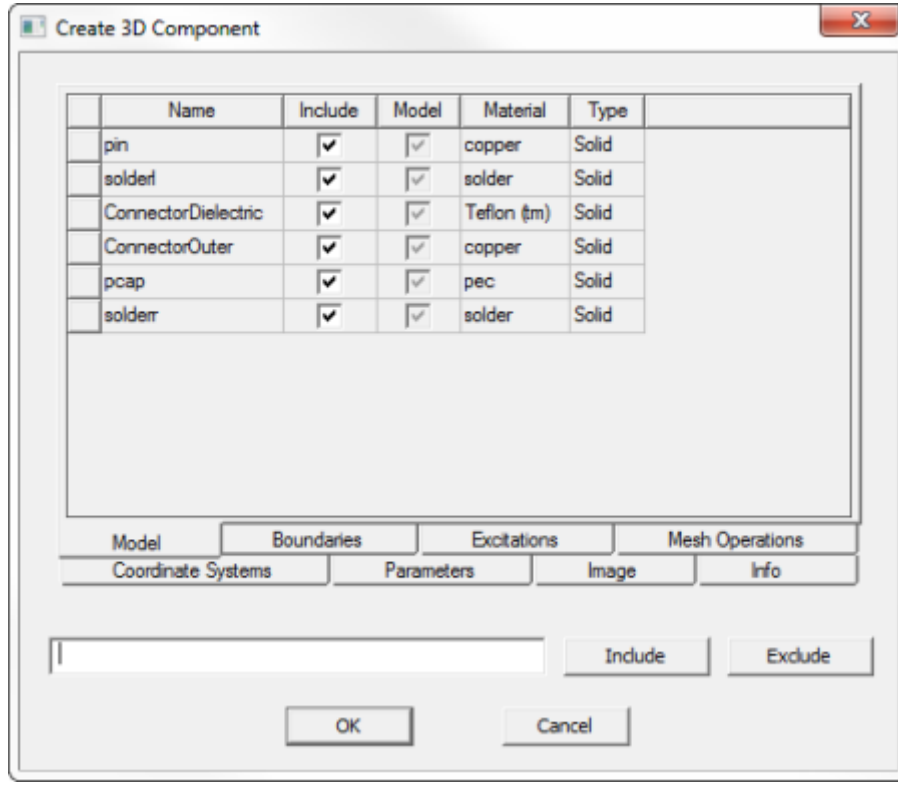
Creating a 3D Component from an Existing Model

To create a 3D component, select an existing object or model, right-click, and select **Create Component**. You can also click **Draw>3D Component Library>Create 3D Component**.



This opens the **Create Component** dialog. The **Model** tab lists all of objects selected before you invoked **Create Component**.

.



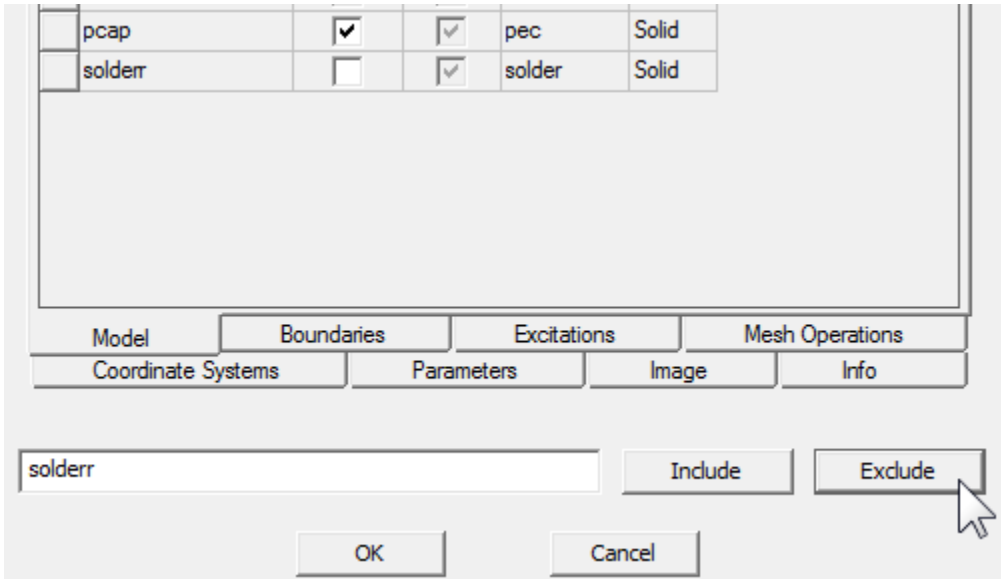
The tabs let you view the component features organized according to geometry, design data, and settings:

- Model tab
- Boundaries tab
- Excitations tab
- Mesh Operations tab
- Coordinate Systems tab
- Parameters tab
- Image tab
- Info tab

Note 3D components selected from the SysLib libraries include tabs only for Parameters, Image, and Info.

Include or Exclude Feature from Component

You can view the various tabs and select which features to **Include** or **Exclude** in creating the component. You can use the check boxes in the Include column for this purpose. You can also use the text field. Type a feature Name which you can then use the command buttons to Include or Exclude.



Object selection for **Include** or **Exclude** impacts selection in other tabs. For example, if you exclude an object from the component creation, the design data, parameters, and coordinate systems corresponding to that object are also excluded.

Sorting Columns Using the Headers

The features listed for each tab include columns for the Name, check boxes for whether to include, and properties. You can click on the column headers to sort by Name, Model, or other feature listed for each tab. Click the column header to selects the column to sort. Click again to invert the column.

You can sort lists by using all columns except Include.

Notes on Design Data

- DC thickness on selected objects is always included in components. It is not listed in **Boundaries** tab.
- The **Create 3D Component** dialog does not list design data without an assignment (for example, a far/near field incident wave in HFSS, or a winding in Maxwell).
- The parent of any included boundary/excitation is included, as long as the parent does not require assignment (for example, a winding in Maxwell).
- Design settings like material overrides are not included.

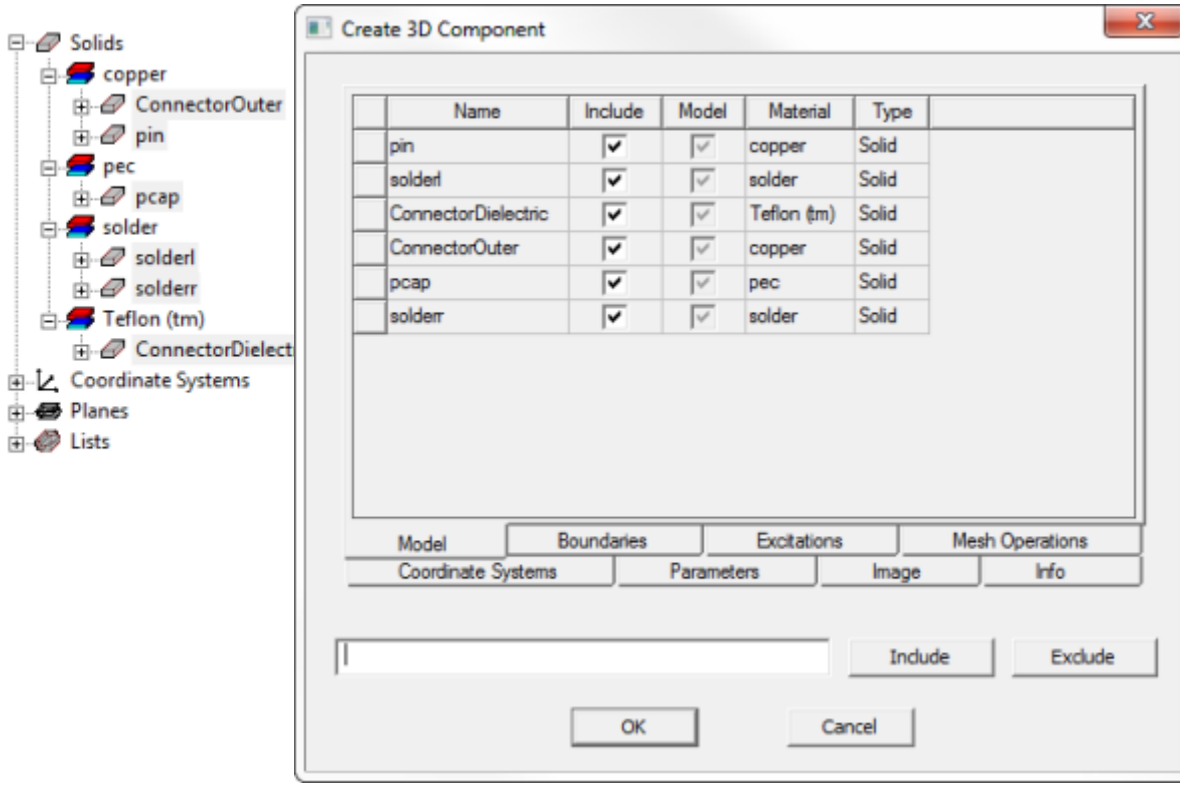
Once you have made the Include and Exclude settings, and have specified the Image and Info you click **OK** to Save 3D Component File.

Create 3D Component: Model Tab

The **Model** tab lists all of objects selected before you invoked **Create 3D Component**. These correspond to the selected objects listed in the history tree.

- Use the check boxes or text field to Include or Exclude objects for the component.
- Click OK to validate objects for inclusion. You cannot include on objects which are excluded from component.
- Columns Material and Type are read only.

.



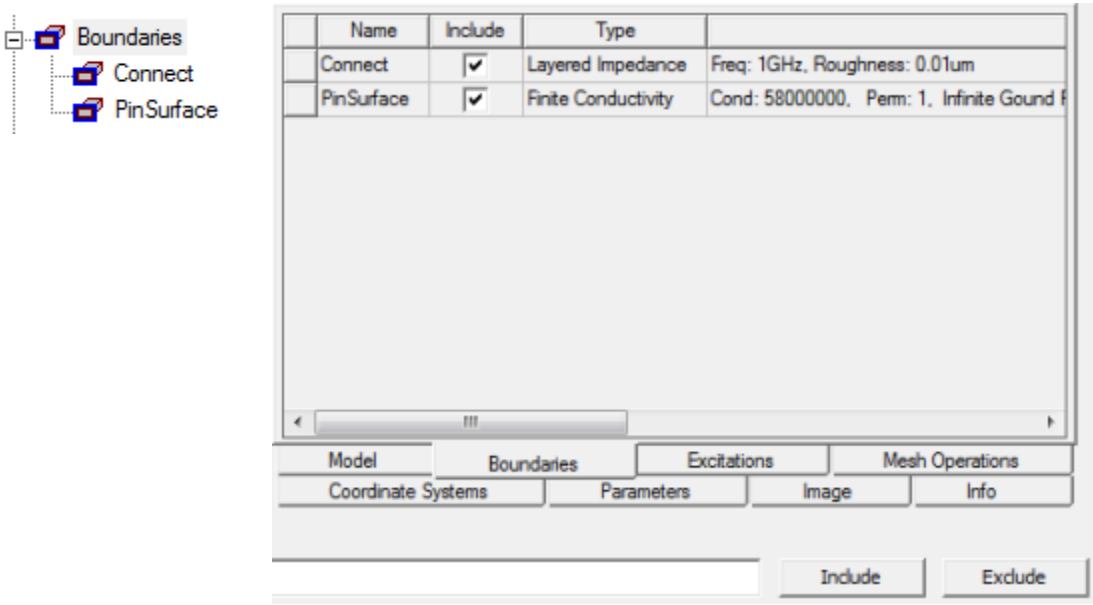
Create Component Boundaries Tab

The list is populated with all the boundaries on selected objects.

- Use the check boxes or text field to Include or Exclude boundaries for the component.
- Click OK to validate boundaries for inclusion. You cannot include boundaries on objects

which are excluded from component.

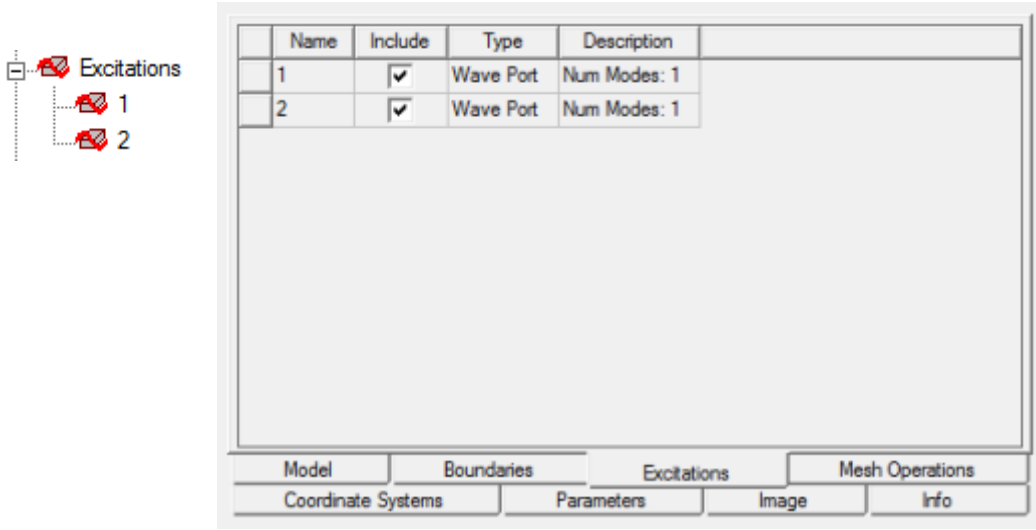
- Columns Type and Description are read only.



Create Component, Excitations Tab

The list is populated with all the excitations on selected objects.

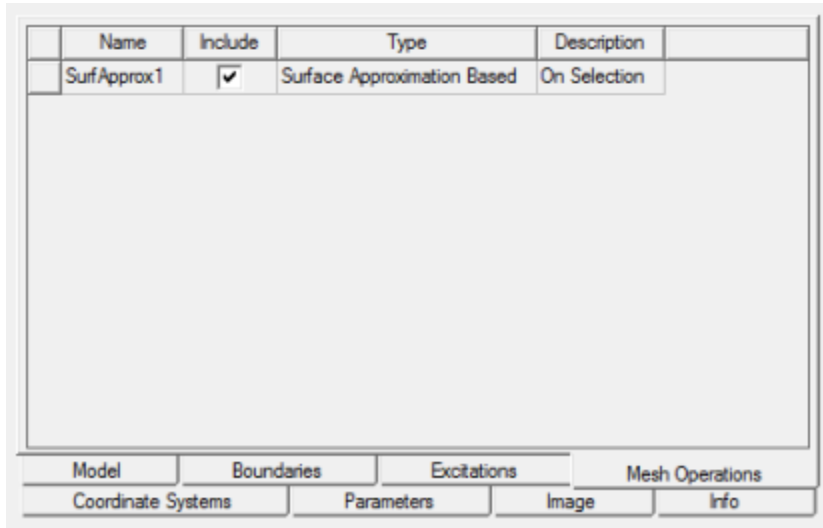
- Use the checkbox to Include or Exclude excitations in the component. You can also use the edit box and **Include**/**Exclude** buttons to filter excitations.
- Click **OK** to validate excitations.
- Columns for Type and Description are read only.



Create Component Mesh Operations Tab

The list is populated with all the mesh operations on selected objects.

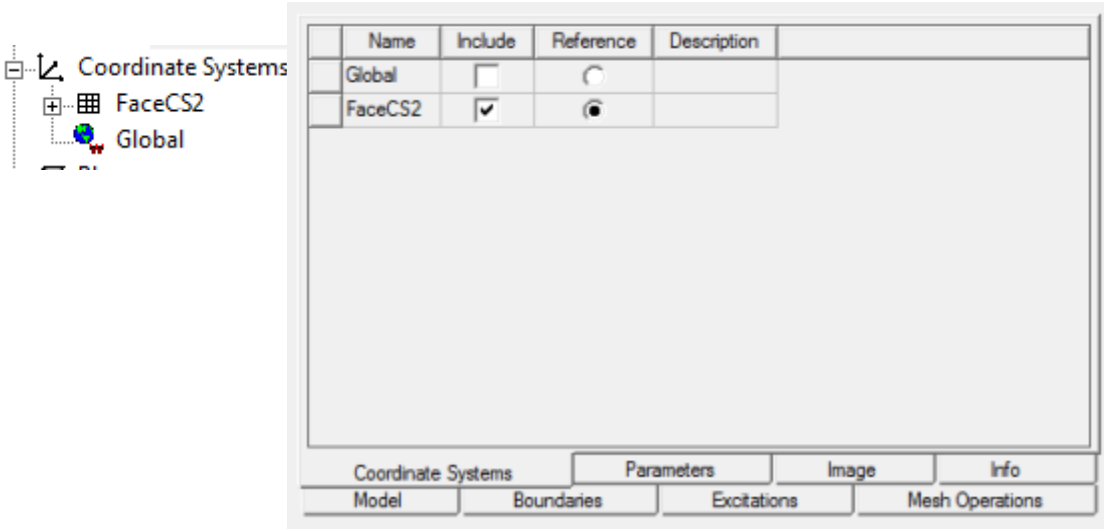
- Use the checkbox to Include or Exclude mesh operations. You can also use the edit box and **Include**/**Exclude** buttons to filter mesh operations.
- Click OK to validate mesh operations.
- Columns for Type and Description are read only



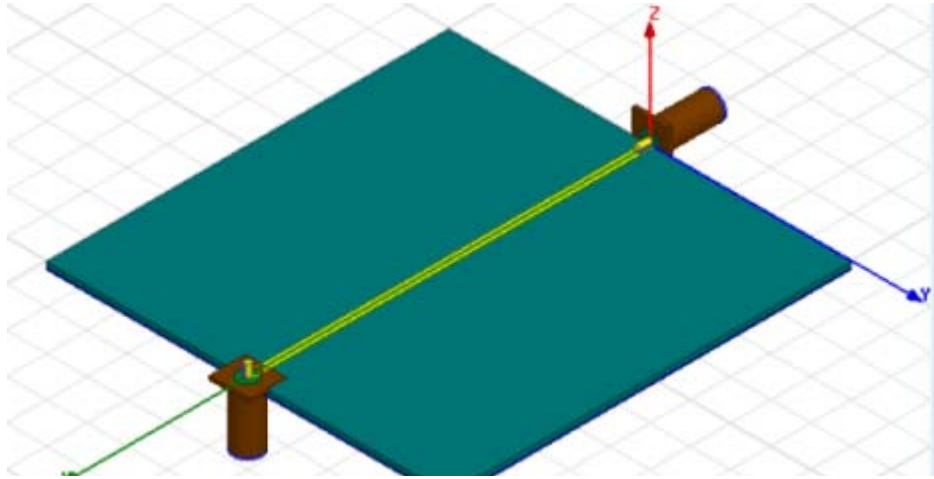
Create Component Coordinate Systems Tab

The list is populated with all the Coordinate Systems on selected objects. By default, only the Coordinate Systems used to define the object orientations are included.

- Use the checkbox to Include or Exclude coordinate systems. You can also use the text field and **Include**/**Exclude** buttons. Global CS cannot be included in the component.
- You can select any of the CS as a component reference.
- By default, the current working CS is the component reference CS.
- Reference CS must be included in the model (except for global).



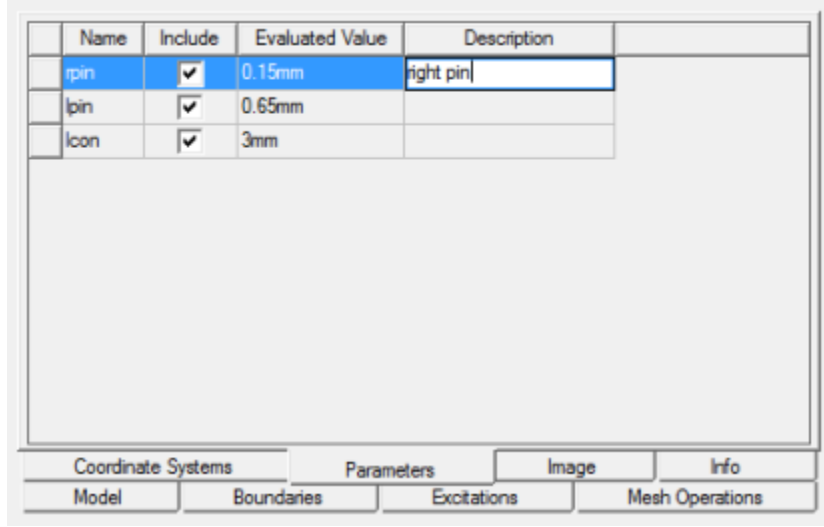
The Reference system that you specify affects the orientation of the component upon insertion. You may want to create different versions of your component based on the target orientation in order to reduce the need for rotations, for example, after insertion.



Create Component Parameters Tab

Component properties can be parameterized by assigning variables. The parameters list is populated with all the variables used by objects (and Coordinate Systems) included in the component.

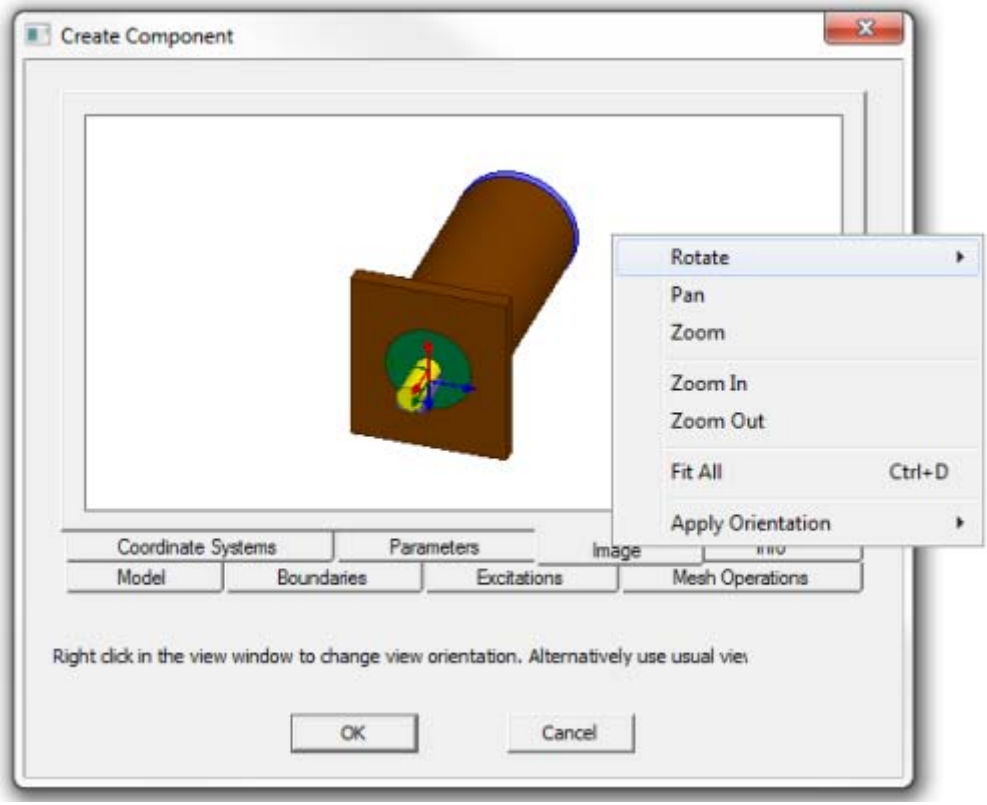
- Use the checkbox to Include or Exclude variables. You can also use the text field and **Include**/**Exclude** buttons.
- You can fill in the description field, if desired.



Create Component Image Tab

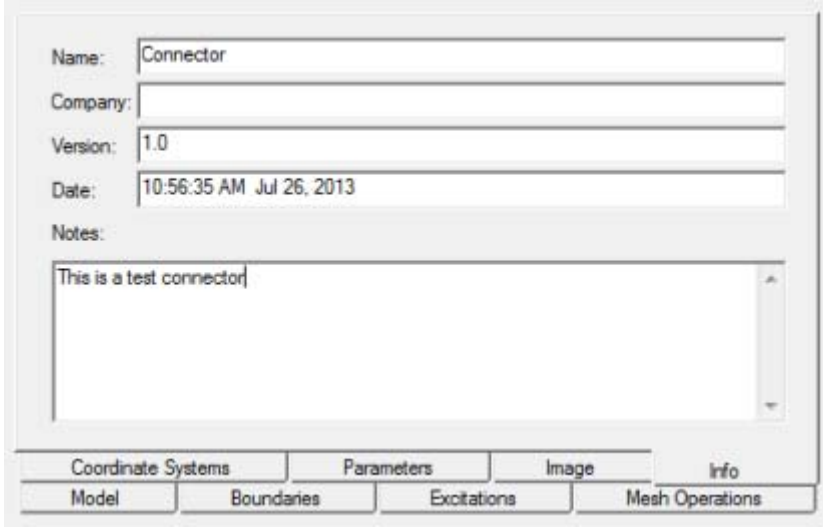
The **Image** tab shows the preview image for the component, based on the reference Coordinate System. The image also responds when you **Include** or **Exclude** objects from the **Model** tab.

- You can right-click for a menu to change the view of the preview image.
- Changing the preview orientation does not affect the modeler window view.



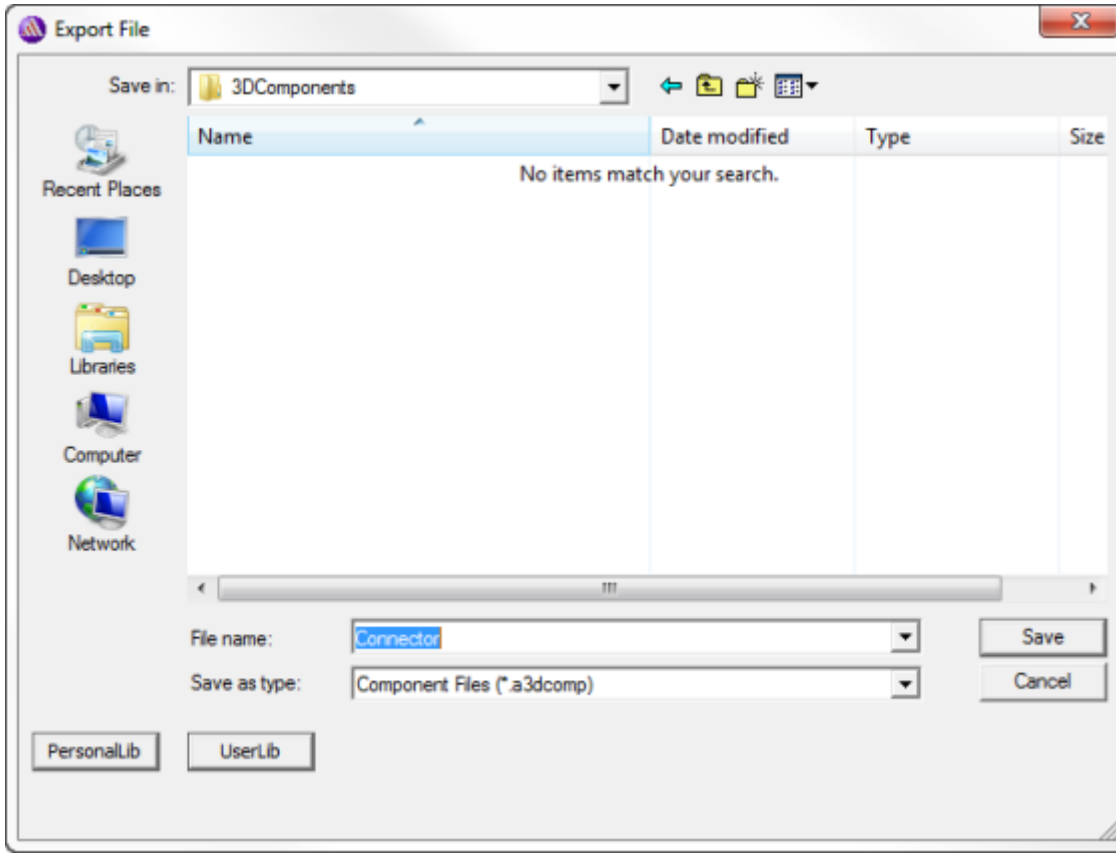
Create Component Info Tab

On the **Info** tab you can view and edit component information, as well as add notes.



Save 3D Component File

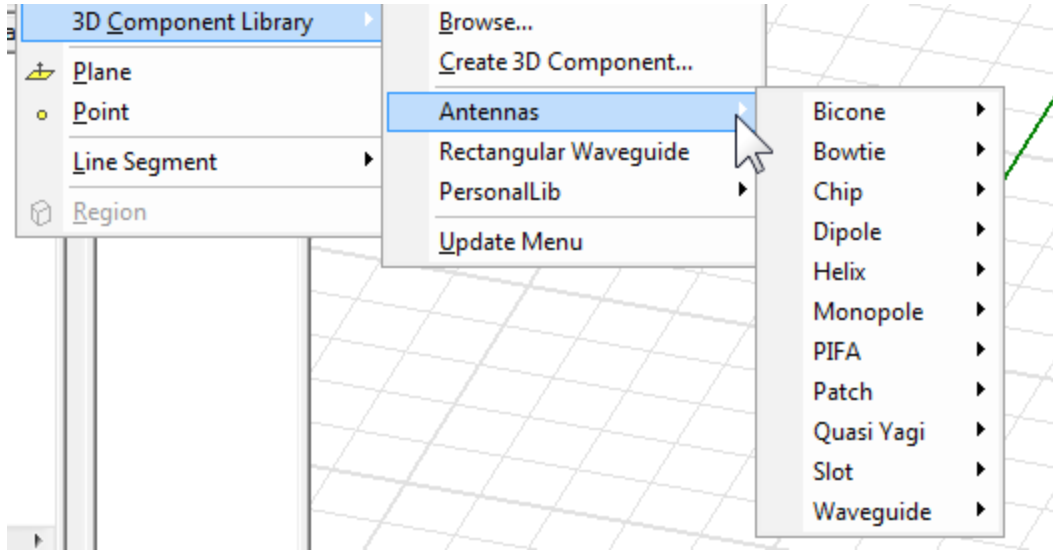
When you **OK** the **Component File** dialog, an **Export File** dialog displays. By default the Save in field shows the model source directory. Click **PersonalLib** or **UserLib** to display a 3D Components directory.



The default File name is the Component name specified in the **Info** tab.

3D Component Library: Antennas

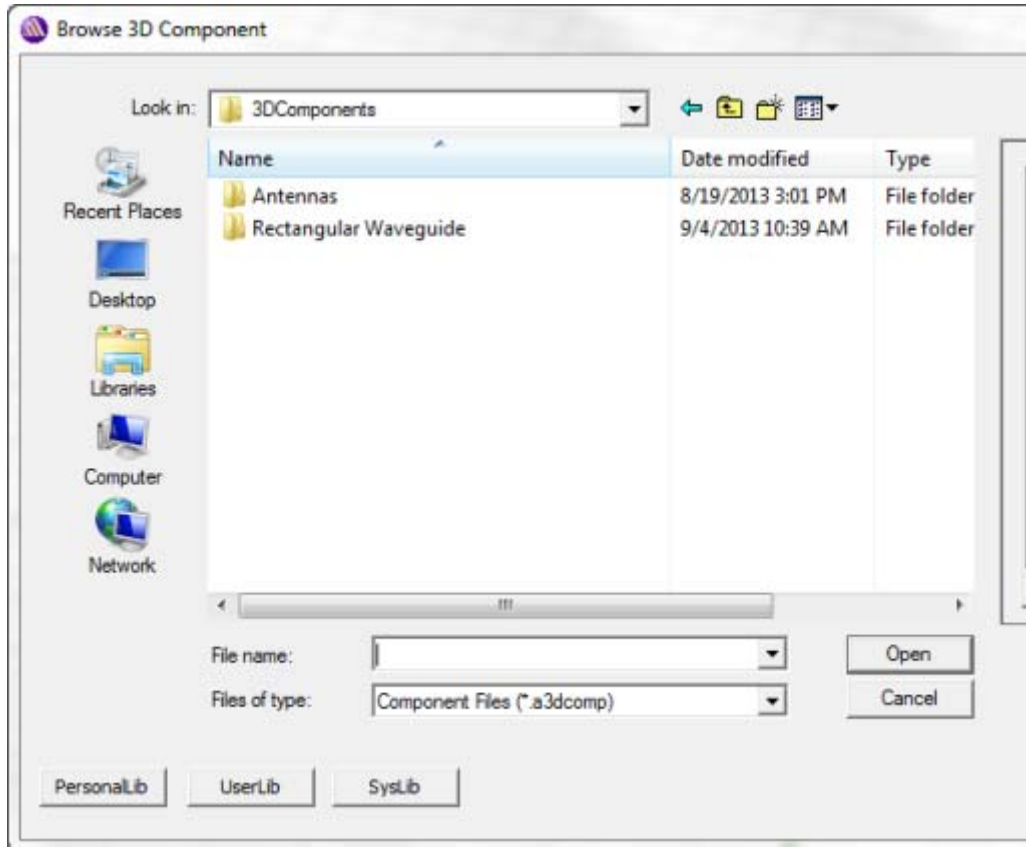
Click **Draw>3D Component Library>Antennas** to see a menu listing the kinds of antennas available..



Selecting a specific component from the cascading menu opens the **Create 3D Component** dialog for that component. The library components contain tabs for Parameters, Image, and Info.

You can also access the Antennas library using **Draw>3D Component> Browse...** to open the **Browse 3D Component** dialog. you then click the SysLib button to display the libraries included

in your installation. You can select Antennas, and navigate the folders to select from available components.

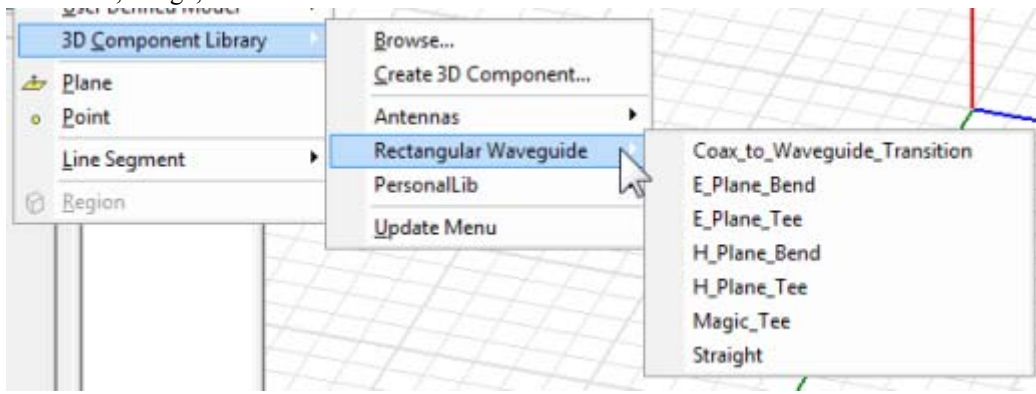


Once you have selected a Component file, view the Image and Info for that component. You can click Open to display the **Insert 3D Component** dialog. You can view the Parameters, Image, and Info tabs for that component. See Inserting a Component into a Design.

3D Component Library: Rectangular Waveguides

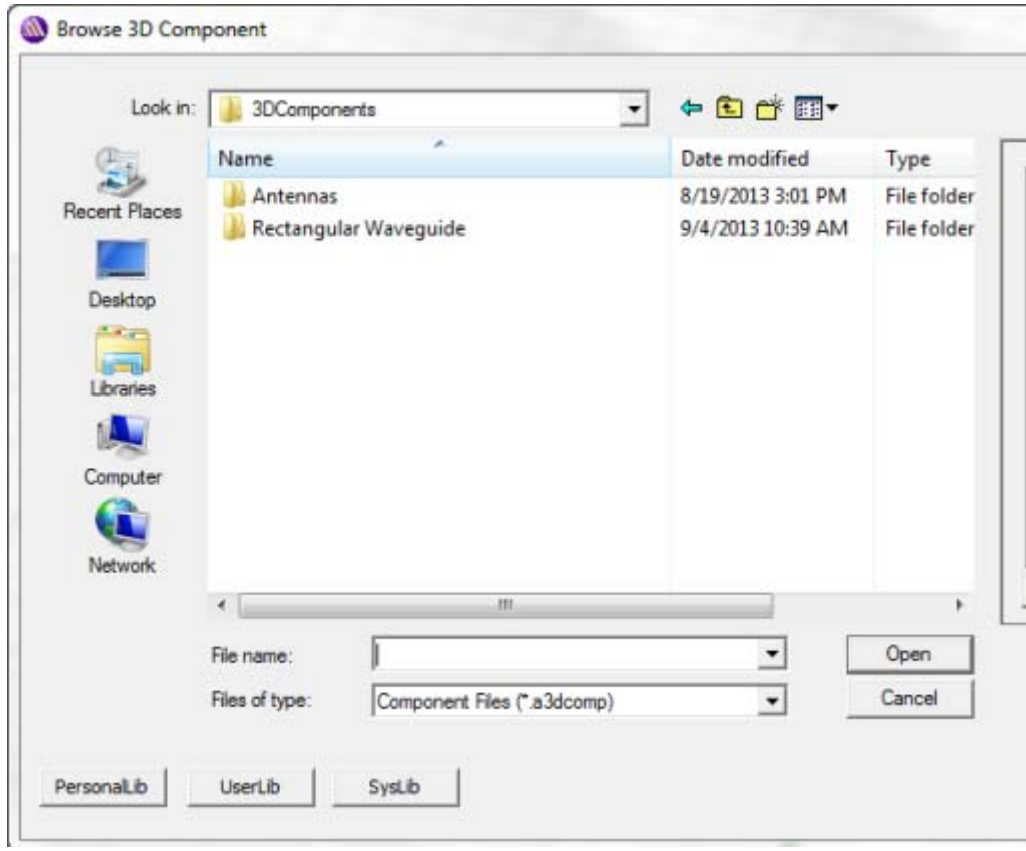
Click **Draw>3D Component Library>Rectangular Waveguides** to see a menu listing the kinds of waveguides available. Selecting a specific component from the cascading menu opens the **Cre-**

ate 3D Component dialog for that component. The library components contain tabs for Parameters, Image, and Info.



You can also access the Waveguides library using **Draw>3D Component> Browse..** to open the **Browse 3D Component** dialog. you then click the SysLib button to display the libraries included

in your installation. You can select Rectangular Waveguides, and navigate the folders to select from available 3D components.

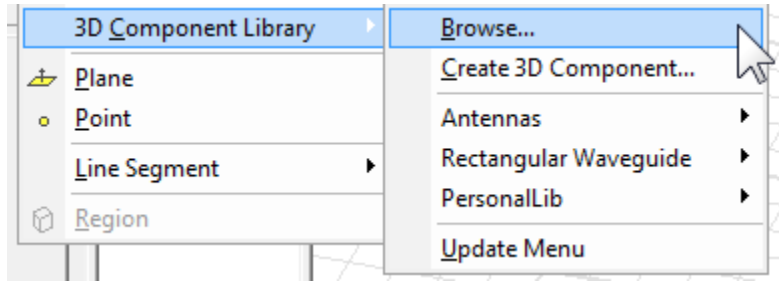


Once you have selected a Component file, view the Image and Info for that component. You can click Open to display the Insert 3D Component dialog. You can view the Parameters, Image, and Info tabs for that component. See Inserting a Component into a Design.

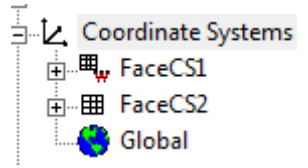
Inserting a Component in a Design

Once you save one or more components to a library, and create target coordinate systems in the design where you intend to place the component, you can use D**raw>3D Component**

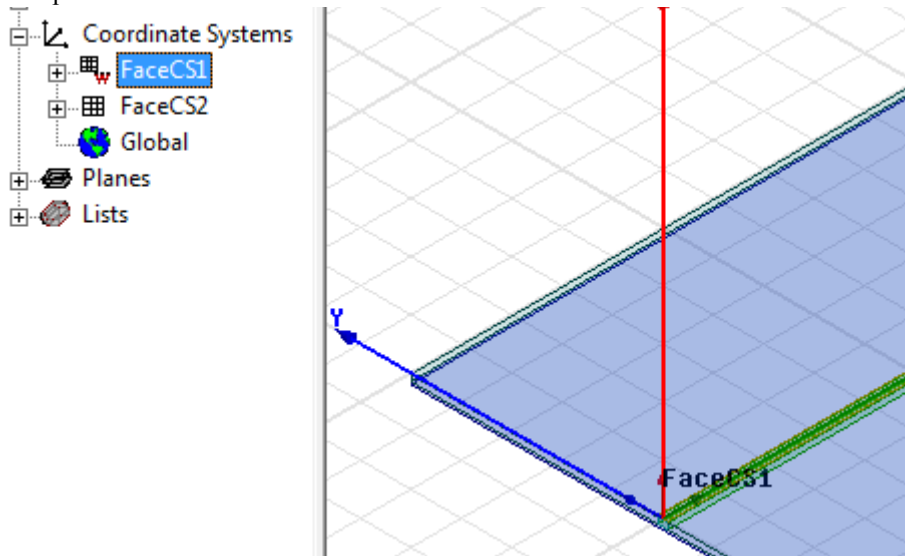
Library>Browse or **Draw>3D Component Library>Antennas** or **Draw>3D Component Library>Rectangular Waveguides** to select a component to insert into a design.



A target coordinate system provides a location for a component. For example, a design includes two additional coordinate systems:



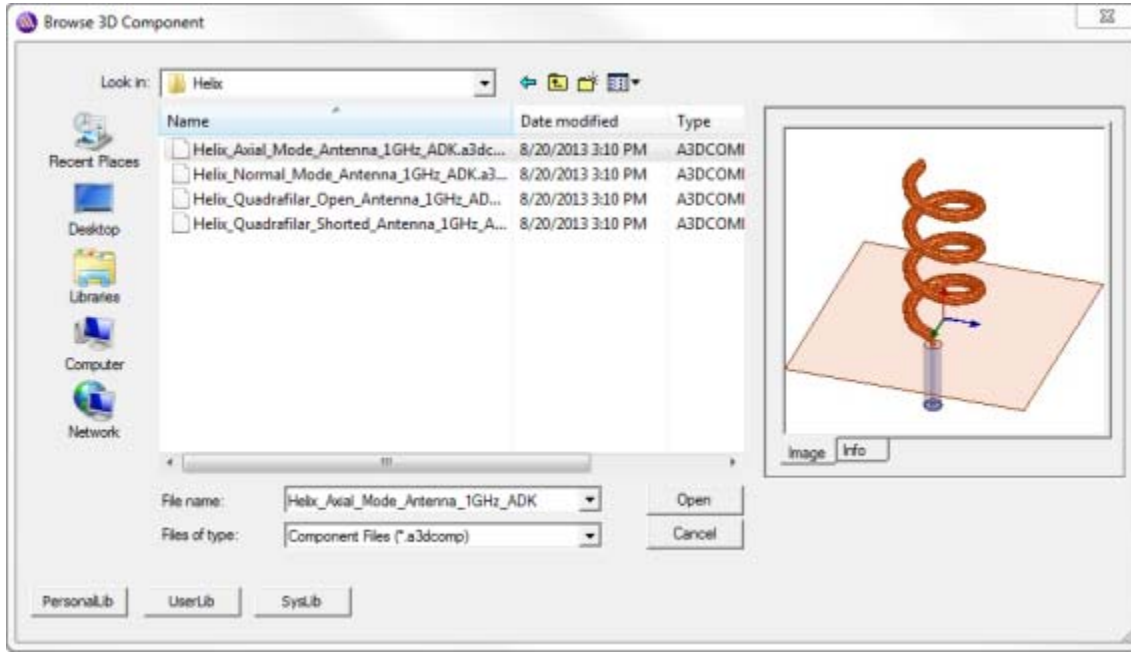
Selection of a coordinate system in the History tree displays a potential location for inserting a component.



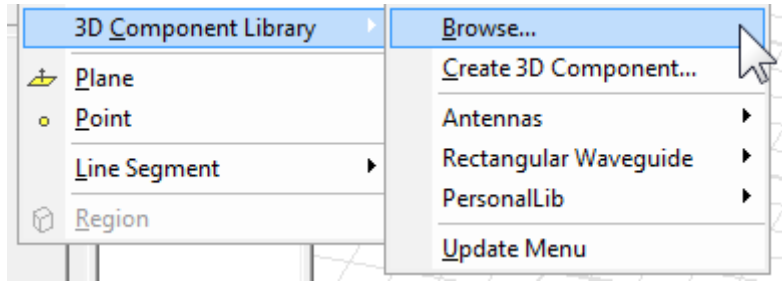
To Insert a Component

1. You can click **Draw>3D Component Library>Browse...** to open the **Browse 3D Component** dialog. You can navigate the file system to the component. Click to to select the Personallib, UserLib, or Syslib where the previously saved components can be found. Selecting the

*.a3dcomp file causes a display of the component image and the File name.

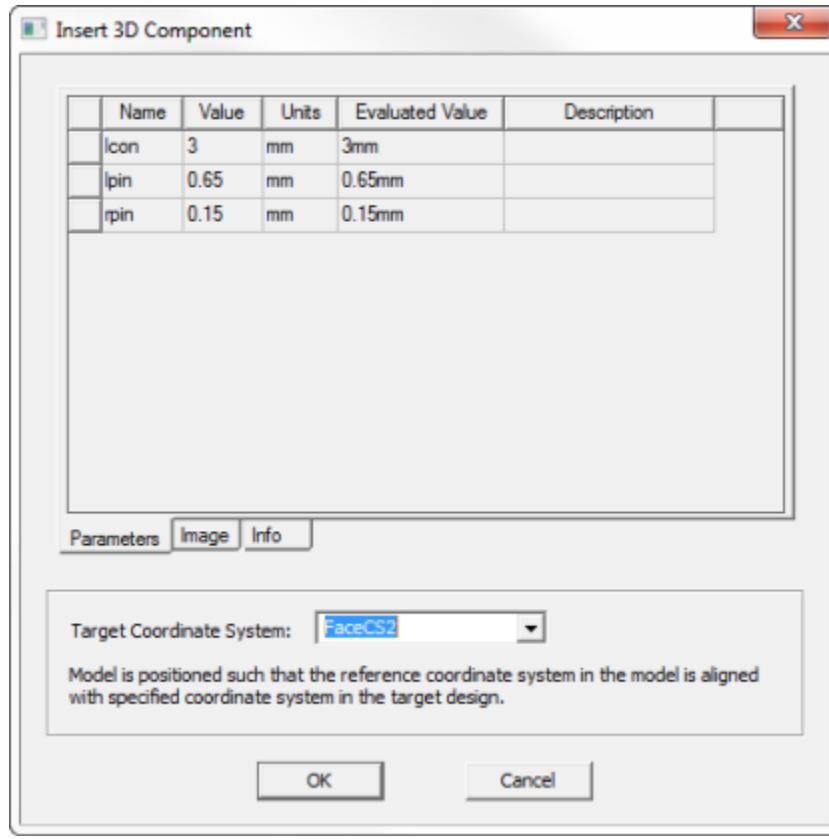


Alternatively, you can click **Draw>3D Component Library>** to navigate to a library where you have placed components and updated the menu for their display.



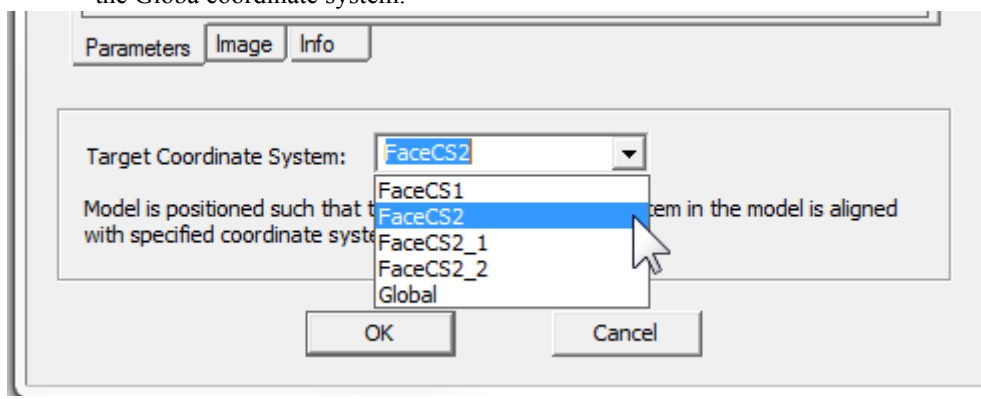
2. Click the **Open** button in the **Browse Component** dialog to close it and display the **Insert**

Component dialog.

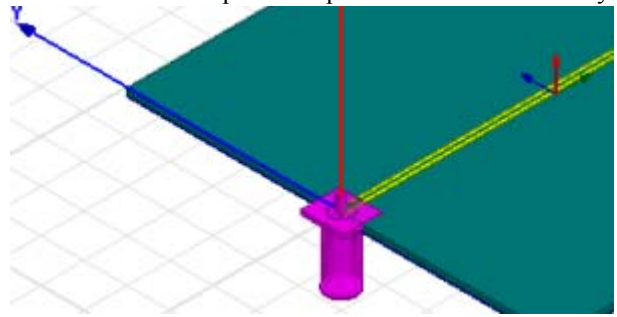


Tabs let you view the Parameters, Image, and Info. You can edit parameter values, and assign variables or expressions for parameters

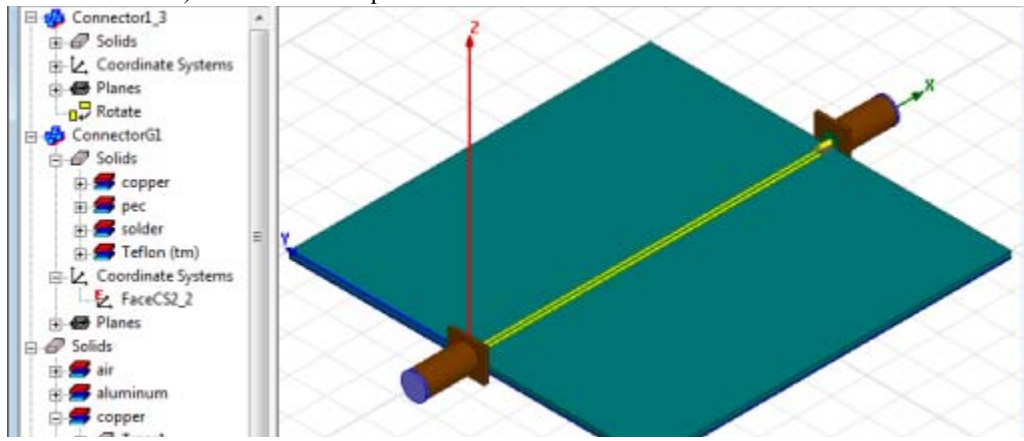
3. Use the menu to select the Target Coordinate System if any have been defined in addition to the Globa coordinate system.



4. When you click **OK** the component is placed at the coordinate system you selected.



- The Component folder in history tree shows objects, coordinate system (and planes from CS) included in component.



- Component properties show parameters
- Boundaries, excitation etc go to their respective folder in project manager

Editing Components

You can edit components in several ways.

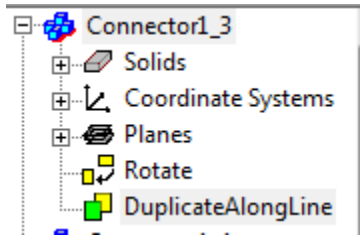
- Edit component geometry by editing its parameters.
- Edit component attributes like component name, part names, material assignment, and model/non-model flags.
- Edit boundaries, excitation and mesh operations coming from component
- Add additional boundaries/excitations on parts coming from component

  However, operations on individual parts of component are not allowed.
- Copy/Paste component
- Delete component
- Arrange component: Move, Rotate, and Mirror
- Duplicate component: Along Line, Around Axis, and Mirror

- Use **Edit>Select>Submodel** to quickly select component

Component Operations

- Component operations appear under that component folder towards the end.



- Component operations acts on parts as well as CS.
- Boundaries, mesh operations are duplicated with duplicate and copy/paste operation, provided that you check the option to do so (see Tools>Options/<design type options>).

Modifying Objects

You can quickly modify the position, dimensions, and other characteristics of objects created in the 3D Modeler window.

What do you want to do?

Object and History Editing

Modify Object Appearance

Modify Object Dimensions

Modify Object Location or Orientation

Surface or Edge Operations

Boolean Operations on Objects

Object and History Editing

- Copy and paste objects.
- Delete objects
- Delete Last Operation
- Cutting Objects
- Duplicate objects
- View and Edit Commands on History Tree Objects
- Purge History
- Generate History

Modify Object Appearance

- Assign color to an object.
- Assign transparency to an object.
- Also see Modifying the Model View

Modify Object Dimensions

- Assigning a Cross Section and Dimension to a Polyline
- Scale the size of objects
- Connect objects
- Move faces or edges
- Convert polyline segments
- Rounding the edge of an object (Fillet)
- Flattening the edge of an object (Chamfer)

Modify Object Location or Orientation

- Move objects
- Rotate objects
- Change the Orientation of an object

- Mirror objects about a plane.
- Offset an object (move every face of an object).
- Also see Modifying the Model View

Surface or Edge Operations for Objects

- Sweep objects.
- Cover lines.
- Cover faces.
- Uncover faces.
- Detach faces.
- Detach edges.
- Create a new object by taking a cross-section of a 3D object.
- Wrap Command
- Project Sheet
- Thicken Sheet

Boolean Operations on Objects

- Unite objects.
- Subtract objects.
- Create objects from intersections.
- Create an object from a face.
- Create an object from an edge.
- Split objects.
- Separate objects.
- Imprint Projection Commands
- Imprinting an Object

Assigning Color to Objects

1. Select the object to which you want to assign a color.

   If the **Properties** window not visible on the desktop, click **View>Properties Window** or use **Edit>Properties**.

2. In the **Properties** dialog box, click the **Attribute** tab.
3. Click **Edit** in the **Color** row.

   The **Color** palette appears.
4. Select a color from the **Color** palette, and then click **OK**.

   The color is assigned to the selected object.

Related Topics

Setting the Default Color of Objects

Setting the Default Color of Objects

1. Click **Tools** >**Options**>**Modeler Options**.
2. Click the **Display** tab.
3. Select **Object** from the **Default color** pull-down list.
4. Click the color button beside the **Default color** pull-down list.

   The **Color** palette appears.
5. Select a color from the **Color** palette, and then click **OK**.

   Any objects you draw after this point will be assigned the default color you selected.

Setting the Default Color of Object Outlines

1. Click **Tools>Options>Modeler Options**. Click the **Display** tab.
2. Select **Object Wire** from the **Default color** pull-down list.
3. Click the color button beside the **Default color** pull-down list.

   The **Color** palette appears.
4. Select a color from the **Color** palette, and then click **OK**.

   The outlines of any objects you draw after this point will be assigned the default color you selected.

Assigning Transparency to an Object

1. Select the object to which you want to assign a transparency.

Note If the **Properties** window not visible on the desktop, click **View>Properties Window** or use **Edit>Properties**.

2. In the **Properties** dialog box, click the **Attribute** tab.
3. Click the value in the **Transparency** row.

   The **Set Transparency** window appears.
4. Move the slider to the right to increase the transparency of the object. Move the slider to the left to decrease the transparency of the object.
5. Click **OK**.

Related Topics

Setting the Default Transparency of Objects

Setting the Default Transparency of Objects

1. Click **Tools**>**Options**>**Modeler Options**. Click the **Display** tab.
2. Move the **Default transparency** slider to the right to increase the transparency of objects.

   Move the slider to the left to decrease the transparency of objects.

Any objects you draw after this point will be assigned the default transparency you selected.

Copying and Pasting Objects

To copy objects and paste them in the same design or another design, use the **Edit>Copy** and **Edit>Paste** commands. For data link purposes, where you want to include the material assignments as well as the geometry, you can use the Import from Clipboard command.

1. Select the objects you want to copy.
2. Click **Edit**>**Copy**  .

   The objects are copied to the Clipboard, a temporary storage area. The selected items are not deleted.

   To cut an item to the clipboard and deleting the original, use the scissors icon on the toolbar.
3. Select the design into which you want to paste the objects. It can be the same design from which you copied the items.
4. Click in the **3D Modeler** window.
5. Select the working coordinate system. Objects are pasted relative to the current working coordinate system.
6. Click **Edit**>**Paste**  .

   The objects appear in the new window.

Items on the Clipboard can be pasted repeatedly. The items currently stored on the Clipboard are replaced by the next items that are cut or copied.

Related Topics

Duplicating Boundaries and Excitations with Geometry

Import a Model from the Clipboard

You can import a model to the Clipboard in order to use a geometry from a different design. To use a geometry with datalink, the geometry ID must be preserved.

To import a model from the Clipboard, the model for the current design must be empty.

The geometry model is imported from the Clipboard with the ID preserved.

1. Select the objects you want to copy. For selecting all objects, you can use **Edit>Select All** or **Ctrl-A**.
2. Click **Edit**>**Copy**  .

   The objects are copied to the Clipboard, a temporary storage area. The selected items are not deleted.
3. Select the design into which you want to paste the objects. It can be the same design from which you copied the items.
4. Click in the **3D Modeler** window.
5. Select the working coordinate system. Objects are imported relative to the current working

coordinate system.

6. Click **Modeler**>**Import From Clipboard**.

The geometry model is pasted from the Clipboard with the ID preserved.

Related Topics

Setup Link Dialog

Selecting Items in the 3D Modeler Window

Copy Image

You can import images of the **3D Modeler** window or of Reports into any other application. The image has to be copied to the clipboard, so that it can be imported into the other application.

To copy an image of the 3D Modeler window and paste into another application:

1. Make the 3D Modeler window active.

   This enables the **Edit>Copy Image** command in the menu bar.

2. Click **Edit>Copy Image**, or right click on the 3D Modeler window to display the shortcut menu and select **Copy Image**.

   The **3D Modeler** window is copied to the Clipboard as an image.

3. Select and open the application into which you want to paste the objects, and paste the image.

To copy an image of a Report to paste into another application:

1. Make the report the active window.

   The enables the **Edit>Copy Image** command in the menu bar.

2. Click **Edit>Copy Image**, or right click on the Report window to display the shortcut menu and select **Copy Image**.

   The report is copied to the Clipboard as an image.

3. Select and open the application into which you want to paste the objects, and paste the image.

Related Topics

Copy and Paste of Report and Trace Data

Copy and Paste of Report and Trace Definitions

Deleting Objects

1. Select the objects to delete.
2. Click **Edit**>**Delete**  .
   - Alternatively, press **Delete**.

The objects are deleted.

Note To maintain valid boundaries, excitations, or other parameters that were associated with the deleted object, reassign them to other objects.

Related Topics

Deleting Polyline Segments

Deleting Start points and End points

Deleting Start Points and Endpoints

If you select a polyline in the history tree, the **Delete Start Point** and **Delete End Point** commands may be enabled. These permit you to delete portions of the line.

1. In the history tree, locate the polyline that contains the segment you want to delete. Expand this part of the history tree.
2. In the history tree, select the polyline you want to edit.

   The segment is highlighted.
3. On the **Edit** menu or the shortcut menu, click either **Delete Start Point** to remove the leading segments or **Delete End Point** to remove the following segments.

   The designated segment is removed, and the line changes.

Delete Last Operation

To delete the last operation on an object:

1. Select the object.
2. Click **Modeler>Delete Last Operation.**

   This undoes the last operation, including removing that operation from the history, and updating the context for the **Undo** and **Redo** commands.

Related Topics

Undoing Commands

Redoing Commands

Cutting Objects

1. Select the objects to cut.
2. Click **Edit**>**Cut**.

The objects are copied to the Clipboard and deleted from the design.

Moving Objects

1. Select the objects to move.
2. Click **Edit**>**Arrange**>**Move**  .
3. Select an arbitrary anchor point in one of the following ways:
   - Click the point.
   - Enter the point's coordinates in the **X**,**Y**, and **Z** boxes.
4. Select a target point in one of the following ways:
   - Click the point.
   - Type the coordinates of a point relative to the anchor point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   All selected objects move the distance determined by the offset between the anchor point and the target point.

Rotating Objects

Rotate objects about the x-, y-, or z-axis using the **Edit>Arrange>Rotate** command.

To rotate objects about an axis:

1. Select the objects to rotate.
2. Click **Edit**>**Arrange**>**Rotate**  .

   The **Rotate** dialog box appears.
3. Select the axis about which to rotate the objects: **X**, **Y**, or **Z**.
4. Type the angle to rotate the objects in the **Angle** box.

   A positive angle causes the object to be rotated in the counter-clockwise direction. A negative angle causes the object to be rotated in the clockwise direction.
5. Click **OK**.

   The selected objects are rotated about the axis.

To rotate *and copy* objects, use the **Edit>Duplicate>Around Axis** command.

Changing the Orientation of an Object

Each object has an **Orientation** property that specifies the coordinate system it uses is Global, or a user defined orientation relative to the Global coordinate.

This property is useful in dealing with anisotropic materials. The properties of anisotropic materials are specified relative to the objects orientation. Changing the orientation of an object provides a way for objects made of the same material to be orientated differently.

To change an object's orientation.

1. Define the coordinate systems you want to have available.
2. Open the properties window for the object.
3. Click on the Orientation property, and select from the Drop down list. If no Orientations other than Global have been defined, none appear on the list.

4. Click OK to close the dialog and apply the changes.

Related Topics

Assigning Material Property Types

Setting Coordinate Systems

Creating a Relative Coordinate System

Mirroring Objects

Mirror an object about a plane using the **Edit>Arrange>Mirror** command. The plane is selected by defining a point on the plane and a normal point. This command allows you to move an object and change its orientation.

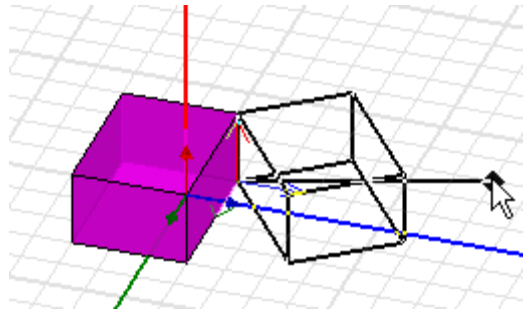
Note The distance between the point on the mirror plane and the point along the normal does not matter — only the vector direction matters.

To mirror an object about a plane:

1. Select the object or objects that you want to mirror. You can select multiple objects.
2. Click **Edit**>**Arrange**>**Mirror**  .
3. Select a point on the plane around which you want to mirror the object.

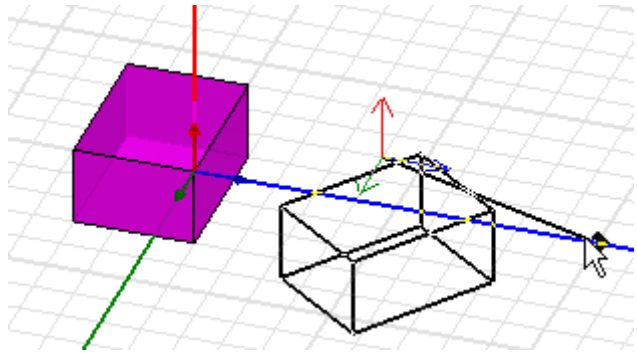
   You can do this by clicking a point, or typing coordinates in the X, Y, and Z boxes in the status bar.

   If you select a point on the object, the mirroring is relative to that point on the object. In the following example, the first point clicked after selecting **Edit>Arrange>Mirror** was on the right-rear bottom corner of the selected object. So the axis of rotation as you move the cursor is that corner. As you move the cursor, it drags a diamond-shape on a vector extending from the initial point. The distance along the vector does not matter. Moving the mouse rotates an outline of the object to new orientations. Clicking the mouse moves the object to location indicated by the outline..



   In this second example, the initial point is at a distance from the original object, designated by

the triad from which the handle for rotation extends to the dragging cursor.



4. Select a second point in one of the following ways:
   - Click the point.
   - Type the coordinates of a point relative to the first point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point. As you type in the values, the outline moves to the coordinates. Hit the Enter key to complete the command.

   The selected object is moved to the plane you specified and oriented relative to the normal point you specify.

To mirror *and copy* objects about a plane, use the **Edit>Duplicate>Mirror** command.

Related Topics

Duplicating and Mirroring Objects

Offsetting Objects

Move every face of a 3D object in a direction normal to its surface using the **Edit>Arrange>Offset** command. The faces are moved a specified distance normal to their original planes. This command enables you to move every face of a solid object without having to individually select and move each face. Use the **Surfaces>Move Faces>Along Normal** command if you want to move just one or more faces of an object.

To offset every face of an object:

1. Select the object you want to offset.
2. Click **Edit>Arrange>Offset**.

   The **Offset** dialog box appears.
3. Type the distance you want to move the object faces from their origins, and then select a unit from the pull-down list.
4. Click **OK**.

   The selected object's faces are moved the distance you specified.

Duplicating Objects

You can duplicate objects within a design using the **Edit>Duplicate** commands. Duplicates are dependent upon the parameters of their *parent* object at the time they were created, that is, they share the parent object's history at the time of creation. The command hierarchy in the history tree will show the duplication command, illustrating which commands affect all duplicates (those performed before the duplication) and which commands would not affect the duplicates (those performed after the duplication). For example, if you modify the radius of a parent object's hole, the change is applied to the holes of the object's duplicates because they share the radius specification history, but if you move the faces of the parent object, its duplicates are not affected because this operation took place after the duplicates were created.

Operations performed on duplicates are independent. For example, if you duplicate a cylinder twice, creating a row of three, and then split the second cylinder, the first and third cylinders are not affected by the split.

When creating duplicates, the parent object is duplicated along a line or around an axis the number of times you specify. You can also create a single duplicate that mirrors the parent object about a plane.

Choose from the following commands:

Edit>Duplicate>Along Line	Duplicates the parent object along a straight line. The child object can be designated as attached to the parent object, but if so, no ports or boundary conditions are duplicated.
Edit>Duplicate>Around Axis	Duplicates the parent object around an axis. The child object can be designated as attached to the parent object, but if so, no ports or boundary conditions are duplicated.
Edit>Duplicate>Mirror	Duplicates a mirror image of the parent object about a plane.

To copy objects to another design, use the **Edit>Copy** and **Edit>Paste** commands.

Note There is currently no method for dissolving the parent/duplicate relationship once a duplicate has been created.

Duplicating Objects Along a Line

To duplicate an object along a straight line, use the **Edit>Duplicate>Along Line** command. The line along which the object is duplicated can be vertical, horizontal, or lie at an angle.

1. Select the object you want to duplicate.
2. Click **Edit>Duplicate>Along Line**.  .
3. Specify the vector along which the object will be duplicated:
   a. Select an arbitrary anchor point in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

      Any point in the drawing region can be selected; however, selecting an anchor point on the

object's edge or within the object makes it easier to select the duplication line.

b. Select a second point in one of the following ways:

- Click the point.
- Type the coordinates of a point relative to the anchor point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

This point defines the direction and distance from the anchor point to duplicate the object.

The **Duplicate Along Line** dialog box appears.

4. Type the total number of objects, including the original, in the **Total Number** box.
5. By option check the **Attach to Original Object** checkbox. If this is checked, no ports or boundary conditions are duplicated for the child.
6. Click **OK**.

The duplicates are placed along the vector you specified.

Duplicating Objects Around an Axis

To duplicate an object around the x-, y-, or z-axis, use the **Edit>Duplicate>Around Axis** command.

1. Select the object you want to duplicate.
2. Click **Edit>Duplicate>Around Axis**  .

The **Duplicate Around Axis** dialog box appears.

3. Select the axis around which you want to duplicate the object: **X**, **Y**, or **Z**.
4. Type the angle between duplicates in the **Angle** box.

A positive angle causes the object to be pasted in the counter-clockwise direction.

A negative angle causes the object to be pasted in the clockwise direction.

5. Type the total number of objects, including the original, in the **Total Number** box.
6. By option check the **Attach to Original Object** checkbox. If this is checked, no ports or boundary conditions are duplicated for the child.
7. Click **OK**.

The object is duplicated around the axis at the angle you specified.

Duplicating and Mirroring Objects

To duplicate and mirror an object about a plane, use the **Edit>Duplicate>Mirror** command. The plane is selected by defining a point on the plane and a normal point. This command allows you to duplicate an object and specify the duplicate's position.

This command is similar to **Edit>Arrange>Mirror**, except that this command duplicates an object, rather than moves it.

1. Select the object you want to mirror.

Click **Edit>Duplicate>Mirror**  .

2. Select a point on the plane on which you want to mirror the object.

A line drawn from this point to the mirror plane will be perpendicular to the plane. The distance between the point on mirror plane and point along the normal does not matter; only the vector direction matters

3. Select a normal point on the plane in one of the following ways:
   - Click the point.
   - Type the coordinates of a point relative to the first point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   A duplicate of the object appears on the plane you specified, oriented according to the normal point you specified.

Related Topics

Mirroring Objects

Scaling Objects

Scale an object's dimensions in one or more directions using the **Edit>Scale** command.

The scale of an object is determined by the distance of each of its vertices from the origin of the model coordinate system. When an object is scaled, the distance of each vertex from the origin is multiplied by the scaling factor, causing the object to be resized and/or moved.

For example, if you specify a scaling factor of 2 in the X direction, each vertex in the model will be moved so that the distance to its origin is doubled. Note that a vertex located at the origin will not move. You can alter an object's proportions by scaling it in one direction.

To scale an object's dimensions in one or more directions:

1. If necessary, set a different working coordinate system to achieve the desired scaling.
2. Select the object to scale.

   Click **Edit>Scale**. The **Scale** dialog box appears.
3. Type the scale factor for each axis.
4. Click **OK**.

   The object is scaled about the working coordinate system's origin.

Related Topics

Modifying Object Attributes using the Properties Window

Sweeping Objects

You can sweep a 2D object around an axis, along a vector, or along a path to create a 3D solid object. Objects that can be swept include circles, arcs, rectangles, trapezoids, polylines, or any 2D object created in the **3D Modeler** window. The 2D object need not be orthogonal to the sweep path.

You can also thicken sheets to make a 3D object.

You can also sweep open 1D objects, such as polylines. This results in open 2D sheet objects.

You can also sweep one or more faces of a 3D object to create a new object. See Sweep Faces Along Normal.

Related Topics

Assigning a Cross Section and Dimension to a Polyline

Sweeping Around an Axis

Sweeping Along a Vector

Sweep Along a Path

Sweep Faces Along Normal

Thicken Sheet

Sweeping Around an Axis

Sweep a 1D or 2D object around the x-, y-, or z-axis using the **Draw>Sweep>Around Axis** command. Sweeping circles around an axis is a convenient way to create an open coil loop.

Before using this command, keep the following guidelines in mind:

- The object and the axis you are sweeping around must lie in the same plane. For example, if you are sweeping an object around the z-axis, the object must lie in a plane that includes the z-axis, such as xz or yz.
- The normal of the object's plane faces must be perpendicular to the axis around which you are sweeping.
- The object may not cross the axis around which it is being swept.

To sweep an object around an axis:

1. Select the object you want to sweep.
2. Click **Draw>Sweep>Around Axis**.

   The **Sweep Around Axis** dialog box appears.
3. Select the axis you want to sweep the object around: **X**, **Y**, or **Z**.
4. Type the angle to sweep the object through in the **Angle of sweep** box.

   The value must be between **-360** and **360** degrees.
5. Type the draft angle.

   This is the angle to which the object's profile, or shape, is expanded or contracted as it is swept.
6. Select one of the following draft types from the pull-down list. The draft type instructs the modeler how to fill in gaps created by expanding or contracting a profile with a draft angle.

Extended	The edges of the new profile are extended with straight tangent lines until they intersect. The facetting of the faces will be displayed.
Round	The edges of the new profile are rounded.
Natural	The edges of the new profile are extended along their natural curves until they intersect. For example, if the original object had sharp edges, the new profile will have sharp edges.

7. Type the number of segments in the **Number of segments** text box.Click **OK**.

Note The default number of segments is zero, which creates a true path. A positive value results in a segmented sweep, while a negative value results in an error.

If the sweep angle is 360 degrees, the number of segments is equal to the value specified. If the sweep angle is less than 360 degrees, half segments appear at the ends.

Projects and scripts from previous software versions are treated as if the number of segments were zero.

The object is swept around the axis. The new object has the properties of the original object. The **Properties** dialog box appears, enabling you to modify the object's properties.

8. Click **OK**.

Sweeping Along a Vector

Sweep a 1D or 2D object along a vector using the **Draw>Sweep>Along Vector** command.

1. Select the object you want to sweep.

Click **Draw**>**Sweep>Along Vector**.Draw the vector you want to sweep the object along:

a. Select the start point by clicking the point or typing its coordinates in the **X**, **Y**, and **Z** boxes.

b. Select the endpoint in one of the following ways:

- Click the point.
- Type the coordinates of a point relative to the start point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

The **Sweep Along Vector** dialog box appears.

2. Type the draft angle.

This is the angle to which the profile is expanded or contracted as it is swept.

3. Select one of the following draft types from the pull-down list box:

Extended	The new object will have sharp edges like the original object. The facetting of the faces will be displayed.
Round	The new object will have rounded edges.
Natural	The new object will have sharp edges like the original object

The object is swept along the vector. The new object has the name and color of the original profile. The **Properties** dialog box appears, enabling you to modify the object's properties.

4. Click **OK**.

Sweeping Along a Path

Sweep a 1D or 2D object along a path that is defined by an open or closed polyline using the **Draw>Sweep>Along Path** command.

When you are sweeping an object along a path, keep in mind that one of the path's endpoints must lie in the same plane as the object being swept. The other endpoint must lie in a plane perpendicular to the object being swept.

To sweep an object along a path:

1. Create the polyline you want to use as a path.
2. Select the object you want to sweep, and then select the new polyline.
3. Click **Draw>Sweep>Along Path**.

   The **Sweep Along Path** dialog box appears.
4. Type the angle of the twist in the path.

   This is the number of degrees the profile will rotate as it is swept through the complete path.
5. Type the draft angle.

   This is the angle to which the profile is expanded or contracted as it is swept.
6. Select one of the following draft types from the pull-down list box:

Extended	The new object will have sharp edges like the original object. The facetting of the faces will be displayed.
Round	The new object will have rounded edges.
Natural	The new object will have sharp edges like the original object

7. Click **OK**.

   The object is swept along the path. The polyline object used as the path is deleted. The new object has the properties of the original object. The **Properties** dialog box appears, enabling you to modify the object's properties.

Sweeping Faces Along Normal

To create a new object by sweeping select 3D object's face a specified distance in a direction normal to its original plane, use the **Modeler>Surface>Sweep Faces Along Normal** command. Note that the adjoining faces will not be sheared or bent.

This command is useful for extruding faces, resizing holes, and removing rounded corners.

To sweep selected object faces in a normal direction:

1. Click **Select Faces** on the shortcut menu.
2. Select the faces of the object you want to sweep.
3. Click **Modeler>Surface>Sweep Faces Along Normal**.
4. The **Sweep Faces Along Normal** dialog box appears.
5. Type the distance you want to sweep the object face from its origin.
6. Click **OK**.

   The face is swept the distance you specified to create a new object.

Related Topics

Moving Faces Along the Normal

Thicken Sheet

To thicken one or more sheet objects to make 3D objects:

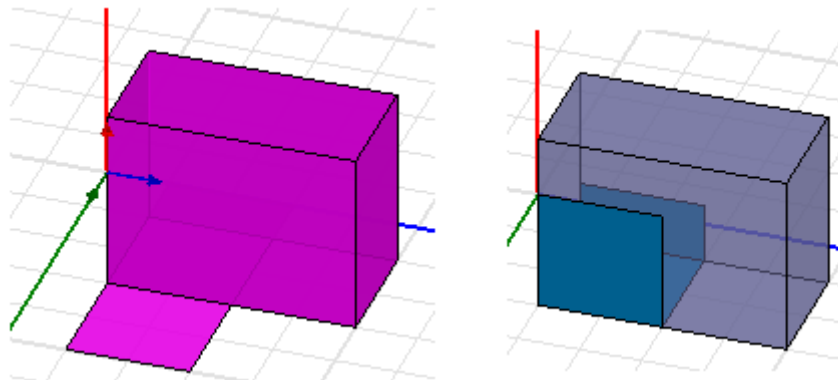
1. Select the sheet or sheets.
2. Click **Modeler>Surface>Thicken Sheet**.

   The **Thicken Sheet** dialog appears.
3. Specify the thickness by typing in the field.
4. Specify the units by selecting from the drop down menu.
5. If you want to thicken both sides, use the checkbox.
6. Click **OK**.

   The dialog closes and the sheets are changed into 3D objects of the desired thickness.

Wrap Command

You can use **Modeler>Surface>Wrap** command to wrap a sheet object around a suitable 3D object (rectangular or segmented.) The sheet object must be in contact with the 3D object. It should have smaller dimensions than the 3D object. If the sheet object does not overlap the corners of the 3D object, the wrap is straightforward, as shown in the figure. A sheet object that overlaps corners may not wrap in straightforward fashion, depending on both the angle(s) involved, and the sheet object. While it is possible, it is not recommended.



To wrap a sheet object:

1. Create a sheet in contact with an appropriate 3D object.
2. Select both objects.
3. Click **Modeler>Surface>Wrap**.

   The sheet object wraps around the 3D object. You can select the wrapped sheet object and the 3D object separately, and assign properties separately.

   If the object cannot wrap, the Message window contains a warning and description.

You can wrap multiple sheets on the same 3D object.

If you delete the 3D object, the wrapped sheet retains the form it took when wrapped.

You can use the **Tools>Options>Modeler** Options to automatically perform a **Modeler>Boolean>Imprint** command after performing a **Wrap.**

Related Topics

Imprinting an Object

Modeler Options: Operation Tab

Covering Lines

To cover a closed 1D polyline object with a face, use the **Modeler>Surface>Cover Lines** command. The polyline object becomes a 2D sheet object.

To convert a closed polyline object to a sheet object:

1. Select the closed polyline object you want to cover.
2. Click **Modeler>Surface>Cover Lines**.

   The object is now covered. It is now a 2D sheet object that can be swept to form a 3D solid object.

Note If you want the modeler to automatically cover all closed polyline objects you draw, including circles, ellipses, rectangles, and regular polygons, select the **Automatically cover closed polylines** option in the **Modeler Options** dialog box. A closed polyline object can also be created by using boolean unite operations on two or more polylines.

Covering Faces

To cover object faces, the faces must be united into a 3D sheet object. To cover the face of a 2D or 3D object, use the **Modeler>Surface>Cover Faces** command.

Covering the face of an open 2D sheet object that had previously been uncovered results in a 3D solid object. For example, for a box, when you select and uncover a face, the solid box becomes a sheet with five faces. When you then select that sheet body box and use the **Cover Faces** command, the box becomes a solid again with six faces.

To cover the faces of objects:

1. Select the faces of the objects you want to cover.
2. Click **Modeler>Surface>Cover Faces**.

The object faces are now covered.

Uncovering Faces

Uncover a surface of a 3D object using the **Modeler>Surface>Uncover Faces** command. Uncovering the surface of a 3D solid object results in an open 2D sheet object.

To uncover the face of a 3D object:

1. Switch to face selection mode: Click **Edit>Select>Faces**.
2. Select a face of the object you want to uncover.

Click **Modeler>Surface>Uncover Faces**. The selected face is uncovered, leaving an open face on the object.

Note You can uncover one face of a 3D object at a time. If you select multiple faces, only the first face will be uncovered.

Detaching Faces

The **Modeler>Surface>Detach Faces** command enables you to remove the face of a 3D object, resulting in two separate objects.

To detach the face of an object:

1. Switch to face selection mode: Click **Edit>Select>Faces**.
2. Select the face of the object you want to detach. You can select multiple faces to detach.

   Click **Modeler>Surface>Detach Faces**. The selected face is now detached, resulting in two 2D sheet objects.

Detaching Edges

The **Modeler>Edge>Detach Edges** command enables you to remove an edge of a wire object, resulting in two separate wire objects.

To detach an edge of an object:

1. Switch to edge selection mode: Click **Edit>Select>Edges**.
2. Select the edge of the object you want to detach. You can select multiple edges to detach.
3. Click **Modeler>Edge>Detach Edges**.

   The selected edge is now detached, resulting in multiple wire objects.

Note Only edges from wire bodies can be used in a detach edge operation.

Creating a Cross-Section

You can take a cross-section of a 3D object to create a new 2D object. This is done using the **Modeler>Surface>Section** command.

Use this command to create cross-sections of 3D objects on the xy, yz, or xz plane. The cross-sections are created as 2D closed polyline objects.

To create a cross-section of an object:

1. Make sure the working coordinate system you want to use for the cross-sectioning plane is set.
2. Select the object from which you want to create a cross-section.
3. Click **Modeler>Surface>Section**. Select the section plane you will use to divide the object: **XY**, **YZ**, or **ZX**.
4. Click **OK**.

   A closed polyline object is created from the object that was sliced by the selected axis. The original, sectioned object is unmodified.

Related Topics

Setting the Working Coordinate System

Connecting Objects

Use the **Modeler>Surface>Connect** command to perform the following operations:

- Connect two or more 1D polyline objects. HFSS will modify the first polyline you select to be a 2D sheet object that connects to the second and any subsequently selected polylines. The second and subsequent polylines selected are deleted.
- Connect two or more 2D sheet objects. HFSS will modify the first 2D object you select to be a 3D solid object that connects to the second and any subsequently selected objects. The second and subsequent objects selected are deleted.

To connect objects:

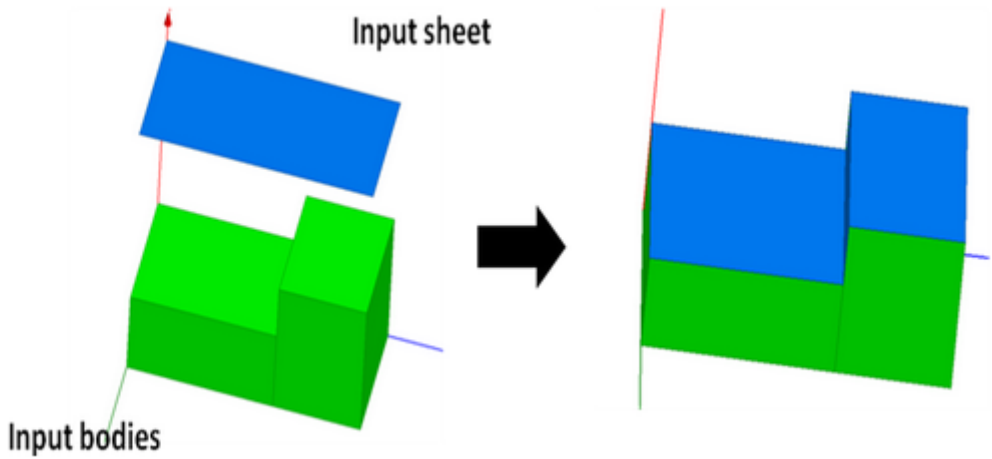
1. Select the objects you want to connect. Select 3D objects as objects, not as faces.
2. Click **Modeler>Surface>Connect**.

   A new object is created that connects the objects you selected. The first object you selected was modified to create the new object and all subsequently selected objects were deleted.

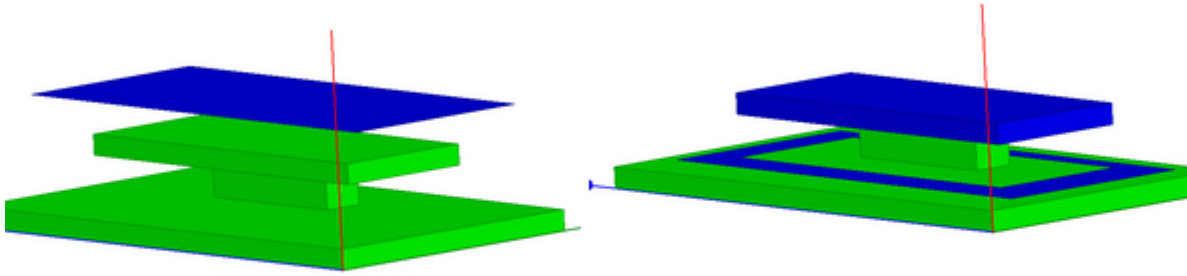
Project Sheet Object

Use the **Modeler>Surface >Project Sheet** command to apply a selected planar sheet object to a suitable 3D object. This permits easy modeling of thin conformal deposits. (The **Wrap Sheet** and **Imprint Projection** commands are not suitable for this use.) For modeling conformal deposits, after using **Project Sheet**, you use Modeler>Surfaces>Thicken Sheet.

To use the **Project Sheet** command you must select one planar sheet and at least one solid body. If you select multiple solids, the sheet is project on all bodies as if they have been united. After the commend executes, a new sheet that lies on the surface of the selected bodies is created. The solid bodies are not changed.

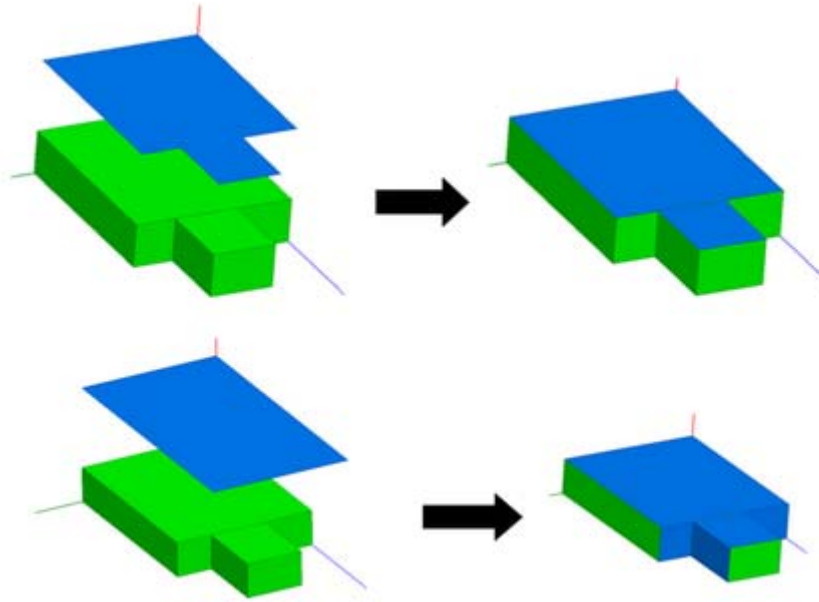


Faces that are hidden from the sheet being projected are not covered, due to the projected sheet.



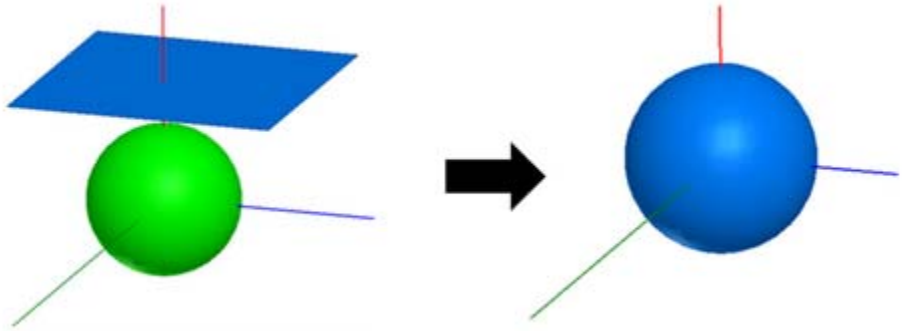
(a) (b)

Only vertical faces (or faces parallel to the projection direction) that lie completely within the projection of the sheet will be covered by the projected sheet. Vertical faces that lie on the boundary of the projection are not covered.



If the sheet cross-section exactly matches the body cross-section, no vertical faces are covered. If the projected sheet exceeds the cross-section of solids, faces that lie with in the sheet projection are covered.

If a surface like a cylinder or sphere lies completely within the sheet projection, the entire surface will be covered.



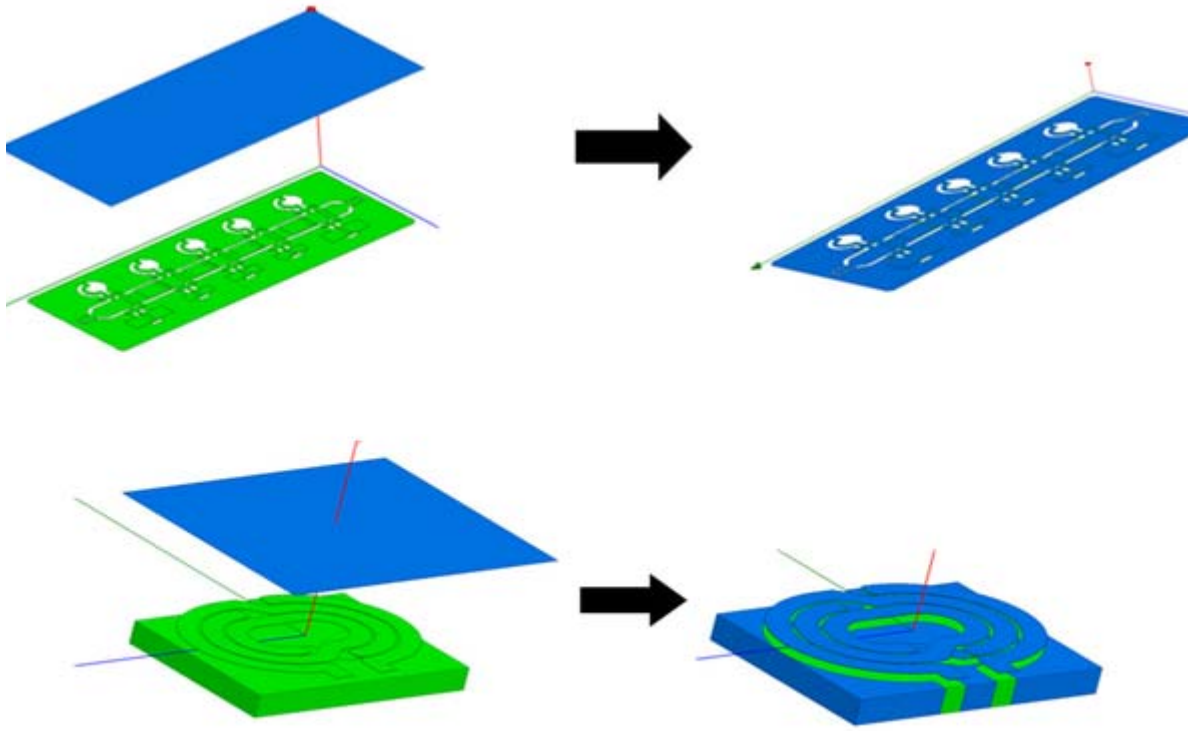
To Project a Sheet:

1. Select a planar sheet object and one or more appropriate solid objects.
2. Click **Modeler>Surface>Project Sheet**.

After you perform **Project Sheet**, the History tree shows the **Project Sheet** command and the create command for the imprinted object. If you select the **Project Sheet** command in the History tree, you can suppress the command via the **Properties** window. If you select the Create <object> icon for the object, you can edit the properties of that object. The changes applied to the object carry over to the imprinting.

For modeling conformal deposits, you must also use Thicken Sheet.

Other examples of Project Sheet on complex models are show in the following figure.



Related Topics

Thicken Sheet

Moving Faces or Edges

You can move the faces of a 3D object in a normal direction using the **Modeler>Surface>Move Faces** commands. Moving object faces enables you to resize, reshape, or relocate an object.

Related Topics

Moving Faces Along the Normal

Moving Faces Along a Vector

Offsetting Objects

Moving Edges Along the Normal

Moving Faces Along the Normal

To move a 3D object's face a specified distance in a direction normal to its original plane, use the **Modeler>Surface>Move Faces>Along Normal** command. The faces that adjoin the original face are extended or shortened along their own planes to meet the new face. Note that the adjoining faces will not be sheared or bent.

This command is useful for extruding faces, resizing holes, and removing rounded corners, as shown below.

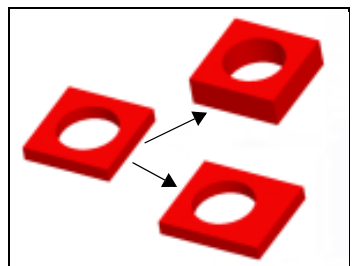
To move an object face in a normal direction:

1. Click **Select Faces** on the shortcut menu.
1. Select the face of the object you want to move.

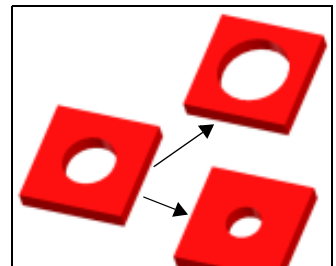
   If you have created a suitable face list, right-click on the list and click **Select Assignment** from the shortcut menu, you can operate on faces on the list.
2. Click **Modeler>Surface>Move Faces**>**Along Normal**.

   The **Move Faces Along Normal** dialog box appears.
3. Type the distance you want to move the object face from its origin.
4. Click **OK**.

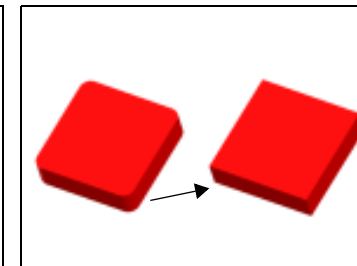
   The face will be moved the distance you specified.



Extruding Faces



Resizing Holes



Removing Rounded Corners

To move every face of an object normal to its surface, use the **Edit>Arrange>Offset** command.

Related Topics

Moving Faces Along a Vector

Offsetting Objects

Moving Edges Along the Normal

Moving Faces Along a Vector

To move the faces of a 3D object a specified distance along a vector use the **Modeler>Surface>Move Faces>Along Vector** command. Each selected face is moved along the vector, normal to its original plane. The faces that adjoin the original face are extended or shortened along their own planes to meet the new face. Note that the adjoining faces will not be sheared or bent.

This command is useful for relocating holes in an object, as shown below.

To move an object face along a vector:

1. Click **Select Faces** on the shortcut menu.
2. Select the face of the object you want to move.

   If you have created a suitable face list, right-click on the list and click **Select Assignment** from

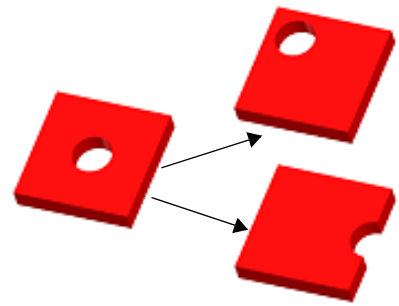
the shortcut menu, you can operate on faces on the list.

3. Click **Modeler>Surface>Move Faces**>**Along Vector**.
4. Specify the vector along which the face will be moved:
   a. Select an arbitrary anchor point in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

      Any point in the drawing region can be selected; however, selecting an anchor point on the object's edge or within the object makes it easier to select the vector.
   b. Select a second point in one of the following ways:
      - Click the point.
      - Type the coordinates of a point relative to the anchor point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

      This point defines the direction and distance from the anchor point to move the face.

   The face is moved along the vector you specified.



Relocating Holes

To move every face of an object normal to its surface, use the **Edit>Arrange>Offset** command.

Related Topics

Moving Faces Along the Normal

Offsetting Objects

Moving Edges Along the Normal

Moving Edges Along Normal

To move a 2D object's edge a specified distance in a direction normal to its original plane, use the **Modeler>Edge>Move Edge** command. The edge is extended or shortened along its own plane. Note that the adjoining faces will not be sheared or bent. The edge can be on a rectangle, an ellipse, a circle, a regular polygon, or an equation based surface.

This command is useful for extending or shrinking faces and resizing holes.

To move an object edge in a normal direction:

1. Click **Select Edge** on the shortcut menu.
2. Select the edge of the object you want to move.
3. Click **Modeler>Edge>Move Edge**.

   The **Move Faces Along Normal** dialog box appears.
4. Type the distance you want to move the object face from its origin.
5. Click **OK**.

   The edge of the object is moved based on the value you specified.

Related Topics

Select Edges.

Moving Faces Along the Normal

Moving Faces Along a Vector

Offsetting Objects

Uniting Objects

To join two or more objects into one object, use the **Modeler>Boolean>Unite** command. The new object has the name, color, boundary, and material assignment of the first object selected. The objects are united at the point of intersection.

1. Select the objects you want to join.
2. Click **Modeler>Boolean>Unite**.  .

   The objects are united.

Note By default, the objects being joined to the first object selected are *not* preserved for later use. If you want to keep a copy of the objects being joined to the first object selected, do one of the following:

- Copy the objects, and then paste them back into the design after uniting them.
- Select **Clone before unite** in the **Modeler Options** dialog box. This option instructs the modeler to always keep a copy of the original objects being joined.

Subtracting Objects

1. Select the object from which you want to subtract other objects.
2. Hold down the **Ctrl** key and select the objects you want to subtract.
3. Click **Modeler**>**Boolean**>**Subtract**  .

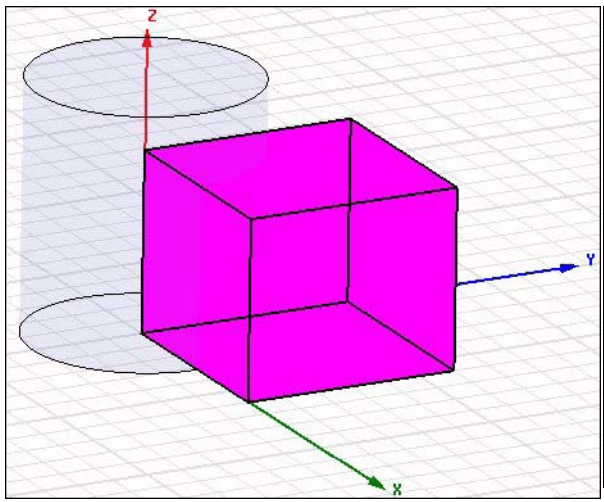
   The **Subtract** dialog box appears.

   Objects listed in the **Tool Parts** list will be subtracted from the object or objects listed in the **Blank Parts** list.
4. Optionally, select an object name in either list and use the left and right arrow buttons to move the object name to the opposite list.

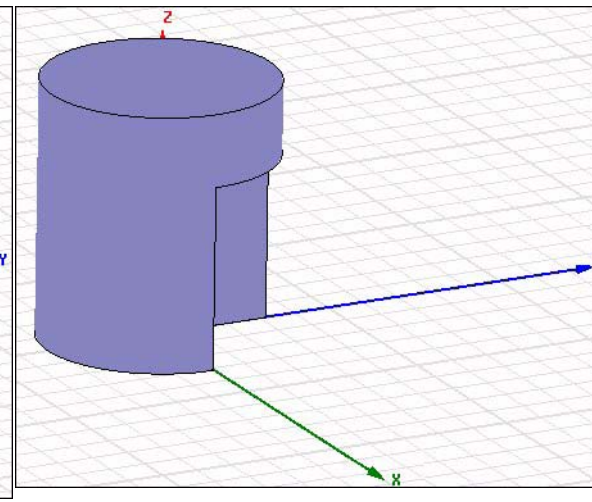
- Alternatively, type the name of object you want to subtract in the empty text box below the **Tool Parts** list, and then type the name of the object from which you want to subtract it in the empty text box below the **Blank Parts** list.

5. Optionally, select **Clone tool objects before subtract**. This instructs HFSS to always keep a copy of the original objects being subtracted.
6. Click **OK**.

   The new object (or objects) retains the name, color, and material of the first object selected.



An intersecting box and cylinder.



A box subtracted from a cylinder.
The cylinder was selected first.

Note By default, the objects being subtracted from the first object selected are *not* preserved for later use. If you want to keep a copy of the objects being subtracted from the first object selected, do one of the following:

- Copy the objects, and then paste them back into the design after subtracting them.
- Select **Clone before subtract** in the **Modeler Options** dialog box. This option instructs HFSS to always keep a copy of the original objects being subtracted.

Creating Objects from Intersections

To create a new object from the intersection of two or more objects, use the **Modeler>Boolean>Intersect** command.

To create an object from an intersection:

1. Select the objects from which you want to take the intersection.

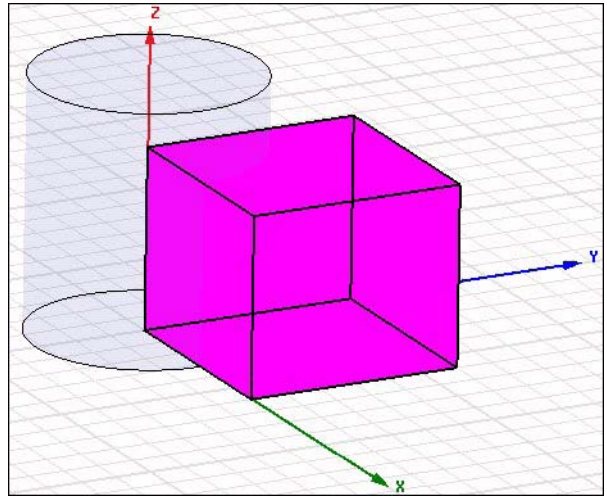
Warning If the objects you selected do not overlap, the result is a null object and both objects vanish.

2. Click **Modeler**>**Boolean**>**Intersect**  .

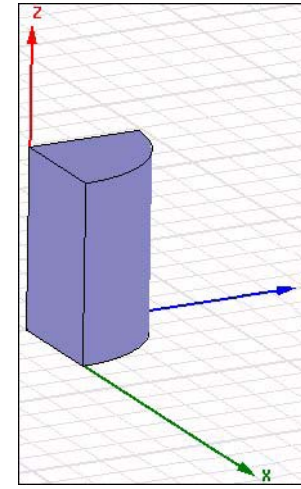
   The original objects vanish, leaving only the new object that was formed from their intersection.

Note By default, the original intersecting objects are *not* preserved for later use. If you want to keep a copy of the objects that intersect the first object selected, do one of the following:

- Copy the objects, and then paste them back into the design after creating the new object from the intersection.
- Select **Clone before intersect** in the **Modeler Options** dialog box. This option instructs the modeler to always keep a copy of the original objects that intersect the first object selected.



An intersecting box and cylinder.



Object formed from the intersection of the box and cylinder.

Creating an Object from a Face

The **Modeler>Surface>Create Object from Face** command copies a selected face, resulting in a new 2D sheet object.

To create a new object from a face:

1. Right-click in the modeler window, and select **Select Faces** on the shortcut menu.
2. Select the object face you want to copy. If you select multiple faces, each becomes a new object.

   Click **Modeler**>**Surface>Create Object From Face**. The face is copied, resulting in a new 2D sheet object.

Hint This command is useful for assigning a boundary to the intersection of two faces. To do this, first select the faces, and then create an object from them using the procedure above. Next, make sure the **Clone before intersect** option is clear in the **Modeler Options** window, and then use the **Modeler>Boolean>Intersect** command to modify the object so that it includes only the intersection of the two faces. Then assign the boundary to the new object.

Related Topics

Assigning a Cross Section and Dimension to a Polyline

Creating an Object from an Edge

Creating an Object from an Edge

The **Modeler>Edge>Create Object From Edge** command copies a selected edge, resulting in a new 2D sheet object.

To create a new object from an edge:

1. Right-click in the modeler window, and select **Select Edges** on the shortcut menu.
2. Select the object edge you want to copy. If you select multiple edges, each becomes a new object.
3. Click **Modeler>Edge**>**Create Object From Edge**.

The edge is copied. The resulting object appears in the history tree as a line object.

Related Topics

Assigning a Cross Section and Dimension to a Polyline

Creating an Object from a Face

Splitting Objects

To an object or objects that lie on the xy, yz, or xz plane, use the **Modeler>Boolean>Split** command.

1. Select the object you want to split. You can select more than one.
2. Click **Modeler>Boolean>Split**  .

   The **Split** dialog box appears.
3. Select **XY**, **YZ**, or **XZ** as the **Split plane**.
4. Select one of the following **Keep fragments** options to specify which object fragments you

want to keep (those on the positive side of the selected plane, those on the negative side of the plane, or all pieces on both sides of the plane):

- **Positive side**
- **Negative side**
- **Both**

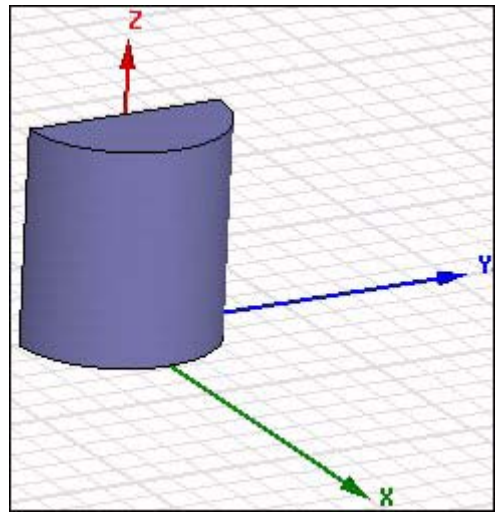
5. Select one of the following **Split objects** options:
   - **Split entire selection**
     Select this option if you do not want to preserve objects that are not crossing the split plane and still part of the selection.
   - **Split objects crossing split plane**
     Select this option so that objects of the selection that do not cross the split plane are preserved after the split operation is performed.

   Note In previous versions of the modeler, the split operation only affected the selected objects that crossed the selected split plane. Other objects were ignored during the operation. In complex geometries, you may want to select everything and perform a split. In some cases, operations are still performed on selected objects that do not cross the split plane (i.e., both parts are retained, yielding the original object and an invalid object). Also, depending on the options specified, some objects not crossing the split may be deleted.

   The **Split objects crossing split plane** option allows you to identify selected objects that do not cross the split plane and ignore them for the operation. For a multiple selection, only those objects that cross the split plane are split; others are kept intact. By design, splits in existing designs from previous versions are not changed.

6. Click **OK**.

The objects are divided as specified.



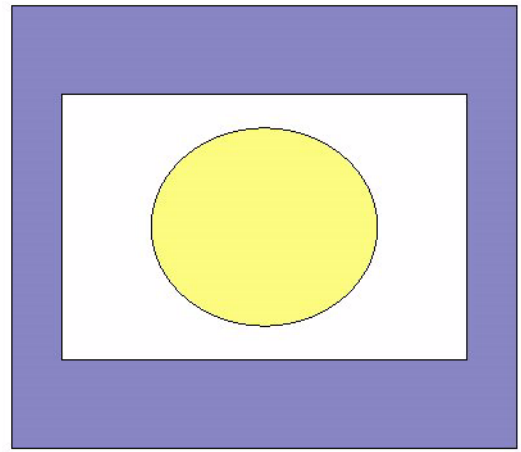
A cylinder split along the positive side of the yz plane.

Separating Bodies

To separate an object with multiple lumps into individual bodies:

1. Select the object you want to separate.
2. Click **Modeler**>**Boolean>Separate Bodies**.

The object is separated.



This figure shows two separate bodies, each with one lump, that were created from one object.

Converting Polyline Segments

A polyline is a single object that includes any combination of straight line, arc line, or spline segments. You can convert a polyline segment from one type to another. The following conversions are supported:

- Straight line segments to arc line or spline segments.
- Arc line segments to straight line or spline segments.
- Spline segments to straight line segments.

To convert polyline segments:

1. In the history tree, locate the polyline that contains the segment you want to convert. Expand this part of the history tree.
2. In the history tree, right-click the polyline segment operation you want to change, and then click **Properties**.

   The Properties dialog appears.
3. In the **Properties** dialog box, click in the **Value** text box of the **Segment Type** row.
4. Select the desired polyline segment type from the pull-down list.

   The polyline segment you selected is changed to the new type.

Note Converting an arc line or spline segment to a straight line segment results in two straight line segments; one segment is created between the start point and midpoint and one segment is created between the midpoint and endpoint.

5. By default, curved surfaces are treated as smooth (True) surfaces. If segmented surfaces are

desired, enter a number of 2 or greater in the **Number of Segments** parameter.

6. Click **OK** to dismiss the properties panel and implement the changes.
   If the changes are not what was expected, undo the change using the **Edit>Undo** command or press **CTRL-Z**.

Related Topics

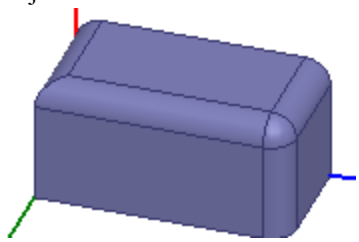
Assigning a Cross Section and Dimension to a Polyline

Creating Segmented Geometry

Surface Approximation

Rounding the Edge of Objects (Fillet Command)

The fillet command rounds the object at the original edges and vertices. This means that the edges and vertices are going to be replaced by new rounded surfaces, so that the original faces of the object reconnect in a smooth manner.



Vertices are only going to be replaced by new rounded surfaces if all the edges connecting to the original vertex are selected; otherwise, the vertex is preserved but moved (if necessary). The edges are replaced by quarter-cylindrical surfaces, of which the radius can be customized (see the **Fillet Radius** property). Vertices are replaced by more complicated new faces. You can control the set-back distance.

The fillet command is disabled if an edge is not selected.

To switch to edge selection mode:

- Right-click the desktop, and select **Select Edges** from the shortcut menu.

To round the edge of an object:

1. Select the edge you want to change.
   This highlights the edge and enables the **Fillet** command.
2. Click **Modeler>Fillet** or click the fillet icon  .
   The **Fillet Properties** dialog is displayed.
3. Enter a value for the **Fillet Radius** in the text field and select units from the drop down menu. The default is millimeters.
4. Enter a value for the **setback distance**.
   The setback distance controls the shape of the vertex. It is the distance of the cross curve from the vertex at the end of the edge. If it is less than the fillet radius it has no effect. You will get an error if it is greater then the length of the edge. Note that the setback feature works only on corners where three or more edges meet and only if all edges meeting at the vertex are selected.

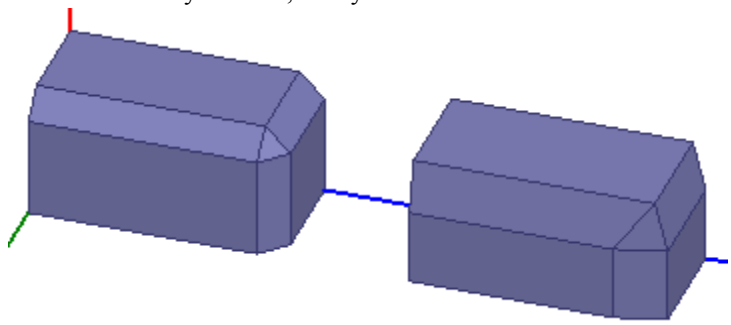
5. Click **OK** to apply the change to the edge.

   The dialog closes and the object is rounded by the radius value relative to the edge you selected.

Flattening the Edge of Objects (Chamfer Command)

The chamfer command flattens the selected edges and vertices of the object. This means that the selected edges and vertices are going to be replaced by new flat surfaces, so that the original faces of the object reconnect through the newly introduced flat surfaces. Vertices are only going to be replaced by new flat surfaces if all the edges connecting to the original vertex are selected; otherwise, the vertex is preserved but moved (if necessary).

Chamfers can be symmetric, or asymmetric



The chamfer command is disabled if an edge is not selected.

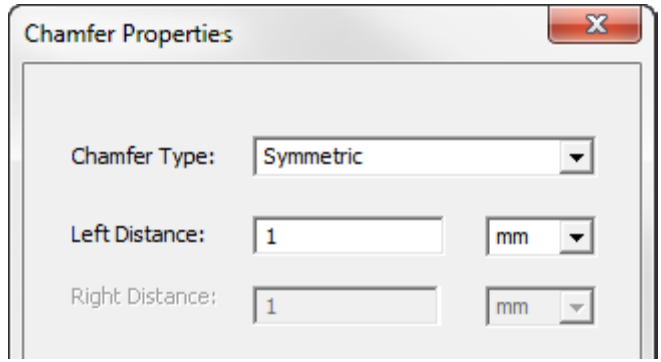
To switch to edge selection mode:

- Right-click the desktop, and select **Select Edges** from the shortcut menu.

To flatten an object's edge for a symmetric chamfer:

1. Select the edge (or edges) you want to change.
   The edge(s) is highlighted, and the **Chamfer** command is enabled.
2. Click **Modeler>Chamfer** or click the **Chamfer the selected edges** icon on the **Modeler Blending** toolbar.

   The **Chamfer Properties** dialog box appears.

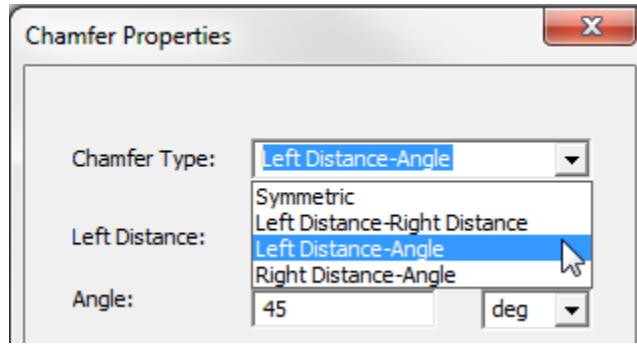


3. Type a value in the **Left Distance** text box, and select the units from the pull-down list.
4. Click **OK** to apply the change to the edge.

The **Chamfer Properties** dialog box closes, and the object is flattened by the radius value relative to the edge you selected.

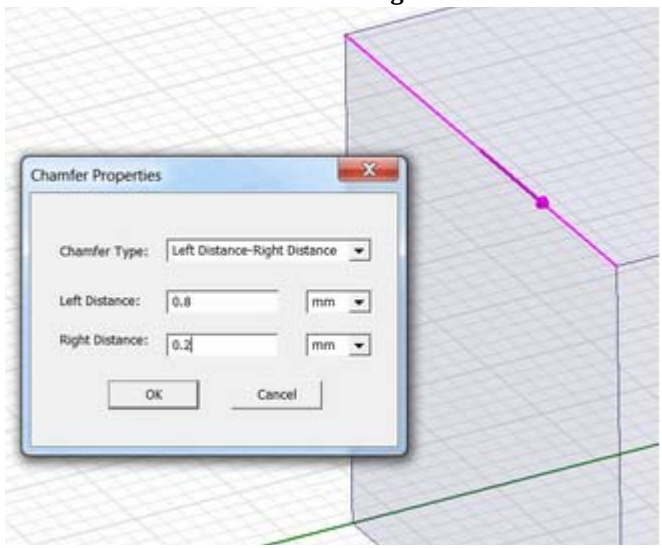
To flatten object's edge for an asymmetric chamfer:

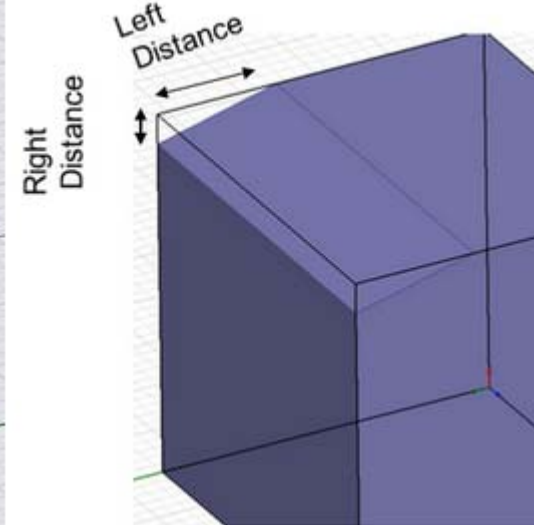
1. Select one or more edges and click **Modeler>Chamfer** to open the **Chamfer Properties** dialog.
2. Use the Chamfer type drop down menu to select the type:



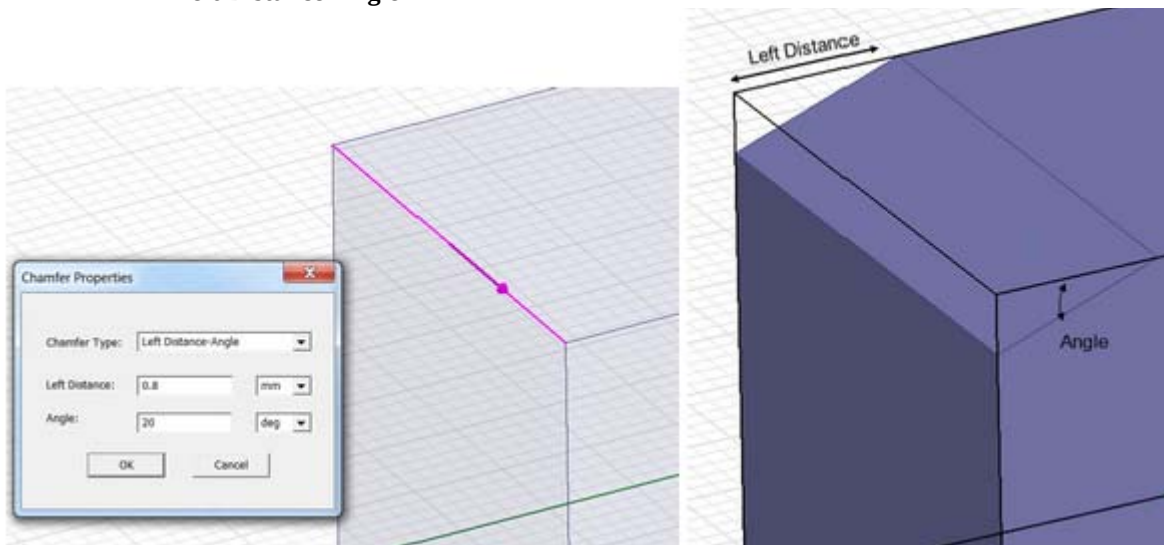
You control an asymmetric chamfer by selecting a type that defines the chamfer asymmetry as Left-Distance Right Distance, as Left Distance-Angle, or as Right Distance Angle. Notice that red directional arrows on the selected edges provide the direction against which left distance and right distance are determined.

Left Distance- Right Distance

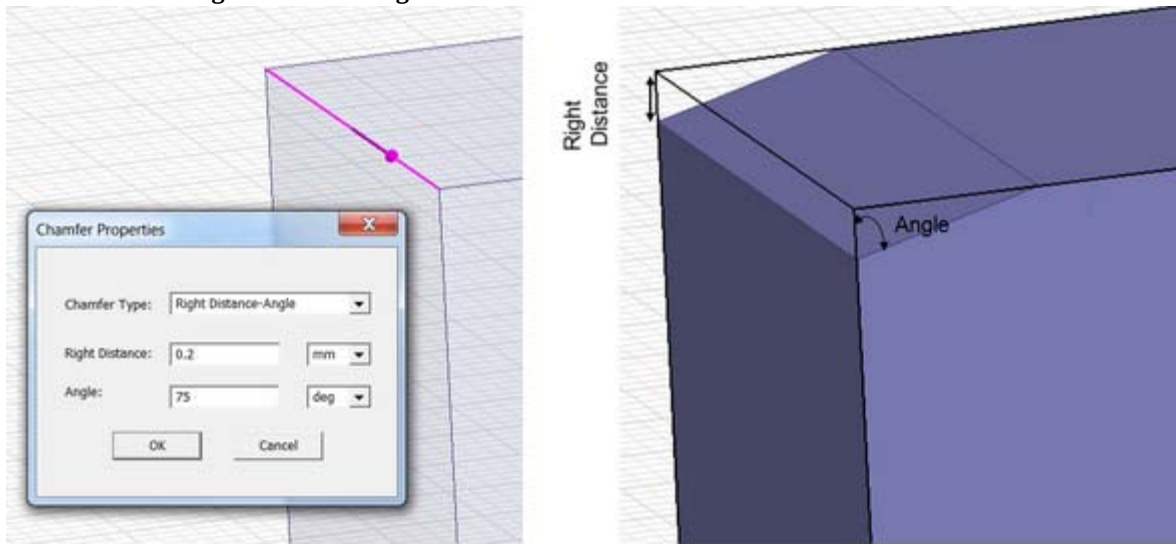




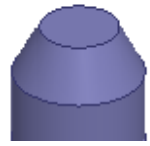
Left Distance-Angle



Right Distance-Angle



Only symmetric and left distance-right distance chamfers are supported for edges formed by a curved surface.



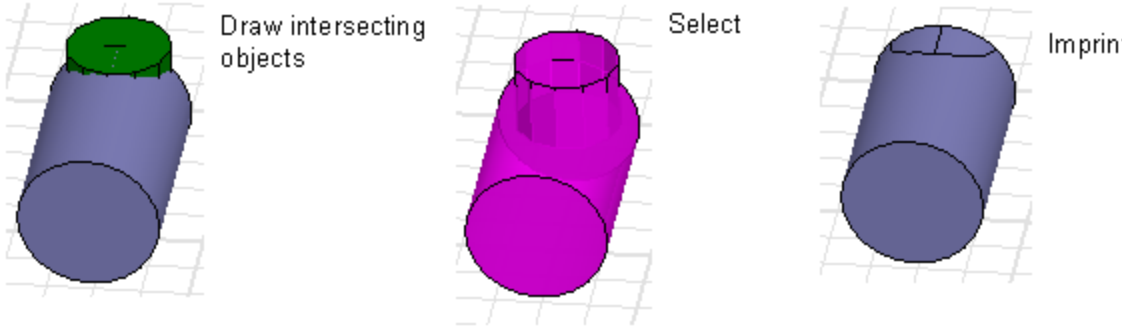
Angle-distance chamfers are not supported in such cases and do not appear on the Chamfer

Type drop down menu.

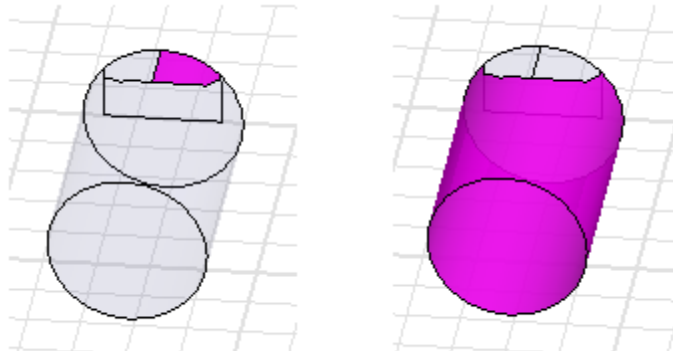
3. Click OK to apply the chamfer to the selected edges.

Imprinting an Object

The **Boolean>Imprint** command lets you imprint the geometry of one object upon another. For example, you could draw a polyhedron intersecting a cylinder, and then imprint the intersecting lines on the cylinder.



You can select the faces of the imprinted surface separately and assign properties as needed.

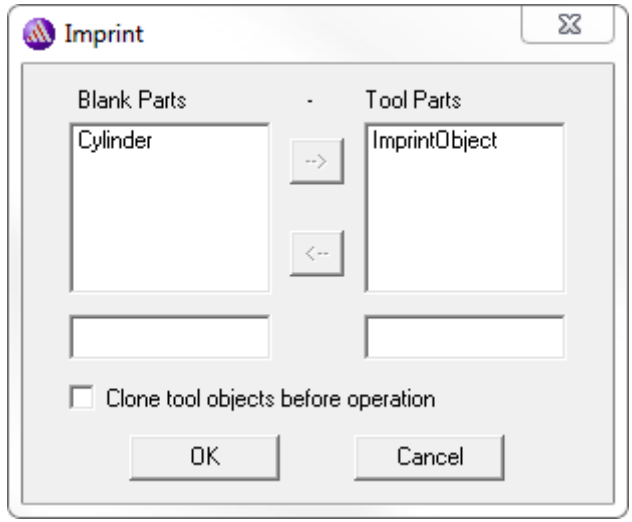


To imprint one object with another:

1. Select the intersecting objects.
2. Click **Modeler>Boolean>Imprint...**

   This displays the Imprint dialog in which you designate which objects are the Blank Parts, and which the Tool Parts. If necessary, you can select the objects in lists, and use the arrow keys to

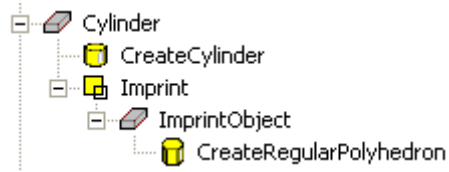
move them. If desired, you can clone the tool objects before the imprint operation.



3. Click OK.

   This closes the dialog and performs the boolean imprinting.

After you perform the imprinting, the History tree retains the Imprint Object command and the create command for the imprinted object



If you select the Imprint command in the History tree, you can suppress the command via the Properties window. If you select the Create <*object*> icon for the object, you can edit the properties of that object. The changes applied to the object carry over to the imprinting.

You can also use **Tools>Options >Modeler Options** to enable the Automatically imprint wrapped sheets feature to perform an **Imprint Object** command after a **Modeler>Surface>Wrap** command.

Related Topics

View and Edit Commands on History Tree Objects

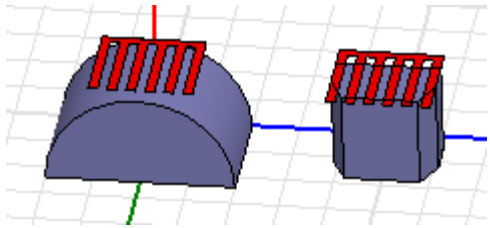
Imprint Projection Commands

Wrap Command

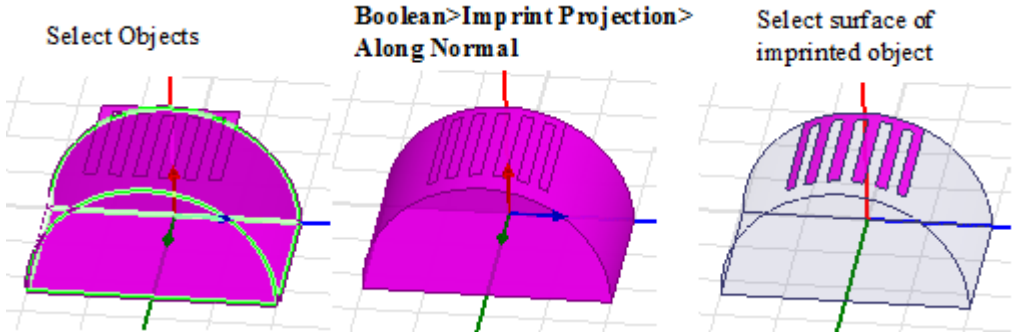
Modeler Options: Operation Tab

Imprint Projection commands

The **Boolean>Imprint Projection** commands lets you project the form of one object to another surface. The receiving surface can be curved or faceted.



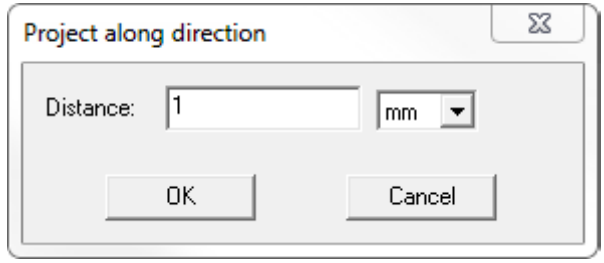
If the surface is curved, the dimensions of the projection will be affected.You can select the faces of the imprinted object separately, and edit properties as needed.



If projected shape extends beyond the face of the receiving object, the shape wraps.

1. Select the intersecting objects.
2. Click **Modeler>Boolean>Imprint Projection>Along Normal** or **Modeler>Boolean>Imprint Projection>Along Direction...**

   If you select **Along Normal**, the projection occurs along the normal. If you select Along Direction, you need to specify two points that describe the direction. Once you have defined a line by clicking two points, you see a dialog for specifying the distance for the projection.



3. Specify a distance and select units from the drop down menu and click OK.

   This closes the dialog and performs the boolean imprinting.

After you perform the imprinting, the History tree shows the Imprint Object command and the create command for the imprinted object

If you select the Imprint Projection command in the History tree, you can suppress the command via the Properties window. If you select the Create <object> icon for the object, you can edit the properties of that object. The changes applied to the object carry over to the imprinting.

Related Topics

View and Edit Commands on History Tree Objects

Imprinting an Object

Purge History

Each object is a sequence of modeler-based operations. The history for each object is shown under its name in the model tree. You can use the **Purge History** command to remove the history of operations while not affecting the geometry itself. This is useful when you wish to perform healing operations on the object. If there is an object for which you want to keep the history, you should make a copy of the object for that purpose before purging.

To purge the history:

1. Select the object.
2. Select **Modeler>Purge History**. The history for the model is purged, and the context for the **Undo** and **Redo** commands is updated.

Related Topics

Working with the History Tree

Generate History

Generate History to Reproduce Portions of Model

If a polyline object (line, spline, or arc), circle, or ellipse is imported or history was previously purged, you can click on the polyline object and select **Generate History** to reproduce the individual line segments used to create the polyline in the model history tree.

To reproduce the line segments in the model history tree:

1. Select the polyline object.

Click **Modeler>Generate History**.

Related Topics

Purge History

Draw Polyline

Selecting Items in the 3D Modeler Window

To modify or learn more about an item's properties, you must first select it. All commands you choose while an item is selected are performed on or in reference to the selected item.

What selection mode do you want to use?

- Select Submodels
- Select Objects.
- Select Faces.
- Select Edges.
- Select Vertices.
- Select Multi (a mode for selecting objects, faces, edges or vertices)
- Coordinates in the drawing space.
- Select By Area
- Select by Variable
- Selecting Objects by Name
- Select by History Tree Group
- Select by intersection error message
- Selecting the Face or Object Behind

Note If selected objects do not display correctly, for some graphics cards, you can improve performance by setting **NVIDIA Control Panel>3D Settings>Manage 3D Settings Global Settings>Global Presets: Workstation App - Dynamic Streaming**

Selecting Objects

By default, the modeler is in *object selection mode*. Simply click an object in the view window or an object name in the history tree and it will be selected. All other objects become relatively transparent.

When the mouse hovers over an object in the view window, that object is highlighted, which indicates that it will be selected when you click. Selected objects become the color specified under the **Display** tab of the **Modeler Options** dialog box.

Tooltips, as you hover the cursor over an entity, indicate the type/ID of entity (object name in the case of objects, Face_id in the case of faces, and so on). This feature helps you distinguish between face-of-sheet-object pick versus sheet-object pick.

If the modeler is not currently in object selection mode, you can switch to it using one of the following methods:

- Press the shortcut key **O**.
- Right-click in the view window, and then click **Select Objects**.
- Click **Edit** > **Select**>**Objects**.

- Select **Object** from the pull-down list in the **3D Modeler Selection** toolbar.

Related Topics

Selecting Several Objects

Selecting Objects by Name

Selecting All Faces of an Object

Creating an Object List

Selecting the Face or Object Behind

Select Edges.

Select Vertices.

Select Multi (a mode for selecting objects, faces, edges or vertices)

Select By Area

Selecting Objects and Surfaces that Lie Inside Other Objects

Clearing a Selection

Selecting Several Objects

1. If you are selecting objects in the Modeler window make sure that the modeler is in object selection mode by pressing the shortcut key **O**. You can always select objects in the History tree.
2. Select several objects in one of the following ways:
   - Hold down **Ctrl,** and click the objects in the view window that you want to select.
   - Hold down **Ctrl**, and click the object names in the history tree that you want to select.
   - In the History tree, select a range of objects by first clicking one object to select it, and then Shift-click to extend the selection of visible items.
   - In the History tree, under Lists, select AllObjects. This is an automatically created list that lets you selects all object.
   - Click **Edit>Select All** to select all objects that were drawn in the active view window, including objects that are not currently visible.
   - Press **CTRL**+**A** or click **Edit>Select All Visible** to select all objects that are visible in the active view window.

   Selected objects become the color that is specified for selected objects under the **Display** tab of the **Modeler Options** dialog box. Use **Tools>Options>Modeler Options** to display the dialog and set the default color. By default, the selected objects are opaque and all other objects become relatively transparent. The settings for the relative opacity and transparency of selected and non-selected objects appear in the **3D UI Options** dialog box. Use **View>Options** to display the **3D UI Options** dialog.

To deselect all objects, do one of the following:

- Click **Edit>Deselect All**.

  Press **Ctrl+Shift+A**.

Related Topics

Selecting Objects by Name

Selecting All Faces of an Object

Creating an Object List

Selecting the Face or Object Behind

Select Edges.

Select Vertices.

Select Multi (a mode for selecting objects, faces, edges or vertices)

Select By Area

Selecting All Objects in a History Tree Group

Selecting Objects by Name

1. Make sure that the modeler is in object selection mode by pressing the shortcut key **O**.
2. Click **Edit>Select>By Name B[illegible]me**  or in the toolbar, select Object from the drop-down menu to the right of the  icon, and click the icon.

   The **Select Object** dialog box appears.
3. In the **Name** list, click the name of the object you want to select. Use the **Ctrl** key to select more than one.
   - Alternatively, type the name of an object you want to select in the empty text box.
4. Click **OK**.

   The object is selected.

Related Topics

Selecting Several Objects

Selecting All Faces of an Object

Creating an Object List

Selecting the Face or Object Behind

Select Edges.

Select Vertices.

Select Multi (a mode for selecting objects, faces, edges or vertices)

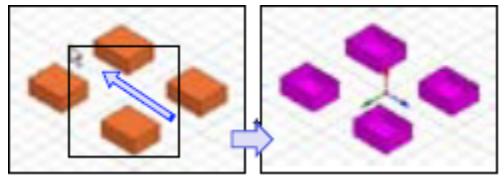
Select By Area

Select By Area

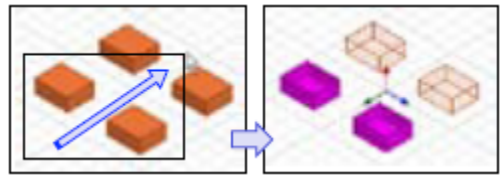
You can select items by area in the Modeler window clicking and rubber-band dragging around objects. Rubber-band selection works differently depending on the drag direction.

- From right to left: Selects all the items that are wholly or partly enclosed within the rubber

band.



- From left to right: Selects all the items wholly enclosed within the rubber band.

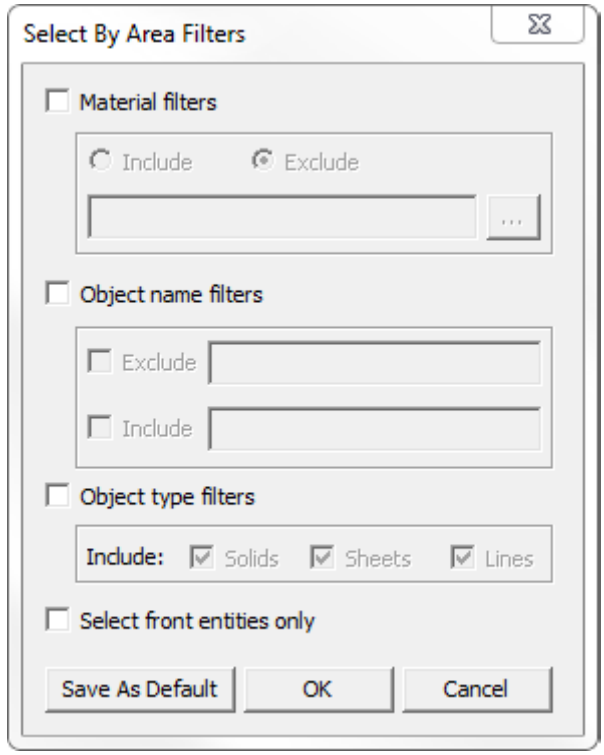


Select By Area works with Selection mode for Objects, Faces, Edges, and Vertices, but not for Select Multi. By default, only items with external surfaces are selected.

However you can control which objects to include or exclude from area selection based on material, object names, or object types.

To do this:

1. Click **Edit>Select By Area Filter** to display this dialog:



2. Check **Material filters** to enable the **Include** and **Exclude** radio buttons. Use the text field to specify filters by name, or use the ellipsis [...] button to display the **Materials** manager for selections.
3. Check **Object name filters** to enable the **Exclude** and **Include** check boxes, and text fields in which you can specify object names.
4. Check **Object type filters** to enable the check boxes for including **Solids**, **Sheets**, and/or lines.
5. Check **Hide unfiltered** objects to make unfiltered objects transparent after selection.
6. If you click **Save As Default**, the settings persist for the project until you change the settings and **Save as Default** again.
7. Click **OK** to close the dialog.

Now when you left-click and drag around an area, those objects which meet the filter criteria are highlighted in the Modeler window, and those objects are shown as selected in the History tree.

Related Topics

Selecting Items in the 3D Modeler Window

Selecting Several Objects

Selecting Objects by Name

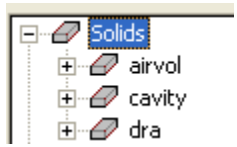
Selecting All Faces of an Object

Creating an Object List

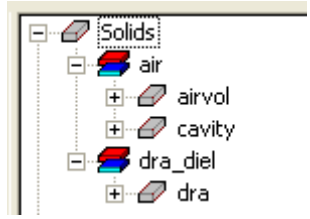
Selecting All Objects in a History Tree Group

The history tree groups objects in several categories. Right click on the group for solids, sheets, lines, non-model objects, or unclassified objects to see a shortcut menu command that lets you **Select All** members of that category.

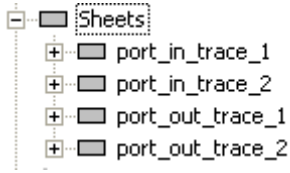
Under each group, you can also **Select All** for objects of same the material or parts of same assignment at once. When you right-click on a solid grouping, the menu lets you check whether to group objects by assignment. For example, this history tree has Solids grouped by object.



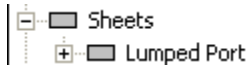
This history tree has Group Objects by Material checked.



This history tree groups Sheet objects by object.



This history tree has Group Sheets by Assignment checked.



Selected objects are highlighted in the modeler window view area.

Related Topics

Selecting Several Objects

Selecting Objects by Name

Select By Area

Selecting All Faces of an Object

Creating an Object List
Setting the Default Color and Transparency of Selected Objects
Setting the Default Color of Highlighted Objects
Working with the History Tree

Setting the Default Color and Transparency of Selected Objects

To set the color of objects when they are selected:

1. Click **Tools>Options>Modeler Options**.

   The **Modeler Options** dialog box appears.
2. Click **Select** on the **Default color** pull-down list.
3. Click the color button beside the **Default color** pull-down list.

   The **Color** palette appears.
4. Select a color from the **Color** palette, and then click **OK**.

   Any objects you select after this point will temporarily become the default color you selected.

By default, the modeler shows selected objects as nearly opaque and shows non-selected objects as nearly transparent. This features helps you distinguish between selected and non-selected objects.

To set the transparency of selected and non-selected objects:

1. Click **View>Options**.

   The **3D UI Options** dialog appears. The **When there is a selection** region contains check-boxes for setting the transparency for selected and non-selected objects.

   Click the checkbox for the value you want to change.

   This enables the value field. The default transparency for selected objects is 0.1, which makes them almost opaque. The default transparency for non-selected objects is 0.9, which makes them highly transparent.
2. Enter a new value, and click **OK** to apply the new transparency values.

Setting the Default Color of Highlighted Objects

1. Click **Tools**>**Options>Modeler Options**.

   The **Modeler Options** dialog box appears.
2. Click the **Display** tab.
3. Click **Highlight** on the **Default color** pull-down list.
4. Click the color button beside the **Default color** pull-down list.

   The **Color** palette appears.
5. Select a color from the **Color** palette, and then click **OK** to return to the **Modeler Options** dialog box.
6. Click **OK**.

   Outlines of all object you hover over temporarily become the default color you specified above.

Creating an Object List

Create an object list when you want to define a list of objects. Creating an object list is a convenient way to identify and select a group of objects for a field plot or calculation. Objects in a list can still be treated as separate objects. The same object can be included in several different lists.

To create an object list:

1. If you are selecting in the Modeler window, make sure that the modeler is in object selection mode by pressing the shortcut key **O**. This is not necessary for selecting in the History tree.
2. Select the objects you want to include in the list.
3. Click **Modeler>List>Create>Object List**.

   The object list is created with the default name Objectlist*n*. It is listed in the history tree under **Lists**. Selecting an object list displays the properties of that list in the Properties window. One of the properties is a list of objects contained in the list.

   To rename the Object list, edit the Name property in the **Properties** window for the list. Object lists are sorted in alphanumeric order.

The object list is treated as one volume when you plot and perform fields calculations. It will be listed in the **Geometry** window of the Fields Calculator when you select **Volume**.

There is an automatically created list called AllObjects. Selecting it selects all objects. If a list contains mixed types of geometry, for example, volume and sheet objects, the volume calculation only uses the geometry of the highest dimension in plots or integral, and so forth.

Example: To plot the E-field on a surface formed by the intersection of the xy-plane and several objects, first define a list of these objects. Then, when plotting fields, select the object list name from the **Geometry** window of the Fields Calculator. Fields will be plotted only at the intersection of the plane and the objects in the list.

Related Topics

Reassigning Objects to Another Object List

Using or Viewing List Objects

Reassigning Objects to Another Object List

You can assign objects after you have created object lists. Creating an object list is a convenient way to identify and select a group of objects for a field plot or calculation. Objects in a list can still be treated as separate objects. The same object can be included in several different lists.

To reassign objects to an existing object list:

1. If you are selecting in the Modeler window, make sure that the modeler is in object selection mode by pressing the shortcut key **O**. This is not necessary for selecting in the History tree.
2. Select the objects you want to reassign.
3. Click **Modeler>List>Reassign**

   A dialog with the existing object lists is displayed. (They appear in the history tree under **Lists**.)
4. Select the list to which you want to assign the selected object(s) and click OK.

The object is reassigned to the selected list, replacing previous list members. The Objects Property in for the List shows the objects contained in the list.

The object list will be treated as one volume when you are plotting and performing fields calculations. It will be listed in the **Geometry** window of the Fields Calculator, when you select **Volume**.

Related Topics

Creating an Object List

Using or Viewing List Objects

Using or Viewing List Objects

To view the objects included in an Object list:

1. In the model tree, expand the **Lists** tree.
2. Right-click the list you want to select, and click **Select Assignment**.
   The objects that are included in that list are highlighted in the modeler, and the properties appear in the desktop.

To view the properties of the object list (including a list of the objects included):

1. In the model history tree, expand the **Lists** tree.
2. Under **Lists**, right-click the list object you want to view, and click **Properties**.
   The **Properties** window appears for that object list. The objects included are listed in the **Objects** row.
3. Click **OK** or **Cancel** to close the **Properties** window.

To use an object from a list in another operation:

1. In the model tree, expand the **Lists** tree.
2. Right-click the list you want to select, and click **Select Assignment**.
   The objects that are included in that list are highlighted in the modeler, and the properties appear in the desktop.
3. Select any other objects you want to use in the operation.
4. Complete the operation.

For example, you could select an object list and another object, and then specify one of the boolean commands (such as unite or subtract).

Selecting Faces

If the modeler is in face selection mode, click an object face in the view window to select it. To select multiple faces, hold the **CTRL** key as you click the faces. You also have the option to create face lists, which define a list of object faces, or you can make face selections from a Face ID list in the By Face dialog.

Switch to face selection mode using one of the following methods:

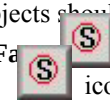
- Press the shortcut key **F**.
- Right-click in the view window, and then click **Select Faces**.

- Click **Edit>Select>Faces**.
- Select **Face** from the pull-down list to the right of the select objects icon  in the **3D Modeler Selection** toolbar.

You can also select faces in the Select Multi mode.

When the mouse hovers over a face in the view window, that face is highlighted, which indicates that it will be selected when you click. Selected faces become the color specified under the **Display** tab of the **Modeler Options** dialog box. All other objects and faces become relatively transparent.

You can also use the **By Face** dialog to select from a list of faces associated with an object:

1. To use the dialog, no objects should be selected to start.
2. Click **Edit>Select>By Fa**  or in the toolbar, select Face or Multi from the drop-down menu to the right of the icon, and click the icon.

   This displays the **By Face** dialog. This contains a list of the available objects.
3. Select an object in the Object Name list.

   The Face ID list is then populated with the faces in that object.
4. Selecting a face ID from the list highlights the face in the 3D window. Use Ctrl-click to select additional faces, or shift-click to select a range of faces.

Related Topics

Selecting All Faces of an Object

Selecting the Face or Object Behind

Selecting Faces by Name

Selecting Faces by Plane

Creating a Face List

Face Selection Toolbar Icons

Select Edges.

Select Vertices.

Select Multi (a mode for selecting objects, faces, edges or vertices)

Selecting All Faces of an Object

1. Optionally, select the object (or objects, faces, edges or vertices) with the faces you want to select.
2. Switch to face selection mode by pressing the shortcut key **F**.
3. If an object is not selected, click a face on the object of interest.
4. Click **Edit>Select>All Object Faces**.
   - Alternatively, right-click in the view window, and then click **All Object Faces** on the shortcut menu.
   - As another alternative, select use the face selection toolbar icons.

   All the faces of the object are selected. If you selected multiple objects, all faces of those

objects are selected.

Related Topics

Selecting Faces

Selecting the Face or Object Behind

Creating a Face List

Face Selection Toolbar Icons

Selecting Faces by Name

1. Make sure that the modeler is in face selection mode by pressing the shortcut key **F**.
2. Click **Edit**>**Select**>**By Name** or in the toolbar, select Face from the dropdown menu to the right of the object selection  icon and click the icon.

   The **Select Face** dialog box appears.
3. In the **Object name** list, click the name of the object with the face you want to select.

   The object's faces are listed in the **Face ID** column.
4. Click the face you want to select in the **Face ID** column. You can select more than one.

   The face is selected in the view window.
5. Click **OK**.

Related Topics

Selecting Faces

Creating a Face List

Selecting Faces by Plane

To select a face that is aligned with a global plane, use one of the following two methods.

1. Make sure that the modeler is in face selection mode by pressing the shortcut key **F**.
2. In the History Tree, expand the **Planes** icon. Left-click on a plane (Global:XY, Global:YZ, or Global:XZ) to display the selected global plane.
3. Click **Edit**>**Select**>**Faces on Plane**.

   The **selected faces** are highlighted.

Alternative method:

1. In the History Tree, expand the **Planes** icon.
2. **Right-click on a plane** (Global:XY, Global:YZ, or Global:XZ) to select the global plane and display a pull-down menu.
3. On the **pulldown** menu, click **Faces on Plane**.

   The **selected faces** are highlighted.

Related Topics

Selecting Faces

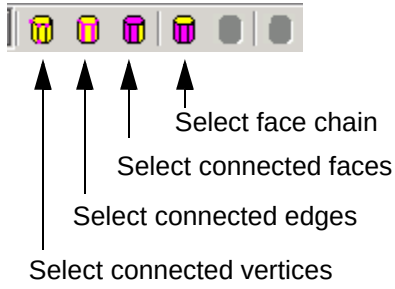
Creating a Face List

Select Multi (a mode for selecting objects, faces, edges or vertices)

Face Selection Toolbar Icons

While working on analyzing complex objects, it is sometimes useful to examine faces, edges and vertices. In particular it is useful to find the connected faces for a face or edge or vertex, connected edges for a face/edge/vertex and connected vertices for a face/edge/vertex. The additional selection modes are available under **Edit->Select** and via the toolbar icons.

Selecting an object face enables the face selection icons in the toolbar.



You can use these icons to modify the selection:

- Select face chain selects faces that touch each other. It allows faces that are part of a "protrusion" to be selected.
- Select connected faces selects faces connected to the current selection.
- Select connected edges selects the edges of the selected face or faces.
- Select vertices selects the vertices of the selected face or faces.

Related Topics

Selecting All Faces of an Object

Selecting the Face or Object Behind

Selecting Faces by Name

Selecting Faces by Plane

Creating a Face List

Select Edges.

Select Vertices.

Select Multi (a mode for selecting objects, faces, edges or vertices)

Creating a Face List

Create a face list when you want to define a list of object faces. Creating a face list is a convenient way to identify and select a specific set of surfaces for a field plot or calculation. The same face can be included in several different lists.

To create a face list:

1. Make sure that the modeler is in face selection mode by pressing the shortcut key **F**.
2. Select the object faces you want to include in the face list.
3. Click **Modeler>List>Create>Face List**.

   The face list is created. It is listed in the history tree under **Lists**. The default name is Facelist*n*. The lists appear in alphanumeric order. To change the name of a face list (for example, to a name describing the listed faces as ports or boundaries), select the list in the History Tree and **Edit Properties**. Editing the Name property changes the name. If necessary, the list order in the History tree changes for the new name.

The face list will be treated as one selection of surfaces when you are plotting and performing fields calculations. The face list will be listed in the **Geometry** window of the Fields Calculator, when you select **Surface**.

If you right-click on an existing face list and click **Select Assignment** from the shortcut menu, you can make boundary assignments or execute the **Move Faces** geometry operations for faces on the list.

Related Topics

Selecting Faces

Radiated Fields Post Processing

Setting Up a Near-Field Sphere

Setting up a Far-Field Infinite Sphere

Selecting Edges

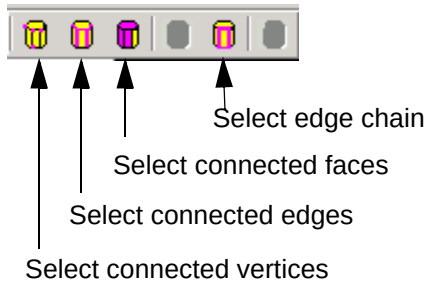
If the modeler is in edge selection mode, simply click an object's edge in the view window and it will be selected. To select multiple edges, hold the **CTRL** key as you click the edges.

When the mouse hovers over an edge in the view window, that edge is highlighted, which indicates that it will be selected when you click. Selected edges become the color specified under the **Display** tab of the **Modeler Options** dialog box. All other objects become relatively transparent.

Switch to edge selection mode using one of the following methods:

- From the menu bar, click **Edit**>**Select**>**Edges**
- Press the "**E**" key to enter edge selection mode.
- Select **Edge** from the pull-down list in the **3D Modeler Selection** toolbar.

Selecting an edge enables the following toolbar icons.



You can use these icons to modify the current selection.

- Select edge chain selects the edges that touch the selected edge.
- Select connected faces selects faces touching to the current selection.
- Select connected edges selects the edges that touch the current selection.
- Select vertices selects the vertices of the selected edge or edges.

Related Topics

Moving Edges Along the Normal

Selecting All Faces of an Object

Selecting the Face or Object Behind

Selecting Faces by Name

Selecting Faces by Plane

Creating a Face List

Select Edges

Select Vertices

Select Multi (a mode for selecting objects, faces, edges or vertices)

Selecting All Edges of an Object or Face

This option allows you to select all edges of an object or face after first selecting the object or face.

1. Select the object or face with the edges you want to select.

   You may also select a single edge of the object or face.

2. Click **Edit>Select>All Object Edges** or E**dit>Select>All Face Edges**.

   - Alternatively, right-click in the view window, and then click **All Object Edges** or **All Face Edges** on the shortcut menu.

   All the edges of the object or face are selected. If you selected multiple objects, all edges of those objects are selected.

Related Topics

Selecting Faces

Selecting the Face or Object Behind

Creating a Face List

Face Selection Toolbar Icons

Selecting Vertices

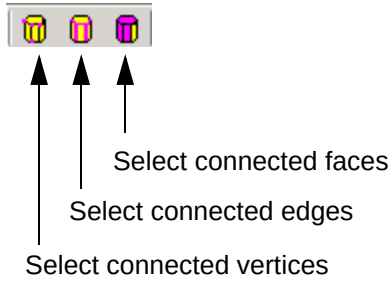
If the modeler is in vertex selection mode, simply click an object's vertex in the view window and it will be selected. To select multiple vertices, hold the **CTRL** key as you click the vertices.

When the mouse hovers over a vertex in the view window, that vertex is highlighted, which indicates that it will be selected when you click. Selected vertices become the color specified under the **Display** tab of the **Modeler Options** dialog box. All other objects become relatively transparent.

Switch to vertex selection mode using one of the following methods:

- Click **Edit** > **Select**>**Vertices**.
- Press the "**V**" key to enter vertex selection mode.
- Select **Vertex** from the pull-down list in the **3D Modeler Selection** toolbar.

Selecting a vertex enables the following selection icons.



You can use these icons to modify the current selection.

- Select connected faces selects faces touching to the current selection.
- Select connected edges selects the edges that touch the current selection.
- Select vertices selects the vertices of edges that touch the current selection.

Related Topics

Selecting All Faces of an Object

Selecting the Face or Object Behind

Selecting Faces by Name

Selecting Faces by Plane

Creating a Face List

Select Edges

Select Vertices

Select Multi (a mode for selecting objects, faces, edges or vertices)

Selecting Multi (a Mode for Selecting Objects, Faces, Vertices or Edges)

The Select Multi mode permits you to select objects, faces, vertices, or edges, depending on where you click. This very useful in conjunction with Measure Mode, for measuring the distances between different entities. Enter Select Multi mode by one of the following methods:

- Press the shortcut key **M**.
- Right-click in the view window, and then click **Select Multi**.
- Click **Edit**>**Select**>**Multi**.
- Select **Multi** from the pull-down list in the **3D Modeler Selection** toolbar.

With Multi mode active:

- To select a vertex, click near a vertex, within 10 pixel radius.
- To select an edge, click near an edge (and 10 pixels away from vertex).
- To select an object, click little farther from edge, between 10 and 20 pixels.
- To select a face, click anywhere else on the interior of face.

Tooltips, as you hover the cursor over an entity, indicate the type/ID of entity (object name in the case of objects, Face_id in the case of faces, and so on). This feature helps you distinguish between face-of-sheet-object pick versus sheet-object pick.

By holding down the Ctrl key, you can make multiple selections.

Related Topics

Controlling the Selection in Multi Mode

Controlling the Selection in Multi Mode

You can control the behavior of this mode by clicking **Edit>Select Multi Mode Settings**. This displays a dialog with check boxes for Object, Face, Edge, and Vertex. Unchecking a box cancels the selection behavior for that category.

You can also control the behavior of this mode by clicking the icons for Object, Face, Edge, and Vertex to the right of the Multi mode selection menu. To add the Multi Mode selection menu and icons to the toolbar:

1. Select **Tools>Customize**.

   This displays the **Customize** dialog with the Toolbars tab selected.
2. Select 3D Modeler Selection mode from the toolbars list by checking it.

   This adds the Mode selection menu and icons to the toolbar.

   You can also add the Mode selection menus from the Commands tab by selecting 3D Modeler Selection from the Category list, and dragging the icons to the toolbar.
3. When Multi is selected as the mode, you can enable or disable Object, Face, Edge, or Vertex selection by clicking the associated icon.

Related Topics

Selecting All Faces of an Object

Selecting the Face or Object Behind

Selecting Faces by Name

Selecting Faces by Plane

Creating a Face List

Select Edges

Select Vertices

Selecting the Face or Object Behind

Clearing a Selection

Measure Modes

Clearing a Selection

To clear an object, face, edge, or vertex selection, do one of the following:

- Click the view window at a point where an object does not exist.
- To clear an object selection, click a point away from the object name in the history tree.
- Click **Edit>Deselect All**.
- Press **Shift+Ctrl+A**.

  The items are no longer selected.

Selecting the Face, Edge, Vertex, or Object Behind

To select the face, edge, vertex, or object behind another selected face, edge, vertex, or object, do one of the following:

- Click **Edit>Select>Next Behind**.
- Right-click in the view window and click **Next Behind**.

  When there are multiple faces behind, the one selected is relatively close to where you right-click.
- Press the shortcut key **B**.

  When there are multiple faces behind, the one selected relatively close to the cursor.
- Press **Ctrl+B**.

This option is useful when you are trying to select a face, edge, vertex, or object that is in the interior of a model, or when you do not want to change the model view to select an item.

Related Topics

Selecting Objects and Surfaces that Lie Inside Other Objects

Selecting Objects and Surfaces That Lie Inside Other Objects

To select objects and surfaces that lie inside other objects (such as an object that lies within an air box, a conductive shield, or the background object), do one of the following:

- Make the objects on the outside of the model invisible using the **View>Visibility** commands. This is useful when you want to select objects using the mouse. Since the mouse cannot select invisible objects, you can select the interior surfaces or objects by clicking on them.
- Use the **Edit>Select>By Name** command to select objects or surfaces inside the model.
- Use the **Next Behind** command on the shortcut menu. This selects the object that lies behind the one you initially selected. This command does nothing if no objects have previously been selected.

Related Topics

Selecting the Face or Object Behind

Using the Mouse to Select Objects

To select objects or surfaces, do one of the following:

- Click the object directly.
- Right-click an object or surface, and use one of the Select commands on the shortcut menu. The following commands appear on the shortcut menu:
  - **Select Objects**
  - **Select Faces**
  - Next Behind: Use this command to select the object or face that lies behind the currently selected object or face. This command chooses objects or faces depending on the graphical pick mode. **Next Behind** does nothing if no object has previously been selected or if the object you select has nothing behind it. You can also use the keyboard shortcut B.
  - **All Object Faces**
  - **Faces on Plane**

The snap mode defines how items are selected by the mouse. By default, **Grid** and **Vertex** snaps are enabled.

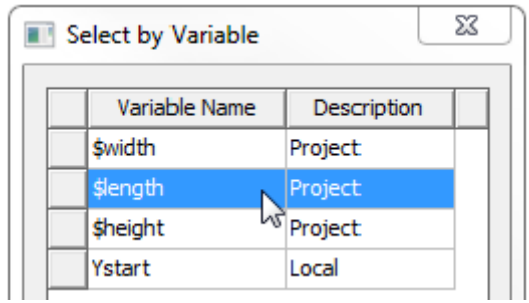
Selecting Objects by Variable

You can select an object based on a variable that affects it. If your design includes variables, you do do so as follows:

1. Click **Edit>Select>By Variable**.

   This displays the Select by Variable dialog, which lists the variables in your design. Both columns are sortable by clicking the header. You can resize and move the dialog. When you next

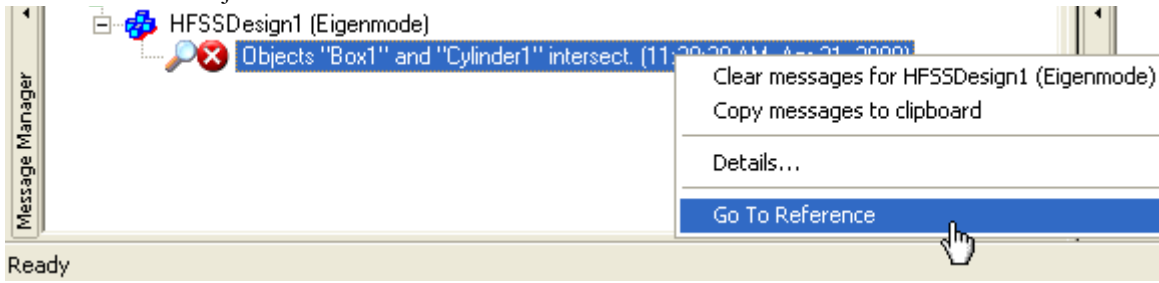
open it, it uses that size and location.



2. Select the variable of interest, and click **OK**.

   The dialog closes, and the object affected by the variable is highlighted in the Modeler window.

   The Message window contains a reference that you can select and use to go to the affected object.



   If you execute the command again, without clearing the current selection(s), the additional object can be highlighted. You can resize and move the dialog. When you next open it, it uses that size and location.

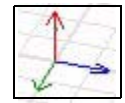
Assigning Coordinates to New Objects

When you insert and draw new 3D, 2D, or 1D objects in the geometry model, you need to define coordinate systems and locations of points, distances between objects, and other geometry items.

- Setting the Reference Point
- Defining Cartesian Coordinates
- Defining Spherical Coordinates
- Defining Relative Coordinates
- Defining Absolute Coordinates

Setting the Reference Point

When you draw objects, the cursor's location is always relative to a reference point. The reference point is displayed with a mini xyz-axis:



To change the reference point:

1. Select the drawing command to use.

   The Measure Data dialog opens. As you move the cursor over the modeler window, the top line in the measure dialog shows the coordinates of the current reference point.

2. Move the cursor to the desired reference point and press **Ctrl+Click** or right click and select **Set Reference Point** from the short cut menu.

   This moves the reference point marker to the new location. The **Measure Data** dialog updates. The coordinates boxes in the Status bar change to accept relative distance information. If you choose, rather than setting the reference point with the cursor, you can press Tab to activate a text cursor in the status bar fields, and enter coordinates directly.

Related Topics

Assigning Coordinates to New Objects

Choosing the Movement Mode

Choosing Snap Settings

Drawing Objects

Defining Cartesian Coordinates

When you draw an object, define a point using Cartesian coordinates by typing the point's distance from the origin in the x, y, and z directions in the **X**, **Y**, and **Z** text boxes, respectively. When defin-

ing a second point, specify its distance from the previously selected point in the x, y, and z directions in the **dX**, **dY**, and **dZ** text boxes, respectively.

1. Select the desired drawing command.
2. Select **Cartesian** from the pull-down list in the status bar.
3. Type the point's x-, y-, and z-coordinates in the **X**, **Y**, and **Z** text boxes.

Hint Press **Tab** to move from one coordinate text box to the next. Press **Ctrl+Tab** to move to the previous coordinate text box.

   - Alternatively, click the point in the view window.
4. When drawing objects other than polylines and helices, the second point you select is relative to the first point. Type the second point's distance from the previously selected point in the x, y, and z directions in the **dX**, **dY**, and **dZ** text boxes, respectively.

Related Topics

Defining Cylindrical Coordinates

Defining Spherical Coordinates

Defining Cylindrical Coordinates

To define a point using cylindrical coordinates,, specify the point's radius, measured from the origin, in the **R** text box, the angle from the x-axis in the **Theta** text box, and the distance from the origin in the z direction in the **Z** text box. When defining a second point, specify its distance from the previously selected point in the in **dR**, **dTheta**, and **dZ** text boxes.

1. After clicking the desired drawing command, select **Cylindrical** from the pull-down list in the status bar.
2. Type the point's r-, theta-, and z-coordinates in the **R**, **Theta**, and **Z** boxes.

Hint Press **Tab** to move from one coordinate text box to the next. Press **Ctrl+Tab** to move to the previous coordinate text box.

   - Alternatively, click the point in the view window.
3. When drawing objects other than polylines and helices, the second point you select is relative to the first point. Type the second point's distance from the previously selected point in the **dR**, **dTheta**, and **dZ** text boxes.

Related Topics

Defining Cartesian Coordinates

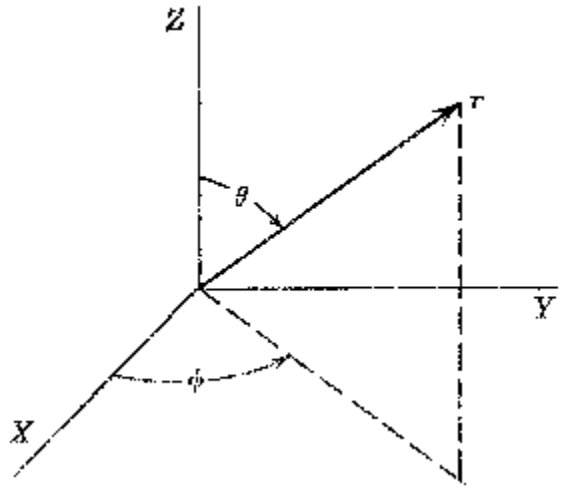
Defining Spherical Coordinates

Defining Relative Coordinates

Defining Absolute Coordinates

Defining Spherical Coordinates

To define a point in spherical coordinates, specify the point's radius, measured from the origin, in the **Rho** text box, the angle from the x-axis in the **Theta** text box, and the angle from the origin in the z direction in the **Phi** text box. When selecting a second point, specify its distance from the previously selected point in the in **dRho**, **dTheta**, and **dPhi** text boxes.



Note Even though you are inputting spherical coordinates, all data is internally stored in Cartesian coordinates.

To define a point in spherical coordinates.

1. After clicking the desired drawing command, select **Spherical** from the pull-down list in the status bar.
2. Type the point's r-, theta-, and phi-coordinates in the **Rho**, **Theta**, and **Phi** text boxes in the status bar.

Hint Press **Tab** to move from one coordinate text box to the next. Press **Ctrl+Tab** to move to the previous coordinate text box.

   - Alternatively, click the point in the view window.
3. When drawing objects other than polylines and helices, the second point you select is relative to the first point. Type the second point's distance from the previously selected point in the **dRho**, **dTheta**, and **dPhi** text boxes.

Related Topics

Defining Cartesian Coordinates

Defining Cylindrical Coordinates

Defining Relative Coordinates

Defining Absolute Coordinates

Using Absolute Coordinates

When entering a point's coordinates, you can specify them in *absolute* or *relative* coordinates. Absolute coordinates are relative to the working coordinate system's origin (0, 0, 0). This is the default setting for the first point you select after clicking a drawing command. Relative coordinates are relative to the reference point, or the previously selected point.

To enter a point's absolute coordinates:

1. Click the desired drawing command.
2. Select **Absolute** from the **Absolute/Relative** pull-down list in the status bar.
3. Specify the point's coordinates in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the appropriate text boxes in the status bar.

Note When drawing objects other than polylines and helices, by default, the second point you select is relative to the first point; **Relative** is automatically selected in the **Absolute/Relative** pull-down list in the status bar. Be sure to select **Absolute** from the **Absolute/Relative** pull-down list in the status bar if you want the second point to be relative to the working coordinate system.

Related Topics

Defining Relative Coordinates

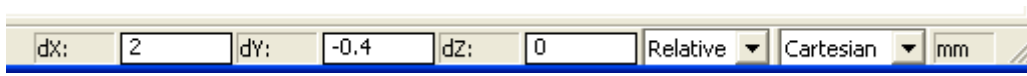
Using Relative Coordinates

When entering a point's coordinates, you can specify them in *absolute* or *relative* coordinates. Relative coordinates are relative to the reference point, or the previously selected point. Absolute coordinates are relative to the working coordinate system's origin (0, 0, 0).

To enter a point's relative coordinates:

1. Click the desired drawing command.
2. Select **Relative** from the **Absolute/Relative** pull-down list in the status bar.

   When you are in relative mode, the text boxes for a coordinate show an "d" before the coordinate description, to indicate "distance from" the working reference. For example:



3. Specify the point's coordinates in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the appropriate text boxes in the status bar.

Related Topics

Defining Absolute Coordinates

Choosing the Movement Mode

When drawing objects, the cursor's location is always relative to a reference point. The reference point is displayed with a mini xyz-axis:

To change the reference point, move the cursor to the desired point and press **Ctrl+Click**.

You can move the cursor to one of the following points:

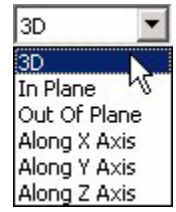
- In the same plane as the reference point (in-plane movement mode).
- Perpendicular to the reference point (out-of-plane movement mode).
- If an object is present to snap to a point in 3D space (3D movement mode).
- Along the x-axis.
- Along the y-axis.
- Along the z-axis.

Changes you make to the movement mode persist until you change them again.

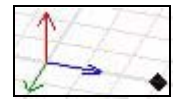
Moving the Cursor In Plane

To move the cursor to a point *on the same plane* as the reference point

1. Click the desired drawing command.
2. Do one of the following:
   - Click **3D Model** > **Movement Mode**>**In Plane**.
   - Click **In Plane** in the movement mode pull-down list in the 3D Modeler Draw toolbar.



The next point you select will be on the same plane as the reference point.



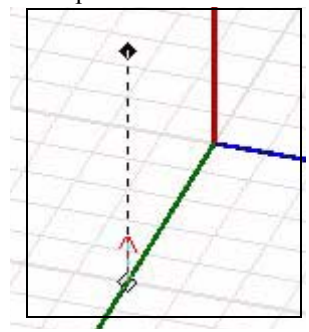
The cursor's location, displayed with a black diamond that indicates it has snapped to the grid, is on the same plane as the reference point.

Moving the Cursor Out of Plane

To move the cursor to a point *perpendicular* to the drawing plane and that intersects the reference point:

- After clicking the desired drawing command, click **3D Model** > **Movement Mode**,>**Out of Plane**.

  A dashed line is displayed between the reference point and the cursor's location, which is now perpendicular to the reference point.



The cursor's location, displayed with a black diamond that indicates it has snapped to a grid point, is perpendicular to the reference point.

Moving the Cursor in 3D Space

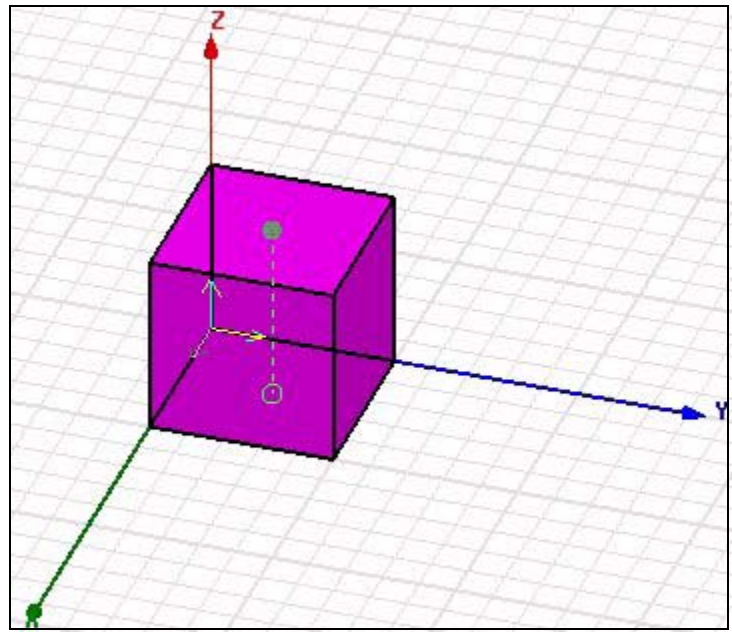
To move the cursor to a point in 3D space relative to the reference point:

1. Click the desired drawing command.
2. Do one of the following:
   - Click **3D Model** > **Movement Mode**>**3D**.
   - Click **3D** in the movement mode pull-down list in the 3D Modeler Draw toolbar.

If one of an object's snapping centers is within snapping range, the cursor will snap to the nearest point in 3D space occupied by the object.

If an object is not within snapping range, 3D movement mode is identical to the in-plane

movement mode.

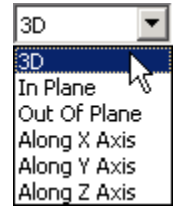


The cursor's location, displayed by a circle that indicates it has snapped to a face center, is (0.5, 0.5, 1.0), a point in 3D space relative to the reference point.

Moving the Cursor Along the X-Axis

To move the cursor to a point away from the reference point in the x direction:

1. Click the desired drawing command.
2. Do one of the following:
   - Click **Modeler>Movement Mode>Along X Axis**.
   - Hold the shortcut key **X**.
   - Click **Along X Axis** in the movement mode pull-down list in the 3D Modeler Draw toolbar:

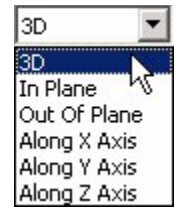


The next point you select will be on the same plane as the reference point in the positive or negative x direction.

Moving the Cursor Along the Y-Axis

To move the cursor to a point away from the reference point in the y direction:

1. Click the desired drawing command.
2. Do one of the following:
   - Click **Modeler>Movement Mode>Along YAxis**.
   - Hold the shortcut key **Y**.
   - Click **Along Y Axis** in the movement mode pull-down list in the 3D Modeler Draw toolbar:

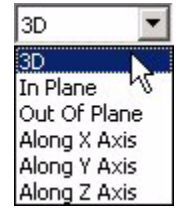


The next point you select will be on the same plane as the reference point in the positive or negative y direction.

Moving the Cursor Along the Z-Axis

To move the cursor to a point away from the reference point in the z direction:

1. Click the desired drawing command.
2. Do one of the following:
   - Click **Modeler>Movement Mode>Along Z Axis**.
   - Hold the shortcut key **Z**.
   - Click **Along Z Axis** in the movement mode pull-down list in the 3D Modeler Draw toolbar:



The next point you select will be on the same plane as the reference point in the positive or negative z direction.

Choosing Snap Settings

By default, the selection point and graphical objects are set to "snap to", or adhere to, a point on the grid when the cursor hovers over it. The coordinates of this point are used, rather than the exact location of the mouse. The cursor changes to the shape of the snap mode when it is being snapped.

To change the snap settings for the active design, you can use either the **Modeler** menu or the toolbar icons  :

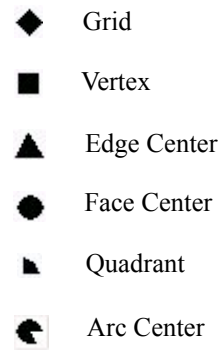
1. Click **Modeler>Snap Mode** or click the toolbar icons.

   If you select the menu command, the **Snap Mode** window appears.

2. Specify the snap mode settings you want.

- If you want the cursor to snap to a point on the grid, select **Grid** or the icon  .
- To snap to a vertex, select **Vertex** or the icon  .
- To snap to the center point of an edge, select **Edge Center** or the icon  . The center point may be on a 1D, 2D, or 3D object edge.
- To snap to the center of an object face, select **Face Center** or the icon  .
- To snap to the nearest quarter point on an edge, select **Quadrant** or the icon  .
- To snap to the center of an arc, select **Arc Center** or the icon  .

  When the cursor snaps to a point, it will change to one of the following snap mode shapes:



Note By default, the mouse is set to snap to the grid, a vertex, an edge center, a face center, and the nearest quadrant. To modify the default snap settings for the active design and all new designs, modify the selections under the **Drawing** tab in the **Modeler Options** dialog box.

Related Topics

Snap Setting Guidelines

Snap Setting Guidelines

For each object (3D, 2D, or 1D), you can define snapping points. Then, when moving in the Modeler window, if the cursor approaches a snap point (gets closer in terms of screen coordinates, not 3D coordinates), the snap point is highlighted. If you then click it, it becomes selected, even if that point is out of the plane you are working in (for example, even if the point has 3D coordinates when you are working in a 2D plane).

In general, select at least one of the snap options in the **Snap Mode** window. If none of these options are selected, the software is in "free mode" and selects whatever point you click, regardless of its coordinates. This can cause problems when you are trying to create closed objects. Although the point you select may appear to be the vertex point of an open object, you may not have actually clicked the exact coordinates of the point.

Measure Modes for Objects

The Measure modes lets you measure the position, length, area, and volume of objects. With two faces selected, with two edges selected, or with an edge and a face selected, the Measure Mode displays the angle and distance between them. The Measure Position mode dynamically measures the distance between a reference point and the cursor location.

1. To access the Measure mode, either:
   - Select **Modeler>Measure.**
   - Right-click and select **Measure** from the short-cut menu.

   After you select **Measure**, a cascading menu appears for **Position**, **Edge**, **Face** and **Object**.
2. Select **Position** to obtain location and distance information between a specified reference point and the cursor location.

   The **Measure Information** dialog box appears.

   With **Measure>Position** selected, the information displayed includes:
   - The location of the current reference point. (Position1)
   - The current cursor location. (Position2)
   - The distance between the Reference and Current location.
   - The X distance.
   - The Y distance.
   - The Z distance.
   - The angle between the current reference point and the current cursor location.

   Note As you move the cursor, the Measure Information dialog displays the current cursor location and measurement information from the reference. Clicking on a new vertex updates the reference to the new location.

   With **Measure>Edge**, **Face**, or **Object** selected, the information displayed for each selected object is the name and:
   - The area and volume of a 3D object.
   - The area of a face.
   - The length of a polyline (in edge selection mode, you can still see this if you select the polyline in the History tree)
   - The length of an edge
   - The location of a vertex.

   For more information on cursor and reference point behavior in this mode, see Measuring Position and Distance
3. To use **Measure>Edge**, **Face**, or **Object** to measure the distance and angle between two selected items:

- Select two points. Click the first and Ctrl-click to select the second.

  The **Measure Information** dialog displays the coordinates of each point, the distance between the points and the angle between Origin-P1, Origin-P2 line.
- Select two faces, the **Measure Information** dialog displays the angle/distance between them.
- The function is similar when you select two edges and when you select an edge and a face.
- You can also measure distance between vertex/face, vertex/edge. In these cases, use the Select Multi mode.

4. To exit the Measure mode, click **Close** on the Measure Information dialog.

Related Topics

Measuring Position and Distance

Setting Coordinate Systems

Modifying the Coordinate System Axes View

Choosing Snap Settings

Choosing Movement Mode (3D, in plane, X, Y, or Z)

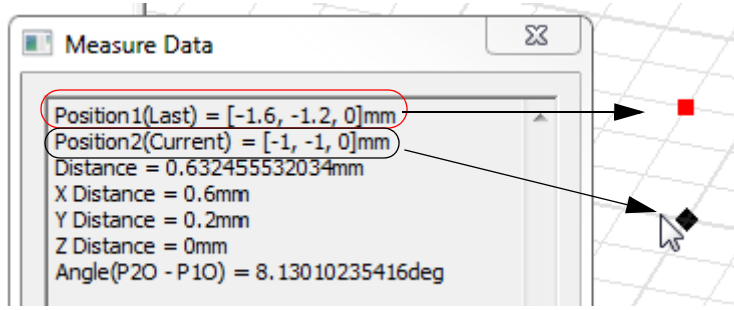
Select Multi (a mode for selecting objects, faces, edges or vertices)

Measuring Position and Distance

To measure the distance between any cursor location relative to a designated reference point:

- Select **Modeler>Measure>Position**.

  This enables the Measure Position mode opens the **Measure Data** dialog. The dialog lists the coordinates of the current reference point (Position1)and the cursor location (Position2). If you click, it shows the last position as a red square, and the current position as a black cursor. It also lists the distance between those points, the X, Y, and Z distances, and the angle between them.



  The shortcut menu displays the Hints item. When Hints are on (the default), a text display in the lower right of the 3D Modeler window, explaining how to set the reference point, and ways to control the movement mode.

- The reference point is displayed as a mini x-y-z-axis:

  Use Ctrl-Click to set the reference point at a new location.
- The cursor leads a diamond-shape selection marker that snaps from grid point to grid point. The **Measure Data** dialog also provides a text identification of the current grid points. If you drag the cursor off design objects, by default, it moves in the xy-plane. You can restrict movement to in a specific plane, out of plane, or z, x, or y. Besides the context menu for movement, you can also use the X, Y, and Z keys to restrict movement. See Choosing a Movement Mode. for further details.

  If you drag the selection marker over an object, it follows the 3D surfaces of the object, dropping a dashed reference line to a point on the current plane. The cursor changes shape to provide information about the object at the corresponding coordinate.:

Grid point

Vertex

Edge Center

Face Center

Quadrant

To measure the distance between two points:

1. Select **Modeler>Measure>Position** to enter Measure Position mode.
2. Ctrl-click to set the reference point.

   The reference point display moves to the selected point. This becomes the coordinate for Position1 in the Measure Data dialog.
3. Drag the cursor to the second point.

   The value of the Position2 dynamically changes as you drag the cursor. You do not need to click. The values shown include:
   - Distance.
   - X distance
   - Y Distance.
   - Z Distance.
4. To close the dialog box and exit **Measure** mode, click the **Close** button. You can also use the ESC key to exit Measure mode.

Related Topics

Measure Modes for Objects

Setting Coordinate Systems

Modifying the Coordinate System Axes View

Choosing Snap Settings

Choosing Movement Mode (3D, in plane, X, Y, or Z)

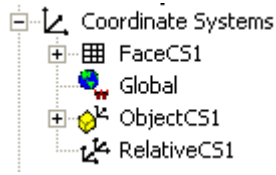
Select Multi (a mode for selecting objects, faces, edges or vertices)

Setting Coordinate Systems

The modeler has four types of coordinate systems (CS) that enable you to easily orient new objects:

- Global CS
- Relative CS.
- Face CS.
- Object CS.

Every coordinate system has an x-axis that lies at a right angle to a y-axis, and a z-axis that is perpendicular to the xy plane. The origin (0,0,0) of every CS is located at the intersection of the x-, y-, and z-axes.The default Global coordinate system and any additional coordinate systems that you create for a project appear in the History tree of the modeler window.



- The *global coordinate system* (CS) is the fixed, default CS for each new project. It cannot be edited or deleted.
- A *relative CS* is user-defined. Its origin and orientation can be set relative to an existing CS. Relative CSs enable you to easily draw objects that are located relative to other objects. If you modify a relative CS, all objects drawn on that CS will be affected and change position accordingly. You can define a relative CS to be *offset* and/or *rotated* from an existing CS. This feature provides a way for objects made of the same anisotropic materials to have different orientations.

  When you set a new relative coordinate system, you specify whether to express the coordinates as Absolute or Relative Coordinates. Absolute uses the specified values in terms of the global coordinate system. Relative interprets the values as differences from the current working CS.

  You have choices for expressing the coordinates as Cartesian, Cylindrical, or Spherical. These are evaluated as Cartesian for the coordinate system properties.
- A *face CS* is also user-defined. Its origin is specified on a planar object face. Face CSs enable you to easily draw objects that are located relative to an object's face.
- An object CS is user-defined as attached to a specific object.

Switch between global, relative, object and face CSs by changing the *working CS* . Simply click the CS you want to use in the history tree. The working CS is indicated by a red *W* that appears at the lower-left corner of the CS name in the history tree (as shown on Global in the figure above). The **Properties** dialog box lists the CS associated with an object as the Orientation. By default, this is Global, but if you have created the object under a different coordinate system, that will be the orientation. You can click on the current orientation to see a drop down list of other orientations that you can assign for an object.

User-defined CSs are saved with the active project. When you open a project, it uses the CS designated as working CS when you last saved.

Related Topics

Creating a Relative Coordinate System

Creating a Face Coordinate System

Creating an Object Coordinate System

Setting the Working Coordinate System

Modifying the Coordinate System Axes View

Assigning Material Property Types

Change the Orientation of an object

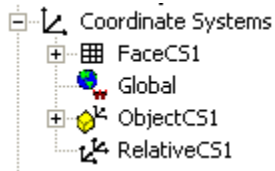
Setting the Working Coordinate System

The working coordinate system (CS) is the current CS with which objects being drawn are associated. The working CS can be the global CS or a user-defined relative CS, Object or Face CS. Select the working CS by clicking its name in the history tree, or follow this procedure:

1. Click **Modeler>Coordinate System>Set Working CS**.

   The **Select Coordinate System** dialog box appears.
2. Click a CS in the list.
3. Click **Select**.

   A red *W* appears at the lower-left corner of the CS name in the History tree, indicating that it is the working CS. In this following figure the Global is the working CS.



   Objects that you draw hereafter will be associated with the CS you selected.

Related Topics

Setting Coordinate Systems

Creating a Relative Coordinate System

When creating a relative CS, you have the following options:

- You can create an offset relative CS, that is, a relative CS whose origin lies a specified distance from another CS's origin. By moving a CS's origin, you can enter coordinates relative to an existing object, without having to add or subtract the existing object's coordinates.
- You can create a rotated relative CS, that is, a relative CS whose axes are rotated away from another CS's axes. By rotating the axes of a CS, you can easily add an object that is turned at an angle relative to another object.

- You can also create a relative CS that is both offset and rotated.

Related Topics

Determining Phase Center using Optimetrics

Creating an Offset Relative CS

To create a relative CS with an origin that lies a specified distance from another CS's origin:

1. In the history tree, click the CS upon which you want to base the new relative CS, making it the working CS.
2. Point to **Modeler>Coordinate System>Create>Relative CS**.
3. On the **Relative CS** menu, click **Offset**  .
4. Select the origin in one of the following ways:
   - Click the point.
   - At the lower right of the modeler window, use the drop down menu to select the system for expressing coordinates (Cartesian, Cylindrical, or Spherical), select either relative or absolute coordinates, then select the units, and type the CS origin coordinates in boxes.

   To select a point that does not lie in the current plane, use the **Movement Mode** commands on the shortcut menu.

   The new relative CS is created. Its origin has moved from the previous working CS, but its axes remain the same. It is listed in the history tree under **Coordinate Systems**. It automatically becomes the working CS; objects that you draw hereafter will be based on the coordinates of this relative CS. Default planes are created on its xy, yz, and xz planes.

Related Topics

Creating a Relative Coordinate System

Creating an Offset and Rotated Relative CS

Creating a Rotated Relative CS

To create a new relative CS with its axes rotated away from another CS's axes:

1. In the history tree, select the CS upon which you want to base the new relative CS, making it the working CS.
2. Click **Modeler>Coordinate System>Create>Relative CS>Rotated**  .
3. Specify the x-axis by selecting a point on the axis in one of the following ways:
   - Click the point.
   - At the lower right of the modeler window, use the drop down menu to select the system for expressing coordinates (Cartesian, Cylindrical, or Spherical), select either relative or absolute coordinates, then select the units, and type the CS origin coordinates in boxes.

   To select a point that does not lie in the current plane, use the **Movement Mode** commands on the shortcut menu.
4. Specify the xy plane by selecting any point on it in one of the following ways:
   - Click the point.

- Type the coordinates of a point that is relative to the previously selected point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

You do not need to specify the z-axis. It is automatically calculated to be at a right angle to the y-axis.

The new relative CS is created. It has the same origin as the previous working CS, but its axes are rotated. It is listed in the history tree under **Coordinate Systems**. It automatically becomes the working CS; objects that you draw hereafter will be based on the coordinates of this relative CS. Default planes are created on its xy, yz, and xz planes.

Related Topics

Creating a Relative Coordinate System

Creating an Offset and Rotated Relative CS

Creating an Offset and Rotated Relative CS

To create a new relative CS that is both offset and rotated from an existing CS:

1. In the history tree, select the CS upon which you want to base the new relative CS, making it the working CS.
2. Point to **Modeler>Coordinate System>Create>Relative CS**.
3. On the **Relative CS** menu, click **Both**  .
4. Select the origin in one of the following ways:
   - Click the point.
   - At the lower right of the modeler window, use the drop down menu to select the system for expressing coordinates (Cartesian, Cylindrical, or Spherical), select either relative or absolute coordinates, then select the units, and type the CS origin coordinates in boxes.

   To select a point that does not lie in the current plane, use the **Movement Mode** commands on the shortcut menu.
5. Specify the x-axis by selecting a point on the axis in one of the following ways:
   - Click the point.
   - Type the coordinates of a point that is relative to the origin in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
6. Specify the xy plane by selecting any point on it in one of the following ways:
   - Click the point.
   - Type the coordinates of a point that is relative to the previously selected point in the **dX**, **dY**, and **dZ** boxes.

   You do not need to specify the z-axis. It is automatically calculated to be at a right angle to the y-axis.

   The new relative CS is created. It is listed in the history tree under **Coordinate Systems**. It automatically becomes the working CS; objects that you draw hereafter will be based on the coordinates of this relative CS. Default planes are created on its xy, yz, and xz planes.

Related Topics

Creating a Relative Coordinate System

Creating a Face Coordinate System

1. Select the object face upon which you want to create the face CS.
2. Click **Modeler>Coordinate System>Create>Face CS**  .
3. Select the origin in one of the following ways:
   - Click the point on the face.
   - Type the point's coordinates in the **X**, **Y**, and **Z** boxes of the editable fields in the status bar.
4. Specify the x-axis by selecting a point on the object face in one of the following ways:
   - Click the point.
   - Type the coordinates of a point that is relative to the previously selected point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   You do not need to specify the y- or z-axes. The modeler assumes that the z-axis is normal to the object face and the y-axis is automatically calculated to be at a right angle to the z-axis.

   The new face CS is listed in the history tree under **Coordinate Systems**. It automatically becomes the working CS; objects that you draw hereafter will be referenced to the coordinates of this face CS. Default planes are created on its xy, yz, and xz planes.

Only operations listed in the history tree *before* the face CS's creation will affect the face CS, and in turn, affect objects dependent upon that face CS. A face CS, or objects created on it, is *not* affected by operations that occur after it is created. Also see the Move CS to End command.

For example, suppose you create a box, then a face CS on a face of the box, and then a cylinder on the face CS. If you then edit the box's dimensions in the **Properties** dialog box, the cylinder will move accordingly. But if you rotate the box using the **Edit>Arrange>Rotate** command, the box will move, but the cylinder will not move because the operation occurs later in the history tree.

Related Topics

Automatically Creating Face Coordinate Systems

Setting the Working Coordinate System

Modifying Coordinate Systems

Setting Coordinate Systems

Move CS to End

Automatically Creating Face Coordinate Systems

You can instruct the modeler to automatically create a new face CS every time you draw on an object's face.

1. Click **Tools>Options>Modeler Options**.

   The **Modeler Options** dialog box appears.

2. Select **Automatically switch to face coordinate system**.
3. Click **OK**.

Now, when you select a face, and then click a drawing command, a new face CS will be created on the face. The modeler automatically sets the new face CS as the working CS. The object you draw is oriented according to the new face CS.

Note The modeler will not automatically create a new face CS if a face CS has already been assigned to the selected face.

Related Topics

Creating a Face Coordinate System

Move CS to End

Creating an Object Coordinate System

Creating an Object Coordinate System

You can create coordinate systems based on any object of solid, sheet, or wire type. The **Modeler>Coordinate System>Create>Object CS** command is enabled when you select an object. An Object CS can be Offset, Rotated or both. Executing one of the Object CS commands changes the cursor to the selection marker mode.

As you drag the selection marker over an object, it follows the 3D surfaces of the object, dropping a dashed reference line to a point on the current plane. The cursor changes shape to provide information about the object at the corresponding coordinate.:

 Grid point

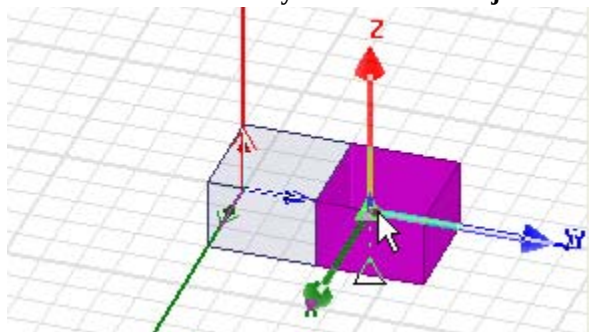
 Vertex

 Edge Center

 Face Center

 Quadrant

For example, in this case, the cursor shows Edge Center triangles as valid selection points for an **Modeler>Coordinate Systems>Create Object CS>Offset** command.



Only operations listed in the history tree *before* the Object CS's creation will affect the Object CS, and in turn, affect objects dependent upon that Object CS. An Object CS, or objects created on it, is *not* affected by operations that occur after it is created. Also see the Move CS to End command.

For example, suppose you create a box, then an Object CS on a face of the box, and then a cylinder on the Object CS. If you then edit the box's dimensions in the **Properties** dialog box, the cylinder will move accordingly. But if you rotate the box using the **Edit>Arrange>Rotate** command, the box will move, but the cylinder will not move because the operation occurs later in the history tree.

Related Topics

Creating a Face Coordinate System

Move CS to End

Creating an Offset Object CS

Creating a Rotated Object CS

Creating an Object CS that is both Rotated and Offset

Expressing Cartesian, Cylindrical, or Spherical Coordinates

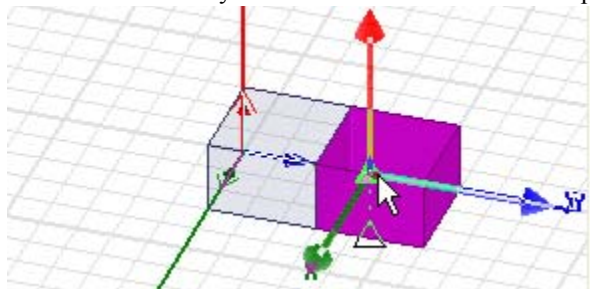
Creating an Offset Object CS

To create an Offset Object coordinate system (CS):

1. Select the working CS.
2. Select the object, and click **Modeler>Coordinate System>Create>Object>Offset**.

- You can select any snap point on object based on current snapping mode to select origin of the

CS. When you hover the mouse over a valid point, a coordinate system preview is shown.



- The point must be on selected object.
- X axis is taken as {1,0,0} and Y Axis as {0,1,0} to create an Object CS. The axis coordinates can be later edited through the Properties dialog.
- When you select the point, validation displays an appropriate message. Points where CS preview is available are always valid.

Related Topics

Creating a Face Coordinate System

Move CS to End

Creating an Offset Object CS

Creating a Rotated Object CS

Creating an Object CS that is both Rotated and Offset

Expressing Cartesian, Cylindrical, or Spherical Coordinates

Creating a Rotated Object CS

To create a Rotated Object coordinate system (CS):

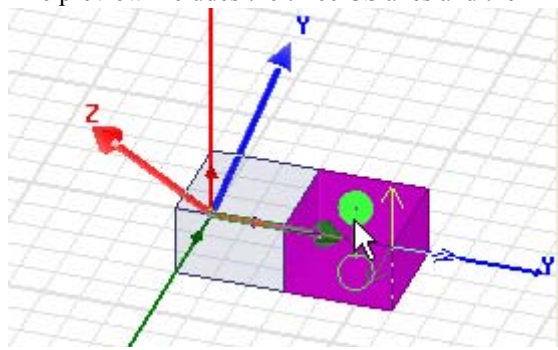
1. Select the working CS.
2. Select the object, and click **Modeler>Coordinate System>Create>Object>Rotated**

- The origin is taken as {0,0,0}. You can edit this later through the Properties window.
- The GUI is in multi select mode where you define the direction by picking any of vertex, straight edge, planar face or conical face.
  - If you select vertex, direction from origin to vertex defines the axis.
  - If you select a straight edge, the edge direction defines the axis.
  - If you select a planar face, face normal defines the axis.
  - If you select a conical face, the face axis defines the CS axis.
- The selected vertex, edge or face must be on the selected object.

3. You are prompted to first select X axis. A preview of X axis is shown for valid selections.
4. You are later prompted to select to define the XY plane. Another direction (edge or face selection) or point (i.e. vertex selection) helps to define the plane.

- During 2nd XY plane selection, a CS preview shows as the cursor hovers over valid selections.

The preview includes the three CS axes and the XY plane.



- In 2D modeler, you are prompted to select only X axis. The Y axis is defined based on 2D modeler type. This resembles **Relative CS** behavior. Face picking is not available in 2D modeler as it defines the direction out of plane.

Related Topics

Creating a Face Coordinate System

Move CS to End

Creating an Offset Object CS

Creating a Rotated Object CS

Creating an Object CS that is both Rotated and Offset

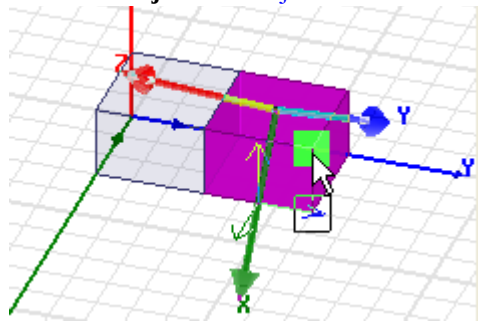
Expressing Cartesian, Cylindrical, or Spherical Coordinates

Creating an Object CS that is Both Offset and Rotated

To create an Object CS that is both offset and rotated:

1. Set the working CS.
2. Select the object

- Behavior is combination of offset and rotated object CS. You are first prompted to select origin as in **Object CS**>Object and then axes as in **Object CS**>Rotated.



- In 2D modeler, you are asked to select only the origin and X axis. The Y axis is computed as for **Object CS>Rotated**. Face picking is not available in 2D modeler as it defines the direction out of plane.

Related Topics

Creating a Face Coordinate System

Move CS to End

Creating an Offset Object CS

Creating a Rotated Object CS

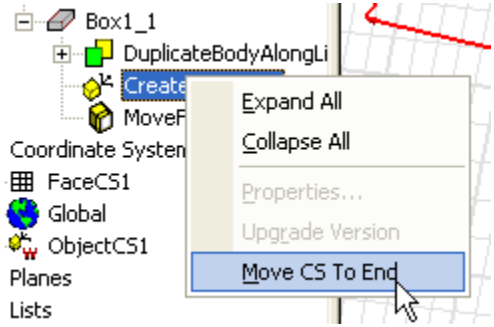
Creating an Object CS that is both Rotated and Offset

Expressing Cartesian, Cylindrical, or Spherical Coordinates

Move CS to End Command for History Tree

Only operations listed in the history tree *before* the Object CS's creation will affect the Object CS, and in turn, affect objects dependent upon that Object CS. An Object CS, or objects created on it, is *not* affected by operations that occur after it is created. It is sometimes useful to have the coordinate system affected after any other operations that might have edited, moved or rotated the object.

If you have at least one History operation after a Face or Object CS in the History tree, selecting a Face or Object CS enables the **Move CS to End** command in the **Modeler>Coordinate System** cascade menu and the History Tree shortcut menu.



Executing the command moves the selected CreateObject CS to the end position in the History tree and updates associated items (other CS, object history, any dependent parts etc.).

Related Topics

Creating a Face Coordinate System

Move CS to End

Creating an Offset Object CS

Creating a Rotated Object CS

Creating an Object CS that is both Rotated and Offset

Modifying Coordinate Systems

Keep in mind that when you edit a CS, the following will also be affected:

- All objects drawn on the CS.
- All CSs that were defined relative to that CS.
- All objects drawn on a CS that was defined relative to that CS.

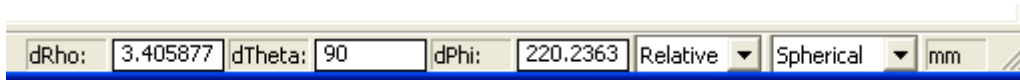
There are two ways to modify a coordinate system: you can select the coordinate system in the history tree in the modeler window, and open its properties dialog. This approach does not also allow you to change whether the coordinate system is Absolute or Relative, or to change how you express the coordinates (as Cartesian, Cylindrical, or Spherical).

If you want to also modify the whether the coordinate system is Absolute or Relative, and to change how your express the coordinate, do the following:

1. Click **Modeler>Coordinate System>Edit**.

   The **Select Working CS** window appears.
2. Click the CS you want to modify.
3. Click **Select**.

   This selects that coordinate system and enables the editable fields in the status bar at the lower right of the Modeler window. After you click the cursor in the first field, you can type in values, and tab to the next fields.



4. You can select Absolute or Relative as the Coordinate system If you selected a relative CS, follow the directions for creating a relative CS.

   If you selected a face CS, follow the directions for creating a face CS.

   If you selected an Object CS, follow the directions for creating an object CS.
5. You select the coordinate system from the drop down menus as Cartesian, Cylindrical, or Spherical.
6. Select the units from the drop down menu.

   The value you give here is translated to Cartesian coordinates in the Properties for the Coordinate system.

Object CS can be edited in the same way they are created. For example, if CS is created in offset mode, it will be edited in offset mode only.

- You can edit the reference CS of object CS through the property. This will impact object CS created in offset and rotated mode. It will not impact object CS created in both modes as it fully depends on object.
- Origin will be editable when object CS is created in rotated mode. Otherwise it will be read only text.
- X Axis and Y Axis will be editable via the Properties dialog when object CS is created in offset mode. Otherwise it will be read only text.

- Either of X Axis or Y axis can be reversed through the Properties dialog. When X axis is reversed, Y axis does not change and vice a versa.
- Some of the examples of origin properties as a text are - Vertex_10, Face_7 center, Edge_9 midpoint, Edge_8 quadrant, Edge_17 arc center.
- Some of the examples of axis properties as a text are - Face_7 normal, Edge_9 direction, Vertex_10, Face_6 axis

Related Topics

Setting Coordinate Systems

Creating a Relative Coordinate System

Creating a Face Coordinate System

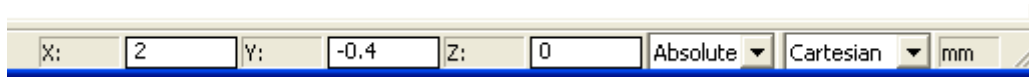
Modifying the Coordinate System Axes View

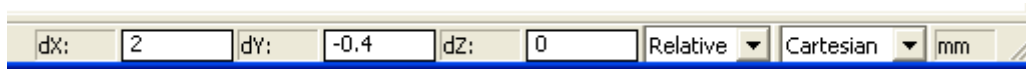
Expressing Cartesian, Cylindrical, or Spherical Coordinates

Expressing Cartesian, Cylindrical, or Spherical Coordinates

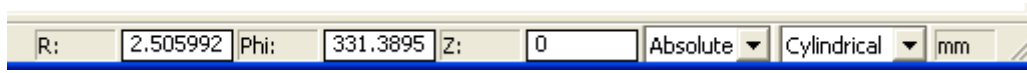
You have choices for expressing the coordinates for as Cartesian, Cylindrical, or Spherical by using the editable fields in the status bar. In each case, you also specify whether to enter the coordinates as Absolute or Relative to the working coordinate system. Click the cursor in the first text field to begin entering values or variables. Tab to the next fields, and Enter when your are done. After you enter values or variables in the text fields, they are evaluated as Cartesian for the Properties window for that coordinate system.

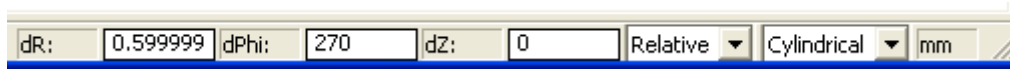
- **Cartesian**, that is the point's distance from the origin in the x, y, and z directions in the **X**, **Y**, and **Z** text boxes.





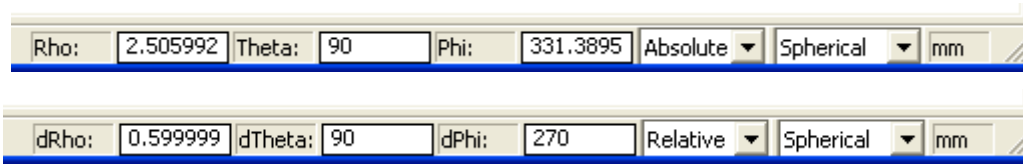
- **Cylindrical**, that is, the point's radius, measured from the origin, in the **R** text box, the angle from the x-axis in the **Theta** text box, and the distance from the origin in the z direction in the **Z** text box.





- **Spherical**, that is, in the point's radius, measured from the origin, in the **Rho** text box, the angle from the x-axis in the **Theta** text box, and the angle from the origin in the z direction in

the **Phi** text box.



Related Topics

Defining Absolute Coordinates

Defining Relative Coordinates

Defining Cartesian Coordinates

Defining Cylindrical Coordinates

Defining Spherical Coordinates

Deleting Coordinate Systems

1. Click the name of the CS you want to delete in the history tree.
2. Click **Edit**>**Delete**  .
   - Alternatively, press **Delete**.

   The CS will be deleted and all objects drawn on it will be deleted. Further, any CS that was dependent upon the deleted CS will be deleted and any objects that were drawn on the dependent CS will also be deleted.

Related Topics

Setting Coordinate Systems

Creating a Relative Coordinate System

Creating a Face Coordinate System

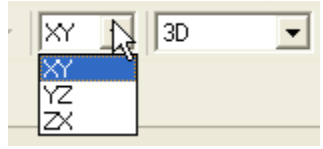
Modifying the Coordinate System Axes View

Setting the Drawing Plane

The **Drawing plane** pull-down list is available on the **Modeler Draw** toolbar, next to the **Movement mode** pull-down list.

To set the drawing plane, do one of the following:

- Select **XY, YZ, or XZ** from the **Drawing plane** pull-down list on the **Modeler Draw** toolbar.



- Click **Modeler>Grid Plane**, and then select a grid plane: **XY**, **YZ**, or **XZ**.

Related Topics

Setting the Grid Plane

8

Assigning Boundaries

Boundary conditions specify the field behavior at the edges of the problem region and object interfaces. HFSS and HFSS-IE designs have different menu options for boundaries.

Within the context of HFSS and HFSS-IE, boundaries exist for two main purposes:

1. to create either an open or a closed electromagnetic model or,
2. to simplify the electromagnetic or geometric complexity of the electromagnetic model.

Click here for HFSS-IE boundaries.

You may assign the following types of boundaries to an HFSS design:

Perfect E	Represents a perfectly conducting surface.
Perfect H	Represents a surface on which the tangential component of the H-field is the same on both sides.
Impedance	Represents a resistive surface.
Radiation	Represents an open boundary by means of an absorbing boundary condition (ABC) that absorbs outgoing waves.
PML	Represents an open boundary condition using several layers of specialized materials that absorb outgoing waves.
Finite Conductivity	Represents an imperfect conductor.
Symmetry	Represents a perfect E or perfect H plane of symmetry.
Master	Represents a surface on which the E-field at each point is matched to another surface (the slave boundary) to within a phase difference.

Slave	Represents a surface on which the E-field at each point has been forced to match the E-field of another surface (the master boundary) to within a phase difference.
Lumped RLC	Represents any combination of lumped resistor, inductor, and/or capacitor in parallel on a surface.
Anisotropic Impedance	Represents a boundary condition used to replace a surface a planar screen or grid with periodic geometry.
Layered Impedance	Represents a structure with multiple layers as one impedance surface.
IE Region	Represents a conductor or dielectric that you assign to be solved using the IE solver.

You may also choose to designate a perfect E, finite conductivity, or impedance boundary as an infinite ground plane if you want the surface to represent an electrically large ground plane when the radiated fields are calculated during post processing.

Note Hiding boundaries also turns off a check for boundary overlaps during boundary assignment. In the case of very large models with many boundaries, hiding boundaries can prevent delays during boundary assignment. Full model validation will subsequently check for boundary overlaps.

For convenience, you can access the **Edit Global Materials** command from the Boundaries menu.

Note By default, the history tree in the 3D modeler window groups sheet objects according to boundary assignment. To change this, select the **Sheets** icon and right-click to display the **Group Sheets by Assignment** checkbox.

Related Topics

Technical Notes: Boundaries

Zoom to Selected Boundary

Setting Default Boundary Base Names

Designating Infinite Ground Planes

Modifying Boundaries

Deleting Boundaries

Reassigning Boundaries

Reprioritizing Boundaries

Edit Global Materials Environment

Duplicating Boundaries and Excitations with Geometry

Showing and Hiding Boundaries and Excitations

Reviewing Boundaries and Excitations in the Solver View

Setting Default Values for Boundaries and Excitations

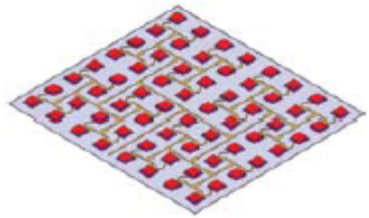
Assigning HFSS-IE Boundaries

Boundaries for Open or Closed Models

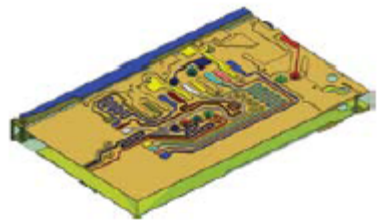
In HFSS a closed model simply represents a structure, or a solution volume, where no energy can escape except through an applied port. For an Eigenmode simulation, this could be a cavity resonator. For a driven modal or terminal solution, this could be a waveguide or some other fully enclosed structure.

An open model represents an electromagnetic model that allows electromagnetic energy to emanate or radiate away. Common examples would be an antenna, a PCB, or any structure that is not enclosed within a closed cavity. While most HFSS simulations deal with models that are open, by default, HFSS initially assumes that any given model is closed. HFSS assumes all outer surfaces of the solution space are covered, or coated, by a perfect electric conductor boundary. In order to create an open model, you specify a boundary on the outer surfaces that will overwrite the default perfect electric conductor boundary.

Electromagnetically "Open" Structures

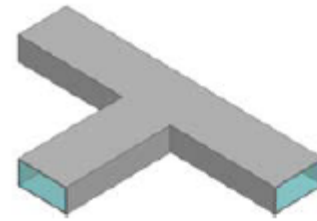


Antenna Array



Printed Circuit Board

Electromagnetically "Closed" Structures



Waveguide

Cavity Resonator

Related Topics

Assigning Boundaries

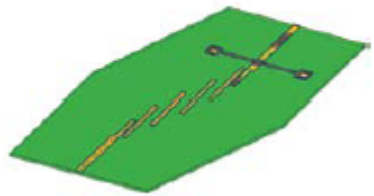
Getting Started Guides

Example Projects

Boundaries for Simplifying Models

Boundaries can be within HFSS is to decrease the geometric/electromagnetic complexity of a given structure or model. These boundaries should only be used internally to a model or possibly on a symmetry plane (as in the Dielectric Resonator Antenna example or the Coax Bend Transient example). They should be applied to specifically created 2D sheet objects or to specific surfaces of 3D objects. While boundaries can be very useful, a user should exercise caution when using them as they can create unintended results if applied incorrectly.

Electromagnetically Simplified Structure



Filter where microstrip traces are modeled using 2D objects with boundary applied.

Geometrically Simplified Structure

Coaxial structure where "shield" is replaced by appropriate boundary.

Not every HFSS model, however, will use simplifying boundaries. When using boundaries to create simpler models, users should take care to not create a model that has unreasonable or inappropriate boundaries applied.

Related Topics

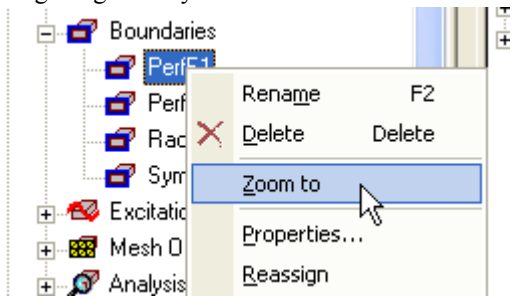
Assigning Boundaries

Getting Started Guides

Example Projects

Zoom to Selected Boundary

To zoom to a selected boundary, right-click on a boundary name in the Project and select the **Zoom to** command on the popup menu. This zooms the view in the Modeler window in or out to show the selected boundary. The current orientation does not change. This can be very useful checking the assigned geometry.



Related Topics

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

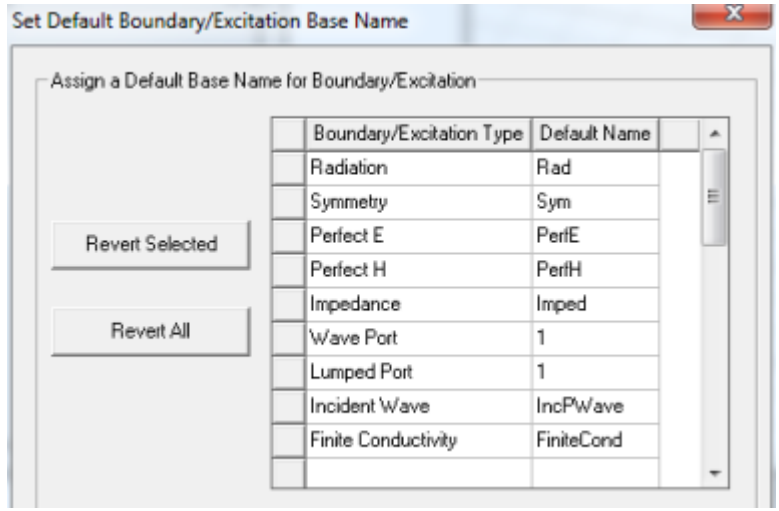
Deleting Boundaries

Setting Default Boundary/Excitation Base Names

To change the default boundary or excitation base names, so that subsequent names increment from the base of your choosing:

1. Click **Boundary>Set Default Base Name** or **Excitation >Set Default Base Name**.

   This displays the Set Default Boundary/Excitation Base dialog. This contains a list of all boundary and excitation types, and the base names for each. The base names for each type have editable text fields. The base names for boundaries and excitations are incremented from the base names here.



2. Edit the text fields to your preferred naming conventions. Names must be less than sixty characters and cannot include spaces. Letters, numbers, and special characters are permitted. Illegal names are not accepted and generate a warning message.
3. Click OK to accept the changes or Cancel to close the dialog without accepting changes.

If you want to revert all or selected names to defaults, use the **Revert All** or **Revert Selected buttons**.

Related Topics

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

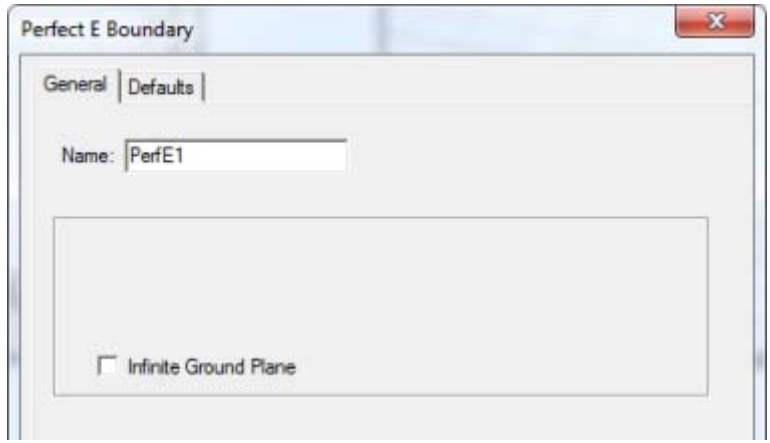
Deleting Boundaries

Assigning Perfect E Boundaries

A perfect E boundary is used to represent a perfectly conducting surface in a structure.

To create a Perfect E boundary

1. Select a surface on which to assign the boundary and click **HFSS** or **HFSS-IE>Boundaries>Assign>Pefect E** to bring up the **Perfect E Boundary** dialog box.



2. For HFSS projects, you can select **Infinite Ground Plane** if you want the surface to represent an electrically large ground plane when the radiated fields are calculated during post processing. For PEC boundaries only, multiple infinite ground planes are supported.

Related Topics

Technical Notes*:* Perfect E Boundaries

Setting Default Boundary/Excitation Base Names.

Getting Started Guides: A Dielectric Resonator Antenna

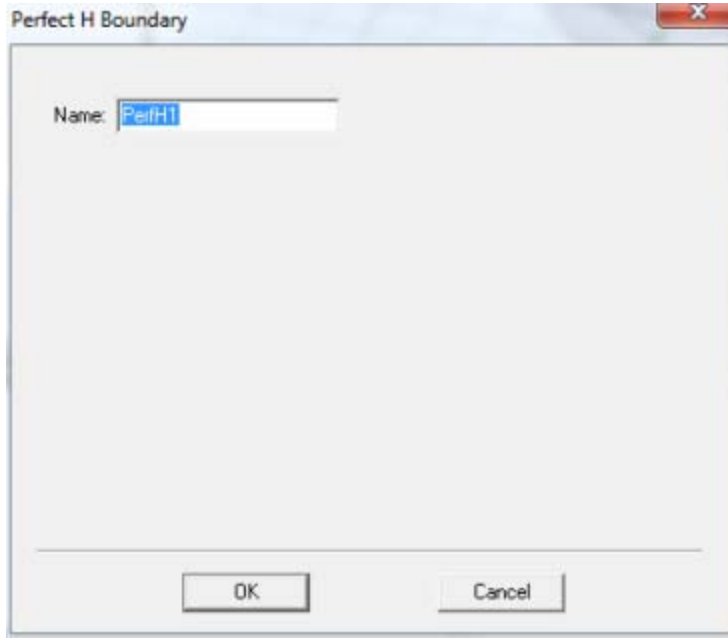
Getting Started Guides: Patch Antenna

Assigning Perfect H Boundaries

A perfect H boundary represents a surface on which the tangential component of the H-field is the same on both sides. For internal surfaces, this results in a natural boundary through which the field propagates. For surfaces on the outer surface of the model, this results in a boundary that simulates a perfect magnetic conductor in which the tangential component of the H-field is zero.

To Assign a Perfect H Boundary:

1. Select a surface on which to assign the boundary and click **HFSS>Boundaries>Assign>Pefect H** to bring up the **Perfect H Boundary** dialog box.



2. No parameters need be set..

Related Topics

Assigning Boundaries

Technical Notes: Boundaries

Zoom to Selected Boundary

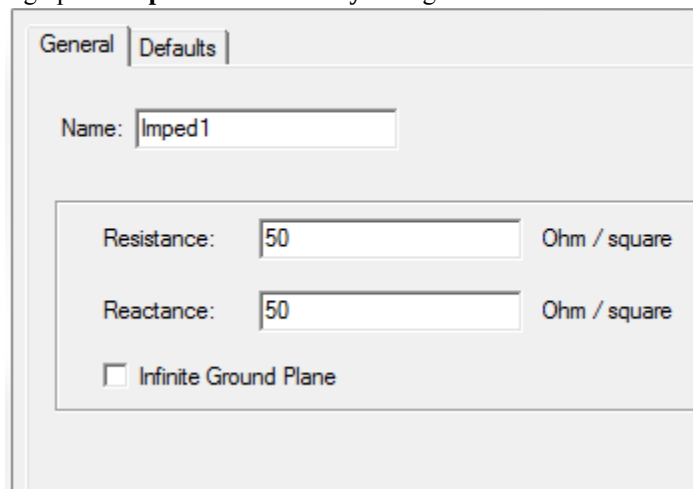
Setting Default Boundary Base Names

Getting Started Guides: A Dielectric Resonator Antenna

Assign Impedance Boundaries

An impedance boundary (IB) represents a resistive surface. The behavior of the field at the surface and the losses generated by the currents flowing inside the resistor are computed using analytical formulae. HFSS does not simulate any fields inside the resistor.

1. Select a surface on which to assign the boundary, right-click **Assign Boundary>Impedance** to bring up the **Impedance Boundary** dialog box.



2. Enter the **Resistance** and **Reactance**.
3. Select **Infinite Ground Plane** if you want the surface to represent an electrically large ground plane when the radiated fields are calculated during post processing. .

Note If you select **Infinite Ground Plane**, the effect of the impedance boundary will be incorporated into the field solution in the usual manner, but the radiated fields will be computed as if the lossy ground plane is perfectly conducting. Only one infinite ground plane is permitted in designs with impedance boundaries

Note You can assign a variable as the resistance and reactance values. Eigenmode designs cannot contain design parameters that depend on frequency, for example, a frequency-dependent impedance boundary condition.

Related Topics

Zoom to Selected Boundary

Setting Default Boundary Base Names

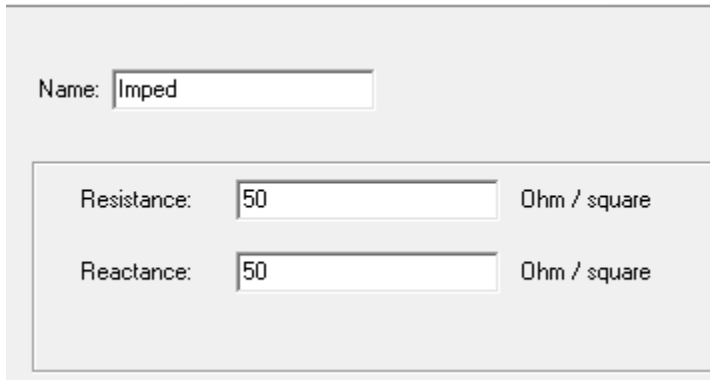
Technical Notes: Impedance Boundaries

Assign Impedance Boundaries for HFSS-IE

The impedance boundaries for HFSS-IE are derived assuming the fields decay to zero inside the object on which the boundary is assigned.

Note For HFSS-IE impedance boundaries assume the interior E-field decays to 0 and thus can not be used as an approximation for a transmission condition through a thin dielectric sheet.

1. Select a surface on which to assign the boundary, right-click **Assign Boundary>Impedance** to bring up the **Impedance Boundary** dialog box.
2. Set the fields in the dialog box and click **OK**.



Note HFSS-IE does not support infinite ground plane for impedance boundaries.

Related Topics

Zoom to Selected Boundary

Setting Default Boundary Base Names

Technical Notes*:* Impedance Boundaries

Assigning Radiation Boundaries

For Eigenmode, Driven Modal or Driven Terminal Designs

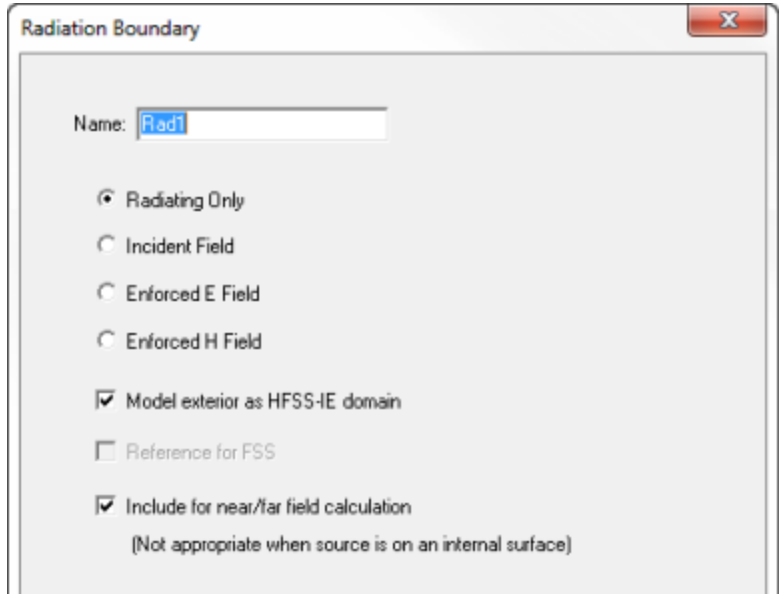
A radiation boundary is used to simulate an open problem that allows waves to radiate infinitely far into space, such as antenna designs. HFSS absorbs the wave at the radiation boundary, essentially ballooning the boundary infinitely far away from the structure. In HFSS, these are sometimes described as Absorbing Boundary Condition, or ABC.

A radiation surface does not have to be spherical, but it must be exposed to the background, convex with regard to the radiation source, and located at least a quarter wavelength from the radiating source. In some cases the radiation boundary may be located closer than one-quarter wavelength, such as portions of the radiation boundary where little radiated energy is expected.

Note Whenever additions/changes are made to radiation boundaries that affect fields, it invalidates those solutions that can possibly have fields. Meshes are not invalidated.

To assign a radiation boundary:

1. Select a surface on which to assign the boundary and click **HFSS >Boundaries>Assign>Radiation** to bring up the **Radiation Boundary** dialog box.



2. If your project uses a field solution from another source, your "target" project must have radiation boundaries with **Advanced Options** defined in order to specify where the fields from the "source" project enter. See the discussion here.
3. Designate the boundary as either:
   - **Radiating Only** - this refers to the original radiating surface properties (the default). If you select this option, the scattered field formulation is applied. (See Technical Notes). If

you do not select this radio button the total field formulation is applied. (See Technical Notes).

- **Incident Field** - the incident field source patterns are projected on these surfaces and are backed by ABC or PML. This is like a generalized space port. HFSS knows the incident field pattern, applies it to the port and expects a reflected field pattern which radiates back. In other words, it behaves as if you excited the project by a Norton or Thevenin generator using an impedance which is the free space wave impedance.
- If you select **Radiating Only** or **Incident Field**, you can also specify whether the surface is used as **Reference for FSS,** that is, as a Frequency Selective Surface - this surface become the input surface for calculations of the reflection/transmission coefficients. The other radiating surface automatically becomes output. Only one FSS can be defined in a given model. Using the **Incident Field** option together with **Reference for FSS** is advantageous for highly reflective and resonant structures. Reflection/Transmission coefficients for FSS designs can be viewed in the solution data panel as S-parameters or you can create an S-parameter report.
- **Enforced H Field** - this has the H tangential component of the incident field directly applied on these surfaces. It is an inhomogeneous Newmann BC. In other words, it behaves as if you excited the project by an ideal electric current source (enforced current). If you select Enforced Field, the **Use IE Formulation** option is grayed out.
- **Enforced E Field** - this has the E tangential component of the incident field directly applied on these surfaces. It is an inhomogeneous Dirichlet BC. In other words, it behaves as if you excited the project by an ideal magnetic current source (enforced current). If you select Enforced Field, the **Use IE Formulation** option is grayed out.
- **Model Exterior as HFSS-IE Domain**- this uses integral equation formulation, which is an exact transparent condition. By taking advantage of conformal radiation volumes to reduce the overall finite element solution domain utilization of the IE boundary will result in more efficient simulation for electrically large open boundary problems. It can be close to or on the structures, but for performance, 0.05 wavelength is recommended. If it is on a surface, you must turn off curvilinear elements. The IE boundary should enclose the entire structure by itself, or with an Infinite ground plane. (Compare Assigning IE Regions). If you model a radiation boundary as an IE domain, the solution setup cannot include calculations of derivatives on those regions.
- **Include for nearfield/far field calculation** - If you select **Radiation Only**, the **Include in near/far field calculation** option is grayed out but checked. That information passes to the solver to detect if the boundary is internal. This is necessary for incident wave problems. All Radiation Only surfaces are included in the near/far field calculation.

  When you select **Incident Field** or **Enforced Field**, you can designate that the surface is included in near/far field calculation by checking. If you do not include any surfaces in the near/far field calculation (whether as **Radiating Only**, or by checking), when you select default radiation surfaces at the near/far field calculation setup panel an error message

states that "No radiating surface has been selected."

Note

If you select either **Enforced Field** or **Incident Field** you should run a validation check in order to avoid an invalid setup. The setup is invalid if any of these surfaces are internal.

If you select either **Enforced Field** or **Incident Field** in most cases, you should avoid internal surfaces. In order to do that, internal objects with Enforced/Incident Field BC should be substructed to become background, or PEC material should be assigned to these objects to become "NoSolveinside".

Note

Do not define a surface that cuts through an object to be a radiation boundary.

In general, do not define the interface of two internal objects to be a radiation boundary. The only exception is when one object is a perfectly matched layer boundary (PML) and the other is the PML base object.

Related Topics

Zoom to Selected Boundary

Setting Default Boundary Base Names

Assigning PML Boundaries

Technical Notes*:* Radiation Boundaries

Getting Started Guides: A Dielectric Resonator Antenna

Getting Started Guides: UHF Probe

Getting Started Guides: Patch Antenna

Assigning PML Boundaries

A perfectly matched layer (PML) boundary is used to simulate materials that absorb outgoing waves. Setting up a PML boundary is similar to setting up a radiation boundary. You start by drawing a virtual object around the radiating structure. However, instead of placing a radiation boundary on its surfaces, you add PMLs to fully absorb the electromagnetic field.

HFSS can create PMLs automatically, or you can create them manually. Create PMLs automatically if the base object touching the PML is planar and its material is homogeneous. HFSS creates a separate PML object for each covered face. The PML boundaries are grouped in the **Project** tree under the **Boundaries** icon. Within these groupings, you can edit the radiation parameters (for example, as Incident Wave Port) in order to set up the right total field excitation based on the physical optics approach. PML radiation boundaries are not generated in Eigenmode projects.

In creating PMLs, you can select non rectangular sheet objects as long as they do not touch any other selected face. The underlying object does not have to be a box. If there are faces that touch, the touching faces must be locally box-like. You can assign variables to the dimension properties of the base object. Changing the variable values also changes the associated PMLs.

PML Compared to Radiation Boundaries

Compared to Radiation boundaries, which create absorbing boundary conditions (ABC), PMLs in general make it more difficult for the iterative solver to reach convergence compared to the same model with using ABCs. PMLs also require significantly more RAM. The advantages for PMLs are that they absorb a much wider range of waves in terms of frequency and direction. As a result, you can put PMLs much closer to the discontinuities. This gives a smaller model. ABCs efficiently absorb normal incident waves. You have to put ABCs far away enough from the discontinuities.

What do you want to do?

Create PMLs automatically.

Create PMLs manually.

Guidelines for Assigning PML Boundaries

Related Topics

Assigning Boundaries

Technical Notes: Boundaries

Zoom to Selected Boundary

Setting Default Boundary Base Names

Getting Started Guides: Radar Cross Section (RCS)

Creating PMLs Automatically

1. Draw a PML base object at the radiation surface.
2. Depending on the design you may select either faces of the object or the entire object.

   When you create a PML that completely covers an object, for example, an airbox around an antenna, you may find it convenient to select the object and have a boundary applied to all faces of the object.

You can select the faces of the PML base object to turn into PMLs. In these cases, select only external, planar faces and exclude faces defined as symmetry boundaries.

3. Click **HFSS** > **Boundaries**>**PML Setup Wizard**.

   The **PML Setup** wizard appears.

4. If you have selected a non-planar object for a PML or one that completely covers another object, you can select **Use Selected Object** *objectName* **as PML Cover**. If you have selected object faces, select **Create PML Cover Objects on Selected Faces**.

5. Type the thickness of each layer in the **Uniform Layer Thickness** text box. You can assign a variable as the thickness value.

   Note The layer thickness cannot be modified directly after PML objects have been created. If you want to be able to modify the thickness, assign a variable as the thickness value.

   If you do not assign a value, you can select **Use Default Formula** to have HFSS calculate a value for you based on geometrical analysis.

6. If the selected faces are on a box object. you can see the option to select **Create joining corner and edge objects**.

   Edge and corner PML objects will be created to join adjacent PML surfaces together, ensuring complete coverage.

7. Under **Base Face Radiation Properties**, click a radio button to specify one of the following:

   - **Radiating Only** - the radiation surface (default).
   - **Incident Field** - the incident field source patterns are projected on these surfaces and are backed by ABC or PML. This is like a generalized space port. HFSS knows the incident field pattern, applies it to the port and expects a reflected field pattern which radiates back. In other words, it behaves as if you excited the project by a Norton or Thevenin generator using an impedance which is the free space wave impedance.

     For **Radiating Only** or **Incident Field**, you can also specify whether the surface is used as **Reference for FSS**, that is, as a Frequency Selective Surface - this surface becomes the input surface for calculations of the reflection/transmission coefficients. The other radiating surface automatically becomes output. Only one FSS can be defined in a given model. Using the **Incident Field** option together with **Reference for FSS** is advantageous for highly reflective and resonant structures. Reflection/Transmission coefficients for FSS designs can be viewed in the solution data panel as S-parameters or you can create an S-parameter report.

     If you check **Reference for FSS**, the PML objects will stay visible.

8. Click **Next**.

   HFSS creates PMLs from the faces you selected. Names are automatically given to the layers. that start with *PML*, which is necessary for HFSS to recognize them as PMLs.

9. Specify how the PMLs terminate by selecting one of the following:

   a. **PML Objects Accept Free Radiation** if the PMLs terminate in free space.

- Then enter the lowest frequency in the frequency range you are solving for in the **Min Frequency** text box.

b. **PML Objects Continue Guided Waves** if the PMLs terminate in a transmission line.

- Then specify the propagation constant at the minimum frequency.

10. Specify the minimum distance between the PMLs and any of the radiating bodies in the **Minimum Radiating Distance** text box. You may choose to have HFSS calculate the value by clicking **Use Default Formula**. The default distance is based on the extent of base object geometry.

    The PML material characteristics depend on the cumulative effect of their near fields at the location of the PML surfaces.

11. Click **Next**.

    HFSS calculates the appropriate PML materials based on the settings you specified and the material of the base object, and assigns these materials to the objects in the PML group.

    A summary dialog box appears, enabling you to modify the settings you specified.

12. Click **Finish**.

Related Topics

Creating PML Boundaries Manually

Modifying PML Boundaries

Guidelines for Assigning PML Boundaries

Technical Notes: PML Boundaries

Assigning Radiation Boundaries

Creating PML Boundaries Manually

See Guidelines for Assigning PML Boundaries.

1. Draw the PML object at the radiation surface, and then select it.
2. In the **Properties** window, give the object a name with the prefix *PML*.

    Object names that start with *PML* are necessary for HFSS to recognize them as PMLs.

3. Click **HFSS**>**Boundaries**>**PML Setup Wizard**.

    The **PML Setup** wizard appears.

4. Select **Use Selected Object as PML Cover**.
5. Select the **Corresponding Base Object**, the object touching the PML, from the pull-down list.
6. Type the thickness of each layer in the **Uniform Layer Thickness** text box. You can assign a variable as the thickness value.

    If you do not assign a value, you can select **Use Default Formula** to have HFSS calculate a value for you based on geometrical analysis.

7. Select the orientation of the PML object, the direction of outward propagation, in the relative, or local, coordinate system.

8. Under **Base Face Radiation Properties**, click a radio button to specify one of the following:
   - **Radiating Only** - the radiation surface (default).
   - **Incident Field** - the incident field source patterns are projected on these surfaces and are backed by ABC or PML. This is like a generalized space port. HFSS knows the incident field pattern, applies it to the port and expects a reflected field pattern which radiates back. In other words, it behaves as if you excited the project by a Norton or Thevenin generator using an impedance which is the free space wave impedance.

     For **Radiating Only** or **Incident Field**, you can also specify whether the surface is used as **Reference for FSS**, that is, as a Frequency Selective Surface - this surface becomes the input surface for calculations of the reflection/transmission coefficients. The other radiating surface automatically becomes output. Only one FSS can be defined in a given model. Using the **Incident Field** option together with **Reference for FSS** is advantageous for highly reflective and resonant structures. Reflection/Transmission coefficients for FSS designs can be viewed in the solution data panel as S-parameters or you can create an S-parameter report.

     If you check **Reference for Frequency Selective Surface (FSS)**, the PML objects will stay visible.
9. Click **Next**.
10. Specify how the PML terminates by selecting one of the following:
    a. **PML Objects Accept Free Radiation** if the PML terminates in free space.
       - Enter the lowest frequency in the frequency range you are solving for in the **Min Frequency** text box.

    b. **PML Objects Continue Guided Waves** if the PML terminates in a transmission line.
       - Specify the propagation constant at the minimum frequency.
11. Specify the minimum distance between the PML and the radiating body in the **Minimum Radiating Distance** text box. You may choose to let HFSS calculate the value by clicking **Use Default Formula**. The default distance is based on the extent of base object geometry.

    The PML material characteristics depend on the cumulative effect of their near fields at the location of the PML surfaces.
12. Click **Next**.

    HFSS calculates the appropriate PML material based on the settings you specified and the material of the base object, and assigns this material to the PML.

    A summary dialog box appears, enabling you to modify the settings you specified.
13. Click **Finish**.

Related Topics

Guidelines for Assigning PML Boundaries

Modifying PML Boundaries

Technical Notes: PML Boundaries

Assigning Radiation Boundaries

Zoom to Selected Boundary

Setting Default Boundary Base Names

Guidelines for Assigning PML Boundaries

Keep the following guidelines in mind when assigning PML boundaries:

- When automatically creating PMLs, HFSS creates a new relative coordinate system for each PML object. This results in the z direction of the PML object coinciding with the normal direction of the base object's face.
- HFSS treats PMLs uniformly with regard to thickness. If the PMLs in your design vary in thickness, create a separate PML group for each thickness.

You should manually create a PML in the following situations:

- The base object is curved.

  HFSS calculates the PML material properties using the normal vector at the center of the base object's face. If the face is curved, the normal vector changes with position. The PML materials will only be good approximations if the normal vector at each point on the face is close to the normal vector at the face center.

  It is a good idea to segment the curved surface of the base object for greater accuracy. Create separate PMLs for each segment. Note that each segment's thickness is treated uniformly. The view angle of the segments should be no wider than 45 degrees. The smaller the angle of each segment, the greater the accuracy of the corresponding PML.
- The material of the corresponding base object touching the PML is not homogenous. An example is a metal-shielded microstrip line with a substrate. One PML could be drawn to terminate the microstrip and another could correspond to the substrate.

  Create as many PML objects as there are subsections of material properties in the base object.

Related Topics

Creating PML Boundaries Manually

Technical Notes: PML Boundaries

Assigning Radiation Boundaries

Modifying PML Boundaries

You can modify parameters of PML boundaries through the **PML Setup Summary** dialog. You can also modify the dimensions of PML boundaries by editing the History tree properties for the original object for which you assigned the PML boundaries.

To modify dimension properties for the original PML object:

1. Go to the History tree and open the hierarchy under the original PML base object.
2. Select the CreateBox command for each part of the geometry you want to modify.
3. In the Properties dialog for that geometry command, edit the properties for XSize, YSize, or ZSize as required.

The changes to the CreateBox parameters apply to the associated PML objects. Note that you can create variables to parameterize these properties.

To modify PML parameters:

1. Make sure that nothing is selected in the **3D Modeler** window.
2. Click **HFSS**>**Boundaries**>**PML Setup Wizard**.

   The **Summary** dialog box of the **PML Setup** wizard appears. By default the Show Objects in groups box is not checked. A table shows each PML Group, its thickness, and material status. Checking the box causes the table list the objects under each group.
3. If more than one group of PMLs were defined, select the PML group you want to modify from the table.
4. Modify the PML parameters.

   A Radio button lets you select either the **Free Radiation** minimum frequency and units, or the **Guided Wave** propagation constant at a minimum frequency.

   You can also specify a Minimum Radiating distance and units.
5. If you make changes, click **Update**.

   This performs the update and enables the **Recalculate Materials** button.
6. Click **Recalculate Materials** to apply the updates.

   HFSS automatically recalculates and assigns the appropriate PML materials to the objects in the PML group.
7. Click **Finish**.

Note If objects are modified after PMLs are created, the PML materials will be invalid and must be recalculated in the **PML Setup Wizard**. For example, if the material of the PML base object is modified, the associated PML materials must be recalculated in the **PML Setup Wizard**.

Related Topics

Assigning PML Boundaries

Assigning Radiation Boundaries

Zoom to Selected Boundary

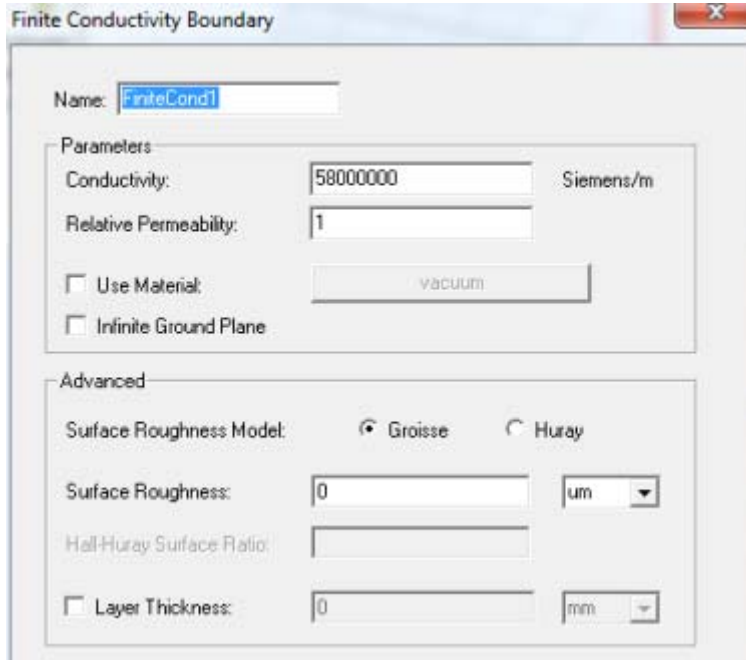
Setting Default Boundary Base Names

Assigning Finite Conductivity Boundaries

A finite conductivity boundary the behavior of the field at the object surface. The finite conductivity boundary is valid only if the conductor being modeled is a good conductor, that is, if the conductor's thickness is much larger than the skin depth in the given frequency range.

To assign a Finite Conductivity boundary:

1. Select a surface on which to assign the boundary and click **HFSS>Boundaries>Assign>Finite Conductivity** to bring up the **Finite Conductivity Boundary** dialog box.



2. Do one of the following:
   - Enter the conductivity in inverse ohm-meters, and then enter the permeability.

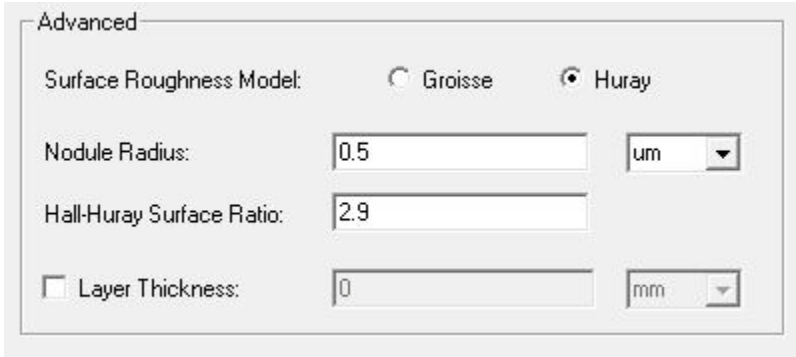
     To assign Finite Conductivity so that the boundary is spatially dependent (that is, in which the material properties change over the length), use the method described via this link.
   - Select **Use Material**, click the default material name button , and then choose a material from the material editor. The conductivity and permeability values of the material you select will be used for the boundary. Note that selecting a perfectly conducting material for a finite conductivity boundary triggers a validation error.
3. Select **Infinite Ground Plane** if you want the surface to represent an electrically large ground plane when the radiated fields are calculated during post processing.

   Note that if you select **Infinite Ground Plane**, the effect of the finite conductivity boundary will be incorporated into the field solution in the usual manner, but the radiated fields will be computed as if the lossy ground plane is perfectly conducting.

4. To select the Surface Roughness Model used for surfaces such as the interface between the conductor and the substrate for a microstrip line, select either Groisse or Huray.

   For the Groisse model , you specify a Surface Roughness parameter (traditional case) as a value (or variable) and units. The default is 0 um. Legacy projects use the Groisse model by default.

   For the Huray Model, you specify the Nodule radius value (or a variable), which describes the radius of copper spheres that model the surface roughness. The default is 0.5 um. Also for the Huray model, you specify the Hall-Huray Surface Ratio, a unitless quantity. The default is 2.9.



   (Using surface roughness with the Finite Conductivity boundary may be more intuitive than using a layered impedance boundary to model the effects.)

5. To specify a layer thickness, click the checkbox to enable the **Layer Thickness** field, and enter a value and select units.

Note You can assign a variable as the conductivity or permeability values or roughness model parameters.

Related Topics

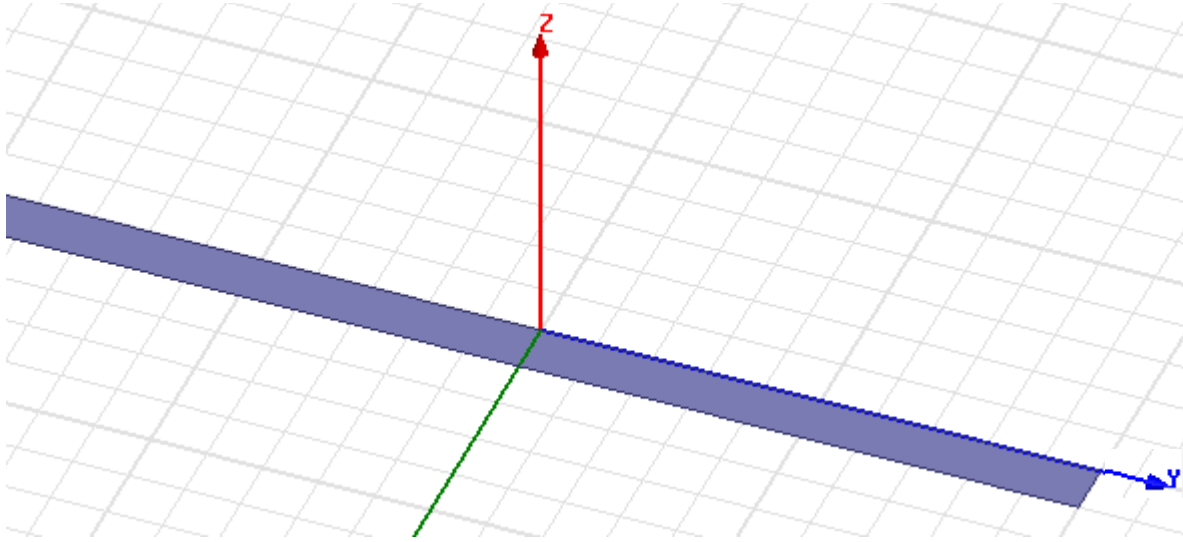
Zoom to Selected Boundary

Setting Default Boundary Base Names

Technical Notes: Finite Conductivity Boundaries

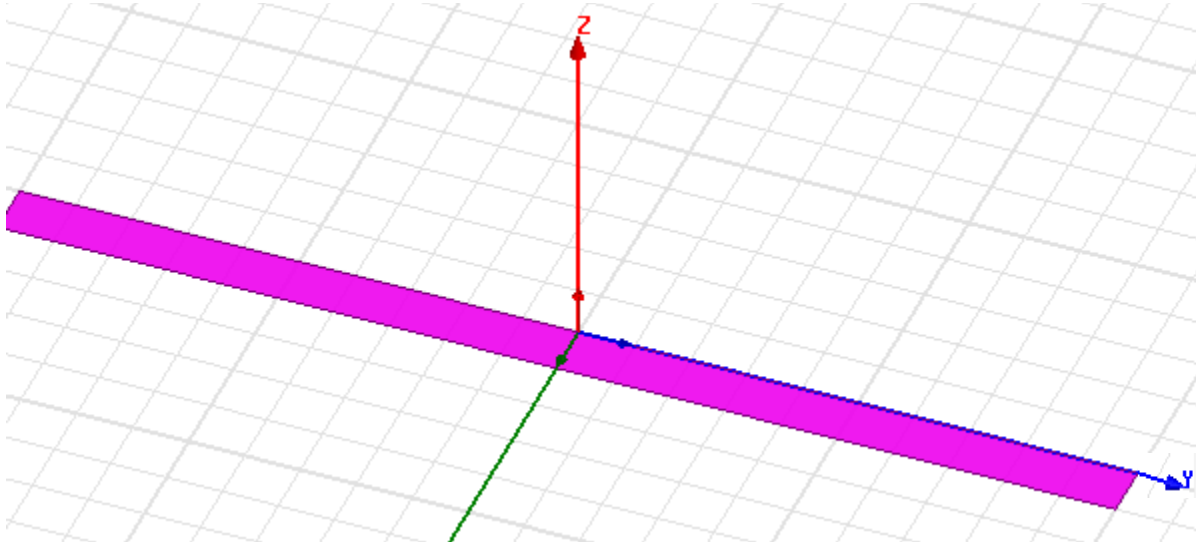
Spatially Dependent Boundaries in HFSS and HFSS-IE

HFSS and HFSS-IE supports spatially dependent materials, that is, materials whose properties change over their length. Consider a long rectangular sheet object that lies on the XY plane along the Y axis.

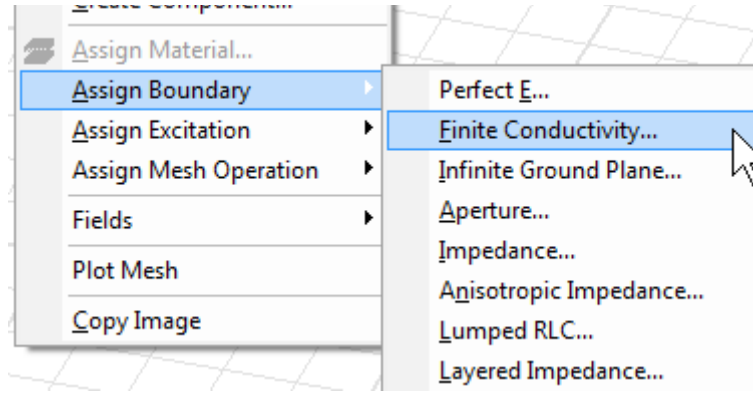


To assign **Finite Conductivity** such that the boundaries are spatially dependent follow these steps:

1. Select the line as shown in the figure below.

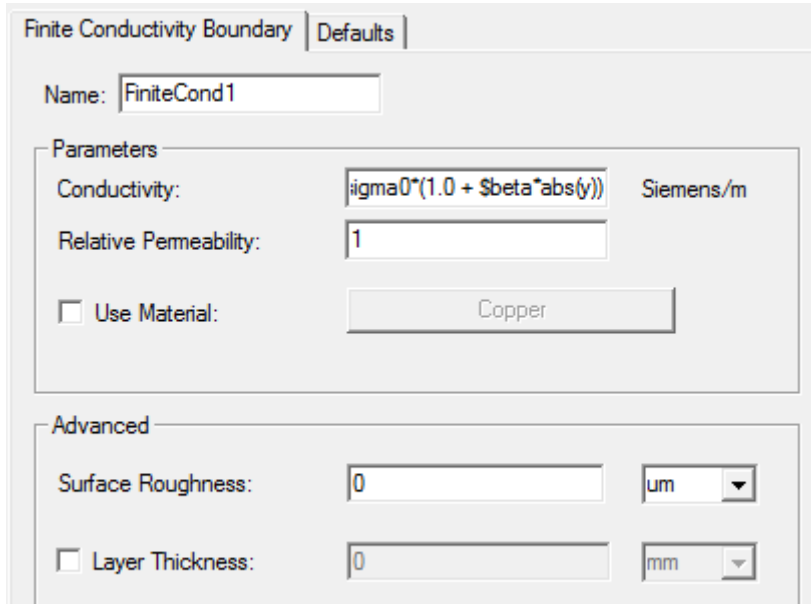


2. Right-click anywhere in the modeler and select **Assign Boundary** > **Finite Conductivity** as shown below.

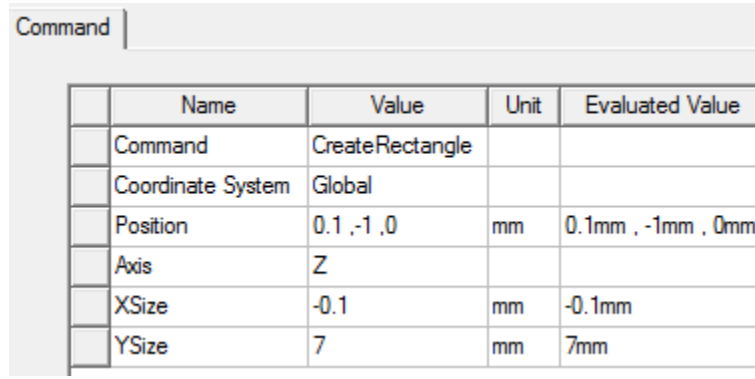


This causes the **Finite Conductivity Boundary** dialog box to appear as shown below.

3. Type the expression in the **Conductivity** field to include the spatially dependent material properties of the sheets. For example, in this design, the **Conductivity** field has the following expression: $sigma0*(1.0 + $beta*abs(y)), where $sigma and $beta are the predefined project variables and y specifies the local y co-ordinate associated with the sheet.



Note: Whatever model units you define, specify the expression in the **Conductivity** field in SI units. To verify the **Coordinate System** associated with the sheet, double-click the **CreateRectangle** command associated with the object from the History Tree to open the **Attribute** window as shown below.


Command

	Name	Value	Unit	Evaluated Value
	Command	CreateRectangle		
	Coordinate System	Global		
	Position	0.1 ,-1 ,0	mm	0.1mm , -1mm , 0mm
	Axis	Z		
	XSize	-0.1	mm	-0.1mm
	YSize	7	mm	7mm

Note The spatial dependency can only be expressed in Cartesian coordinates

Related Topics

Spatially Dependent Materials in HFSS

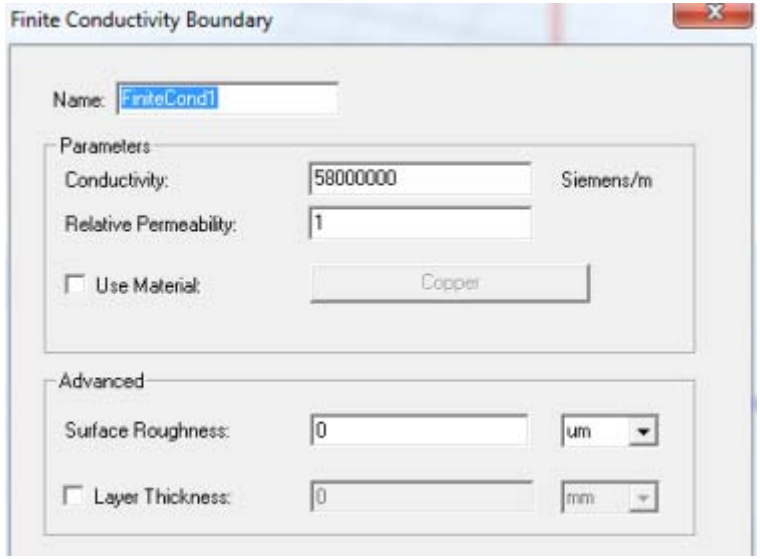
Assigning Finite Conductivity Boundaries in HFSS-IE

A finite conductivity boundary approximates the behavior of the field at the object surface.

The finite conductivity boundary is valid only if the conductor being modeled is a good conductor, that is, if the conductor's thickness is much larger than the skin depth in the given frequency range.

To assign a Finite Conductivity boundary:

1. Select a surface on which to assign the boundary and click **HFSS-IE>Boundaries>Assign>Finite Conductivity** to bring up the **Finite Conductivity Boundary** dialog box.



2. Do one of the following:
   - Enter the **Conductivity** and the **Relative Permeability**.

     To assign Finite Conductivity so that the boundary is spatially dependent (that is, in which the material properties change over the length), use the method described via this link.
   - Select **Use Material**, click the material name button, and then choose a material from the material editor. The conductivity and permeability values of the material you select will be used for the boundary. Note that selecting a perfectly conducting material for a finite conductivity boundary triggers a validation error.
3. To specify the roughness of surfaces such as the interface between the conductor and the substrate for a microstrip line, enter a value for **Surface Roughness** and select the units (default, microns) from the pull down menu.

   (This may be more intuitive than using a layered impedance boundary to model the effects.)
4. To specify a layer thickness, click the checkbox to enable the **Layer Thickness** field, and enter

a value and select units..

Note You can assign a variable as the conductivity or permeability values.

Related Topics

Zoom to Selected Boundary

Setting Default Boundary Base Names

Technical Notes: Finite Conductivity Boundaries

Getting Started Guides: A 20 GHz Waveguide Combiner

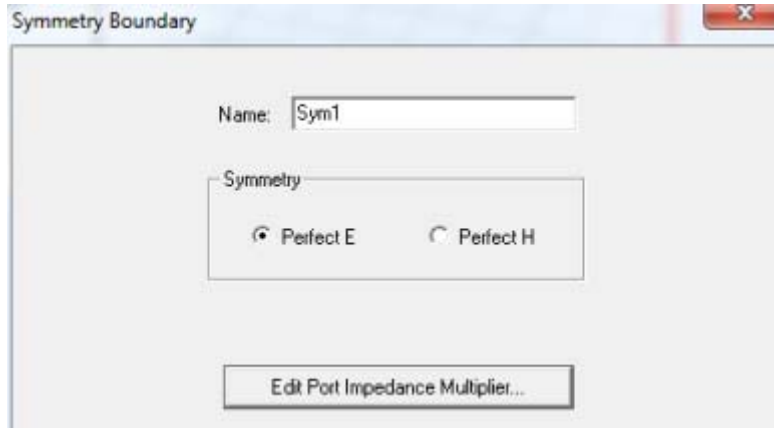
Assigning Symmetry Boundaries

For Driven Modal or Eigenmode Designs

A symmetry boundary represents a perfect E or perfect H plane of symmetry. Symmetry boundaries enable you to model only part of a structure, which reduces the size or complexity of your design.

To assign a Finite Conductivity boundary:

1. Select a surface on which to assign the boundary and click **HFSS>Boundaries>Assign>Symmetry** to bring up the **Symmetry Boundary** dialog box.



2. Select the type of symmetry plane the boundary represents: **Perfect E** or **Perfect H**.
3. Click **Impedance Multiplier**.

   If the design includes a port, you must adjust the impedance multiplier or the computed impedances will not be for the full structure.

   The **Port Impedance Multiplier** dialog box appears.
4. Type a value in the **Impedance Multiplier** box, and then click **OK**.

Related Topics

Technical Notes: Symmetry Boundaries

Setting the Impedance Multiplier

Technical Notes: Impedance Multipliers

Getting Started Guides: A Dielectric Resonator Antenna

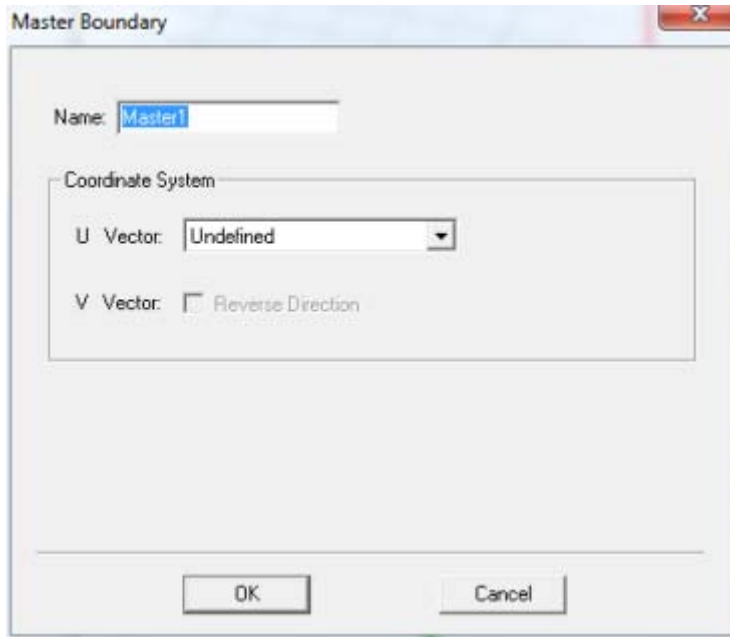
Getting Started Guides: A 20 GHz Waveguide Combiner

Assigning Master Boundaries

Master and slave boundaries enable you to model planes of periodicity where the E-field at every point on the slave boundary surface is forced to match the E-field of every corresponding point on the master boundary surface to within a phase difference. The transformation used to map the E-field from the master to the slave is determined by specifying a coordinate system on both the master and slave boundaries.

To assign a Master boundary:

1. Select a surface on which to assign the boundary and click **HFSS>Boundaries>Assign>Master** to bring up the **Master Boundary** dialog box.



2. You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. HFSS uses the U vector that you draw and the normal vector of the boundary face to calculate the Vvector. If necessary, you can reverse the direction of the V vector.
   a. Select **New Vector** from the **U Vector** pull-down list.

      The **Master Boundary** dialog box disappears while you draw the U vector.
   b. Select the U vector's origin, which must be on the boundary's surface, either by:
      - Clicking the point for the vector origin.
      - Typing the point's coordinates in the **X**, **Y**, and **Z** boxes.
   c. Select a point on the u-axis to indicate the U vector direction.

      The **Master Boundary** dialog box reappears and the model display shows the U vector

and V vector as red and blue arrows respectively.

d. If you need to reverse the direction of the V vector, select **Reverse Direction**.

HFSS will compute the E-field on this boundary and map it to the slave boundary using the transformation defined by the master and slave coordinate systems.

Related Topics

Technical Notes: Master and Slave Boundaries

Assigning Slave Boundaries

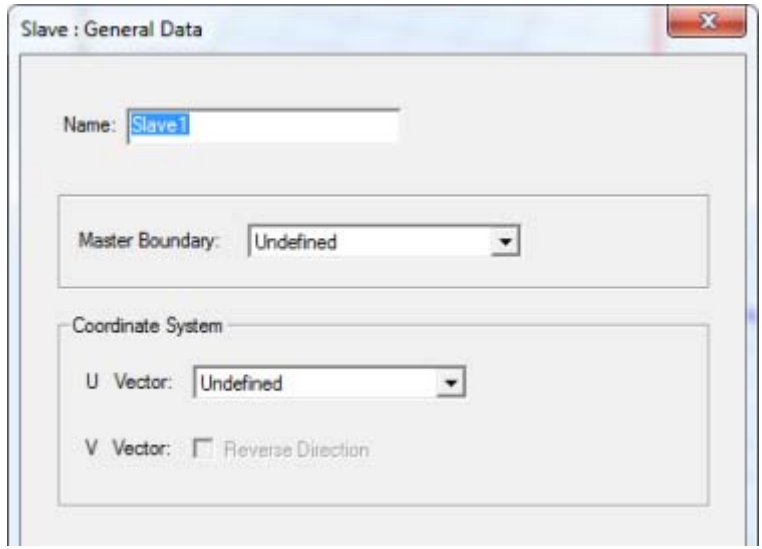
Getting Started Guides: Floquet Ports

Assigning Slave Boundaries

Master and slave boundaries enable you to model planes of periodicity where the E-field at every point on the slave boundary surface is forced to match the E-field at every corresponding point on the master boundary surface to within a phase difference. The transformation used to map the E-field from the master to the slave is determined by specifying a coordinate system on both the master and slave boundaries.

To assign a Slave boundary:

1. Select a surface on which to assign the boundary and click **HFSS>Boundaries>Assign>Slave** to bring up the **Slave Boundary** dialog box.



2. Select the corresponding master boundary from the **Master Boundary** pull-down list.

   If a master boundary has not yet been defined, return to make this selection when it has been defined.

3. You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. HFSS uses the U vector that you draw and the normal vector of the boundary face to calculate the V vector. If necessary, you can reverse the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.

      The **Slave Boundary** dialog box disappears while you draw the U vector.

   b. Select the U vector's origin, which must be on the boundary's surface, in one of the following ways:

      - Click the point for the vector origin.
      - Type the point's coordinates in the **X**, **Y**, and **Z** boxes.

   c. Select a point on the u-axis to indicate the U vector direction.

The **Slave Boundary** dialog box reappears and the model display shows the U vector and V vector as red and blue arrows respectively.

d. If you need to reverse the direction of the V vector, select **Reverse Direction**.

4. Click **Next**.
5. You have the option to relate the slave boundary's E-fields to the master boundary's E-fields in one of the following ways:
   - For driven designs, select **Use Scan Angles to Calculate Phase Delay** to enable the **Scan Angle** fields. Then enter **Phi** and **Theta** scan angles. These apply to whole model, in the global coordinate system. The phase delay is calculated from the scan angles; however, if you know the phase delay, you may enter it directly in the **Phase Delay** box below.

Note For Eigenmode problems, the **Use Scan Angles to Calculate Phase Delay** fields are disabled.

   - Select **Field Radiation**, and then enter the phase difference, or phase delay, between the boundaries' E-fields in the **Phase Delay** box. The phase delay applies only to this boundary.

Note You can assign a variable as the phi, theta, or phase delay values.

HFSS will compute the E-field on the master boundary and map it to this boundary using the transformation defined by the master and slave coordinate systems.

Related Topics

Technical Notes: Master and Slave Boundaries

Assigning Master Boundaries

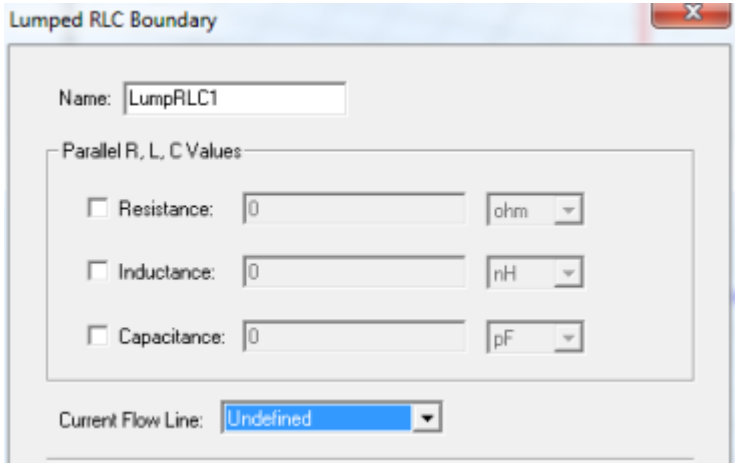
Getting Started Guides: Floquet Ports

Assigning Lumped RLC Boundaries

A lumped RLC boundary represents a parallel combination of lumped resistor, inductor, and/or capacitor applied to a surface. Multiple RLC boundaries can be used to model other circuit configurations. For example, a lumped RLC serial circuit connection can be modeled with three connected RLC surfaces: one with only resistance, one with only inductance, and one with only capacitance.

To create a lumped RLC boundary:

1. Select a surface on which to assign the boundary and click **HFSS** or **HFSS-IE>Boundaries>Assign>Lumped RLC** to bring up the **Lumped RLC Boundary** dialog box.



2. Select **Resistance**, **Inductance**, and **Capacitance** as needed and specify values and units for each selected element. Optionally, you can assign a variable to any of these values.
3. To specify where on the surface the current and voltage will be controlled, define a **Current Flow Line**. The selection field initially appears as **Undefined**. Select **New Line** to define a vector line on the boundary surface.

Note HFSS and HFSS-IE assume the lumped RLC is assigned to a rectangular face. If you assign a non-rectangular face, HFSS and HFSS-IE issue a warning, but proceed with the solution. Using a non-rectangular face can result in less accurate representation of the lumped RLC. See the technical notes on RLC boundaries for more information.

Related Topics

Define a Vector Line

Technical Notes: Lumped RLC Boundaries

Setting Default Boundary/Excitation Base Names.

Vector Line

To draw a vector line to indicate the current flow:

1. Select **New Line** from the Lumped RLC Boundary dialog's pull-down list.

   The dialog box disappears while you draw the vector line.
2. Select the start point in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes on the status bar.
3. Select the endpoint using the mouse or the keyboard.
4. Once the line has been defined, you can edit it as follows:

   Select **Swap End Points** from the dialog pull-down list to switch the start and endpoints of the line, reversing the line's direction.

Related Topics

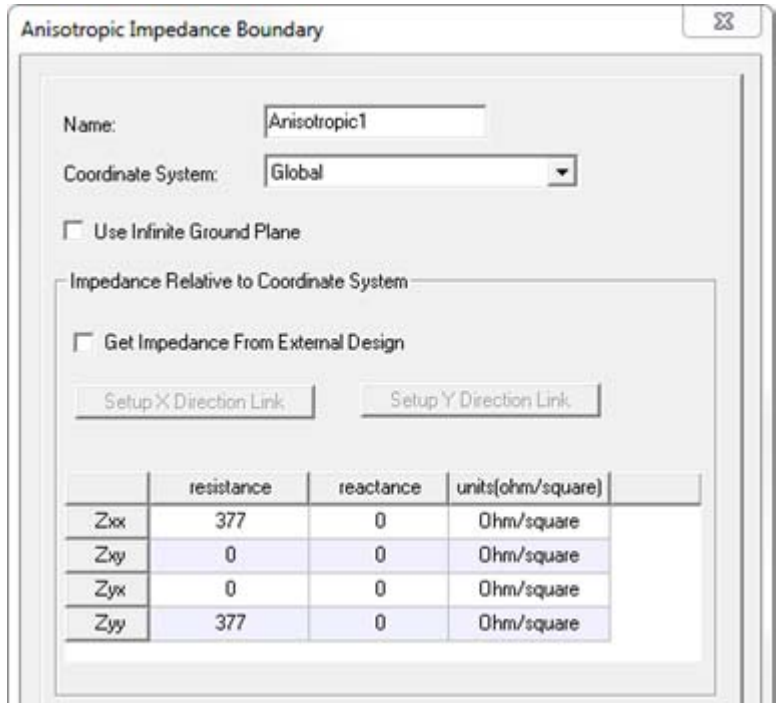
Assigning Lumped RLC Boundaries

Assigning Anisotropic Impedance Boundaries

Planar screens or grids of large extent with periodic geometry can be replaced by an Anisotropic impedance boundary. The boundary applies a homogeneous characteristic impedance to the surface in an effort to create an equivalent electrical representation of the geometric grid pattern. For legacy designs, this boundary replaces the Screening Impedance boundary.

To assign an Anisotropic Impedance boundary

1. Select a surface on which to assign the boundary and click **HFSS >Boundaries>Assign>Anisotropic Impedance** to bring up the **Anisotropic Impedance Boundary** dialog..



2. The Coordinate System drop-down menu lists the Global Coordinate system and any relative coordinate systems if you have defined them in the design. Select the Coordinate System that defines the anisotropic characteristic of the impedance boundary. (See Creating a Relative Coordinate System.)

Note An anisotropic boundary must not touch ports in a design.

3. Select **Infinite Ground Plane** if you want the surface to represent an electrically large ground plane when the radiated fields are calculated during post processing.
4. If you want to use an external design to define the impedance, click the Get Impedance from External Design checkbox to enable the **Setup Link** button. If you defining an anisotropic

impedance, there will be two buttons: **Setup X Direction Link** and **Setup Y Direction Link**.

5. If you have not selected **Use External Design**, the Resistance and Reactance fields are enabled.

   In these fields, you set the **Resistance** and **Reactance**. If Anisotropic Impedance is checked, the wizard shows Resistance and Reactance fields for X Axis alignment and Y axis alignment. These values can be numeric or can reference variables. Isotropy will be inferred if Zxy and Zyx are left as zero.

Related Topics

Setting Default Boundary/Excitation Base Names.

Assigning Material Property Types

Change the Orientation of an object

Creating a Relative Coordinate System

Creating a Face Coordinate System

Setting the Working Coordinate System

Modifying the Coordinate System Axes View

Technical Notes: Anisotropic Impedance

Get Impedance from External Design

To get impedance from an external design:

1. Click the **Setup** button to display the **Setup Link** dialog.

   For anisotropic impedance cases, select **Setup X Direction Link** button to chose a design which will define the impedance in the X direction. Then select **Setup Y Direction Link** button to chose a design which will define the impedance in the Y direction.

   The **Setup Link** dialog has three fields under the **General** tab: Project File, Design, and Solution.

2. Specify the Project file for the design that is the source. A browse button [...] lets you look through your file system. If you do not specify a project file, but select the current model, the current Project File is automatically filled in.
3. Specify the Design for the source. If the source is in the current design, you can select this from a drop down menu. If you select the current model, the Project File is automatically filled in.
4. Use the radio button to specify whether to save the source path relative to **The project directory of the source project** or **This project**.
5. Specify the Solution to use. A drop down list lets you select from the available solutions.

   The "Default" solution is the product dependent solution of the first Setup.

   That is the setup listed first in the source design's project tree (alphanumerical order). A product specific solution of this setup becomes the default solution. In most products, it is Last-

Adaptive. In a Transient solution type, it is "Transient."

Note The solution in the source design must provide data for the target design's adaptive frequency as well as its sweeps. That is, the adaptive frequency for the target design must be included in the sweep in the source design.

If necessary, you can open the source design and add an appropriate frequency point to an existing sweep.

6. Use the checkbox specify whether to **Simulate source design as needed**.
7. Use the checkbox to specify whether to **preserve the source design solution**. Note that in **Extractor mode**, the source project will be saved upon exit. **Extractor mode** means that the software is opened during the link solely for the purpose of solving.
8. Under the **Variable Mapping** tab, you can set the desired variable values in the source design. If the source and target designs contain same named variables, you can choose to Map Variable By Name. In this case, same named variables are mapped automatically.
9. Click OK to close the **Setup Link** dialog and return to the Anisotropic Impedance Boundary dialog.

Related Topics

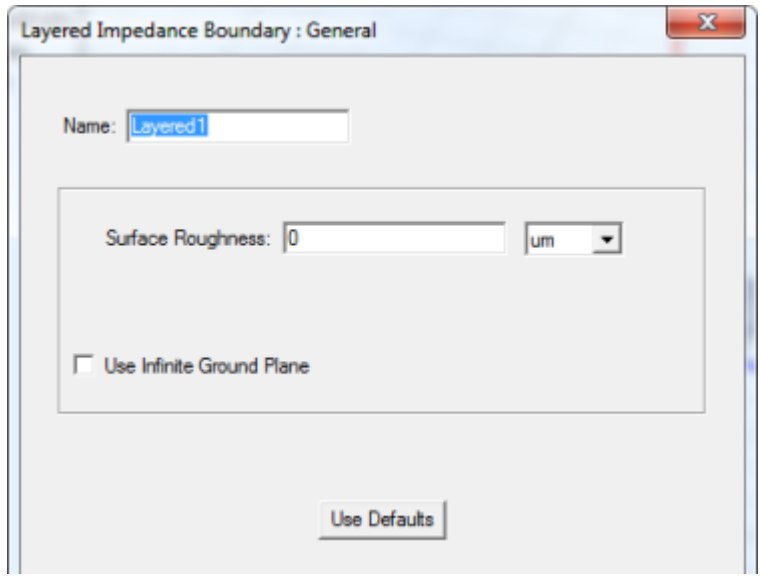
Assigning Anisotropic Impedance Boundaries

Assigning Layered Impedance Boundaries

A layered impedance boundary is used to model multiple thin layers in a structure as one impedance surface. The effect is the same as an impedance boundary condition, except that HFSS calculates the impedance of the surface based on data you enter for the layered structure. Surface roughness is also taken into account. The layered impedance boundary is supported for single-frequency solutions and for Discrete and Interpolating frequency sweeps. Eigenmode designs cannot contain design parameters that depend on frequency: for example, a frequency-dependent impedance boundary condition.

To assign a Layered Impedance boundary

1. Select a surface on which to assign the boundary and click **HFSS >Boundaries>Assign>Layered Impedance** to bring up the **Layered Impedance Boundary** dialog box.



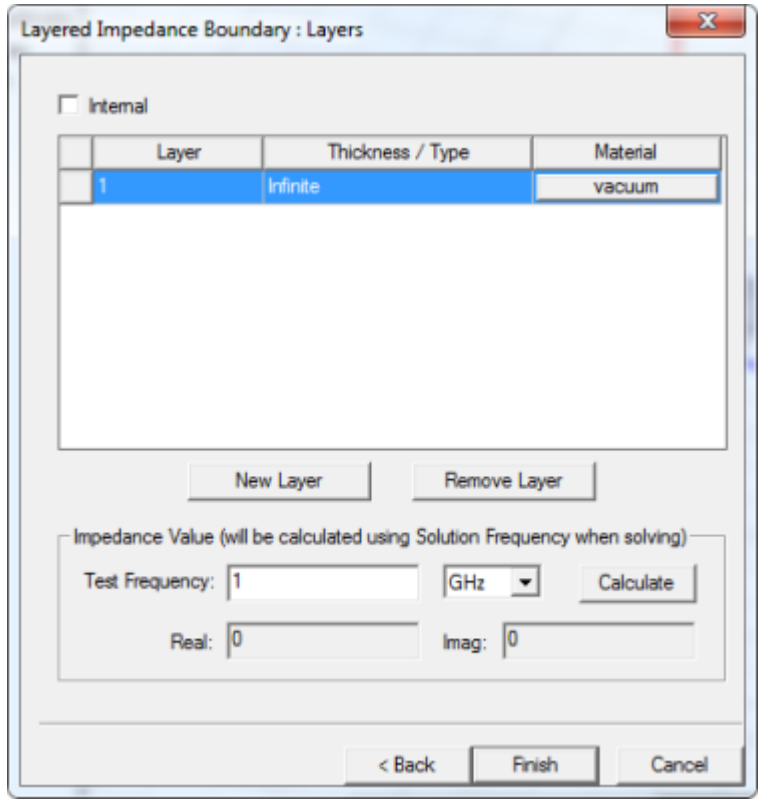
2. Enter the **Surface Roughness** for the layered structure.

   If the layered structure is internal to the design, enter the average surface roughness of the two outermost sides. You can assign a variable as this value.

3. Select **Infinite Ground Plane** if you want the surface to represent an electrically large ground plane when the radiated fields are calculated during post processing.

   For designs with layered impedance boundaries, only one infinite ground plane can exist in the design.

4. Click **Next** or the **Layers** tab, depending on the general option setting.



5. If the layered structure is external to the design, do the following:
   - By default, HFSS assumes the layered structure is external to the design; the outermost layer of the structure is listed. Select whether this layer is an **Infinite**, **Perfect E**, or **Perfect H** layer from the **Thickness/Type** list.

   If the layered structure is within the 3D model, do the following:
   a. Select the **Internal** option.
   b. Enter a thickness for the first layer in the **Thickness/Type** column. You can assign a variable as this value.
6. To change the first layer's material, click **vacuum** and follow the procedure for assigning a material.
7. To add a new layer to the structure:
   a. Click **New Layer**.

      The new layer is added at the end of the list.
   b. Enter a thickness for the layer in the **Thickness/Type** column. You can assign a variable

as this value.

c. To change the layer's material, click **vacuum** and follow the procedure for assigning a material.

8. Optionally, to reorder layers, click the first row square and drag the row to the desired position.
9. Optionally, to view the impedance values that will be calculated based on the data provided, do the following:
   a. Enter the frequency at which the solution is being solved in the **Test Frequency** text box.
   b. Click **Calculate**.

      The real and imaginary components of the HFSS-calculated layered impedance value appear.

Note A warning will be posted if a fast sweep is defined in a design that contains a layered impedance boundary, since the impedance may only be accurate for the center frequency.

Related Topics

Technical Notes: Layered Impedance Boundaries

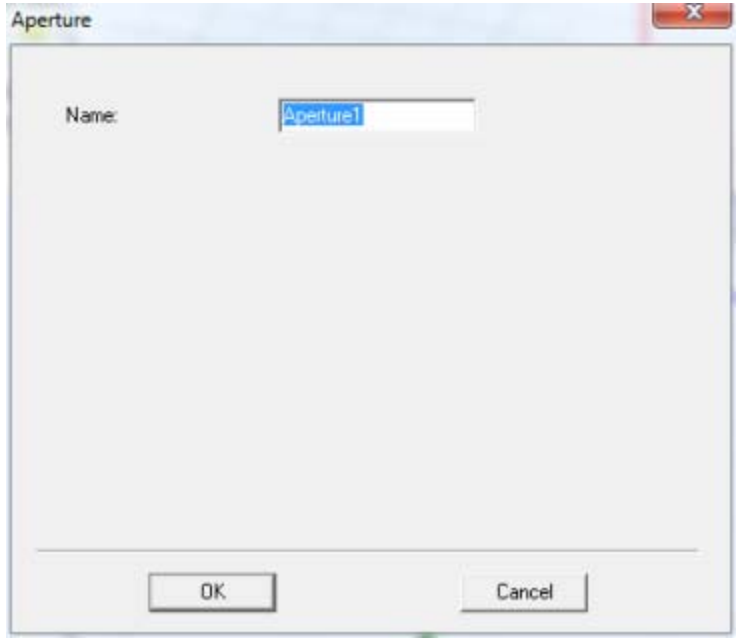
Assigning Aperture Boundaries in HFSS-IE

An aperture boundary represents holes in the design.

- Aperture boundary can only be assigned on sheet objects. This objected will be meshed as part of the solution process.
- Apertures can be assigned to any 2D objects (only objects: faces are not allowed). So if the 2D object touches the surface of 3D object, and you assign an aperture to the 2D object, you essentially create a hole on the 3D object. If the 2D object is touching ground plane, that means aperture is on ground plane.

To create an aperture in the infinite ground plane:

1. Create a 2D sheet object on the XY plane at the elevation of ground plane.
2. Select the object and right-click on **Boundaries>Assign>Aperature** to display the **Aperature** dialog.



3. Type the boundary's name in the **Name** text box or accept the default name. (To change the default base name to one of your choosing, see Setting Default Boundary/Excitation Base Names.)

Related Topics

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

Deleting Boundaries

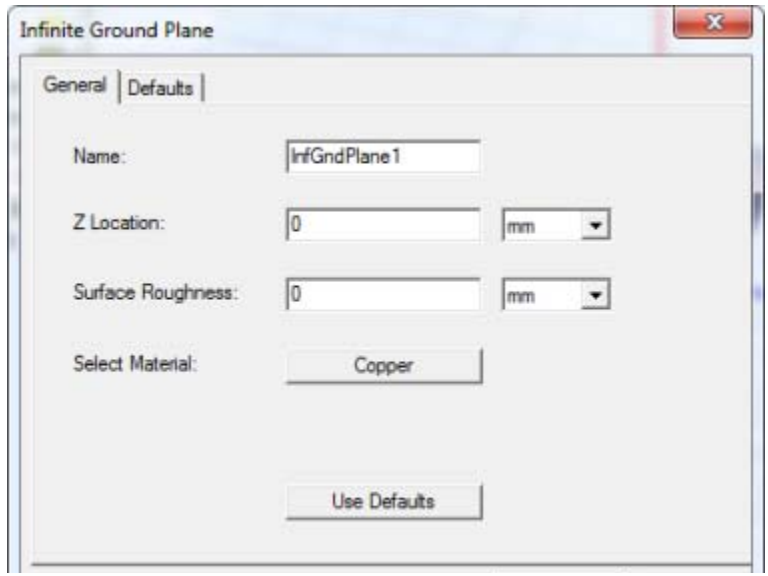
HFSS-IE Feature

Assigning Infinite Ground Plane Boundaries in HFSS-IE

An HFSS-IE design can contain an Infinite Ground Plane boundary aligned with the global XY plane. The Infinite Ground Plane will not be assigned to any geometry, since it will often exist in a location that does not have an appropriate face or sheet for assignment.

To assign an infinite ground plane geometry in HFSS-IE:

1. With no objects selected, right-click on **Boundaries** in the **Project** tree, and select **Assign>Infinite Ground Plane** to display the **Infinite Ground Plane** dialog.



2. You can accept the default name, or specify one.
3. Specify the Z location and the units.
4. Set the surface roughness and units.
5. A Select Material button displays the name of the default material. To change the material, click the button to display the materials dialog.

Related Topics

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

Deleting Boundaries

Assigning Materials

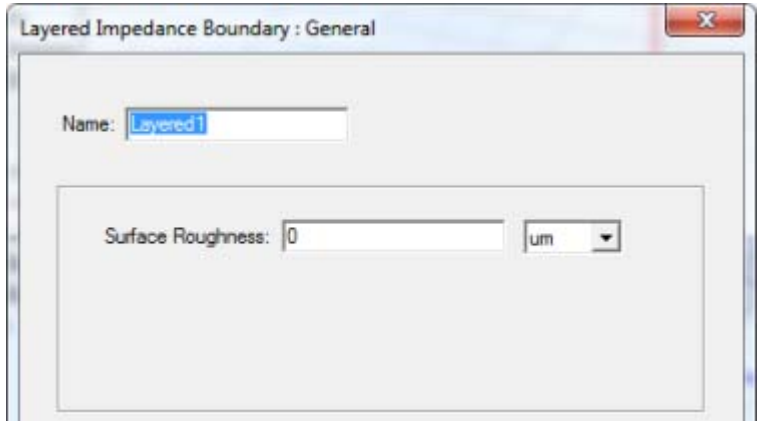
HFSS-IE Feature

Assigning Layered Impedance Boundaries in HFSS-IE

A layered impedance boundary is used to model multiple thin layers in a structure as one impedance surface. The effect is the same as an impedance boundary condition, except that HFSS calculates the impedance of the surface based on data you enter for the layered structure. Surface roughness is also taken into account. The layered impedance boundary is supported for single-frequency solutions and for Discrete and Interpolating frequency sweeps. Eigenmode designs cannot contain design parameters that depend on frequency: for example, a frequency-dependent impedance boundary condition.

To assign a Layered Impedance boundary

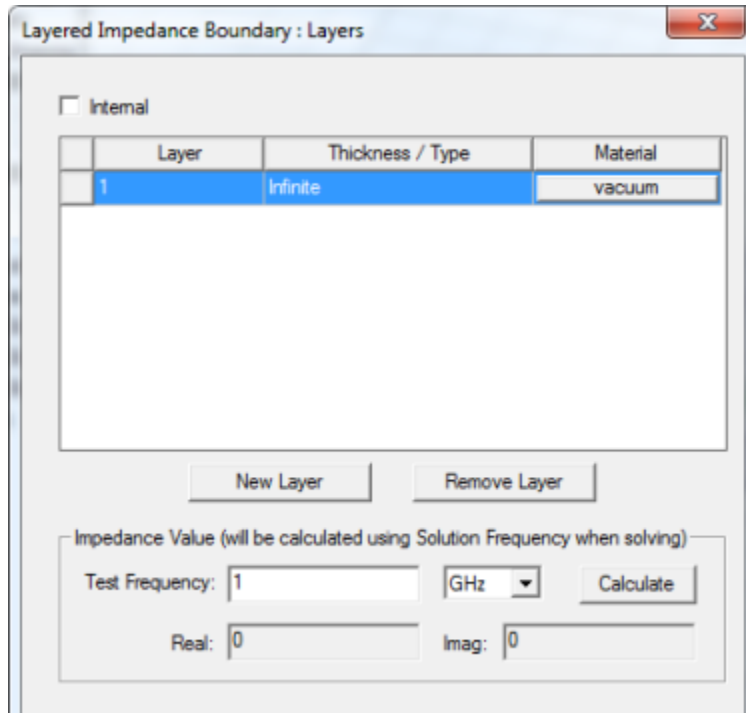
1. Select a surface on which to assign the boundary and click **HFSS-IE >Boundaries>Assign>Layered Impedance** to bring up the **Layered Impedance Boundary** dialog box.



2. Enter the **Surface Roughness** for the layered structure.

   If the layered structure is internal to the design, enter the average surface roughness of the two outermost sides. You can assign a variable as this value.

3. Click **Next** or the **Layers** tab, depending on the general option setting.



4. If the layered structure is external to the design, do the following:
   - By default, HFSS assumes the layered structure is external to the design; the outermost layer of the structure is listed. Select whether this layer is an **Infinite**, **Perfect E**, or **Perfect H** layer from the **Thickness/Type** list.

   If the layered structure is within the 3D model, do the following:

   a. Select the **Internal** option.

   b. Enter a thickness for the first layer in the **Thickness/Type** column. You can assign a variable as this value.

5. To change the first layer's material, click the material button for the row and follow the procedure for assigning a material.
6. To add a new layer to the structure:

   a. Click **New Layer**.

      The new layer is added at the end of the list.

   b. Enter a thickness for the layer in the **Thickness/Type** column. You can assign a variable as this value.

   c. To change the layer's material, click the material button for that row and follow the procedure for assigning a material.
7. Optionally, to reorder layers, click the first row square and drag the row to the desired position.

8. Optionally, to view the impedance values that will be calculated based on the data provided, do the following:
   a. Enter the frequency at which the solution is being solved in the **Test Frequency** text box.
   b. Click **Calculate**.

      The real and imaginary components of the HFSS-calculated layered impedance value appear.

Note A warning will be posted if a fast sweep is defined in a design that contains a layered impedance boundary, since the impedance may only be accurate for the center frequency.

Related Topics

Technical Notes: Layered Impedance Boundaries

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

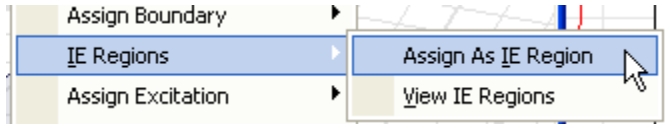
Deleting Boundaries

Assigning Materials

HFSS-IE Feature

Assigning IE Regions

For driven modal and driven terminal solutions you can assign objects or sheets as IE Regions to be solved with the IE Solver. This permits a hybrid simulation approach, using the advantages of the FEM and IE solvers.



- 3D objects must be dielectric or conducting based on bulk material conductivity. Some boundary conditions can also be assigned on the surfaces of 3D objects. These boundaries are: PEC, Finite Conductivity, Impedance, Layered Impedance, Lumped RLC, and Anisotropic Impedance.
- Sheet objects can also be allowed only if they have supported boundaries assigned. These boundaries are: PEC, Finite Conductivity, Impedance, Layered Impedance, Lumped RLC, or Anisotropic Impedance boundaries. They cannot be dielectric.
- Dielectric IE Region must either
  - be contained within a FEM solve inside object
  - have its surface covered by FEM solve inside objects
- Metallic IE Region must either
  - (Interior) Be contained within a Dielectric IE Region
  - (Exterior) Be outside entire FEM region.
- Faces of two IE Regions may not touch
- Dielectric IE Region may contain implicitly subtracted FEM objects (and imply solver handling like FEBI on their interface)
- Metallic IE Region may touch FEBI surface

Note

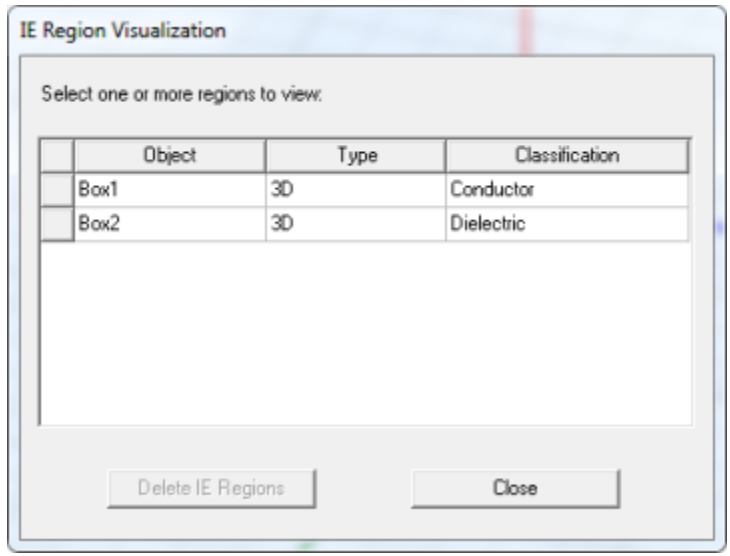
For designs like antennas mounted on a platform such as an aircraft or battleship, it is beneficial both in terms of memory and solution time to model antennas using the finite element method, while modeling remaining metallic structures as metallic IE regions. In these cases, a metallic IE region is actually in contact with a FEBI boundary, where appropriate boundary conditions are enforced at the interface between the two solvers.

In cases where an IE region is in contact with a FEBI boundary, it is recommended that the FEBI boundary be placed at least a third wavelength away so that the FEM domain can have sufficient space to perform its adaptive mesh refinement more accurately.

You must select an appropriate object or face in order to enable the menu. From the Project tree with the Boundaries icon selected, click **Assign>IE Region** or in the Modeler window right-click for the shortcut menu and select **IE Region>Assign as IE Region**.

If a selected object has already been assigned as IE Region, the menu presents **Unassign as IE Region**.

To view existing assignments , you can select the **Boundaries** icon in the Project Tree and right click **IE Region>View IE Regions,** or in the Modeler window right-click for the shortcut menu and select **IE Region>View IE Regions**. Select an item to highlight the object in the Modeler window.



Click **Delete IE Region** to remove the assignment from the selected object(s). This is equivalent to the **Unassign IE Region** command.

The Lambda refinement setting in the Solution Setup is ignored in a dielectric IE Region.

For designs with IE Regions, when generating a Report, you can select an IE Surface Fields Report type. When a design contains an IE Region, for post processing, with IE Surface Fields selected as the Field type in the Context area in Signal Calculator, you can select J and Q as Quantities as inputs for the Calculator Stack.

Related Topics

Assigning Radiation Boundaries (See Model Exterior as HFSS-IE Domain option)

Designating Infinite Ground Planes

To simulate the effects of an infinite ground plane in an HFSS design:

- Select the **Infinite ground plane** check box when setting up a perfect E, finite conductivity, or impedance boundary condition. For Impedance, Layered Impedance, and Finite Conductivity Boundary conditions, HFSS supports only one infinite boundary condition per design. For PEC, multiple antenna ground planes are supported.

This selection only affects the calculation of near- and far-field radiation during post processing. HFSS models the boundary as a finite portion of an infinite, perfectly conducting plane. If the infinite ground plane does not touch a radiation boundary, you will receive a warning. A finite sheet, which does not touch the radiation boundary condition cannot be an infinite ground plane.

Related Topics

Technical Notes: Infinite Ground Planes

Modifying Boundaries

To change the properties of a boundary, do one of the following:

- Double-click the boundary's icon in the project tree.

  The boundary's dialog box appears, in which you can edit its properties.
- Right-click the boundary in the project tree, and then click **Properties** on the shortcut menu.

  The boundary's dialog box appears, in which you can edit its properties.
- Click **HFSS**>**List**.

  The **Design List** dialog box appears, in which you can modify the properties of one or more boundaries.

Related Topics

Showing and Hiding Boundaries and Excitations

Deleting Boundaries

Deleting Boundaries

To delete *one boundary*:

1. Select the boundary you want to delete by selecting its icon in the project tree.
2. On the **Edit** menu, click **Delete**  .

To delete *all boundaries*:

- Click **HFSS**>**Boundaries**>**Delete All**.

You can also *delete one or more boundaries* in the **Design List** dialog box:

1. Click **HFSS** menu, click **List**.

   The **Design List** dialog box appears.
2. Under the **Boundaries** tab, click the row of the boundary you want to delete.
3. Click **Delete**.

Related Topics

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

Reassigning Boundaries

You can reassign a boundary to another surface. This is useful when you have modified objects with assigned boundaries, invalidating the boundaries. For example, if you unite two objects with assigned boundaries, the second object's boundary will become invalid because united objects maintain the characteristics of the first object selected. In this case, you would need to reassign the boundary or delete it

1. Select the object or object face to which you want to assign an existing boundary.
2. Click **HFSS>Boundaries>Reassign**.

   The **Reassign Boundary** window appears.
3. Select an existing boundary from the list, and then click **OK**.

   The boundary is reassigned to the object or object face.

Note When reassigning a boundary that includes vectors in its definition, HFSS attempts to preserve the vectors with the new assignment, but this is not always possible.

Alternatively, select the object or object face to which you want to assign an existing boundary. Right-click the existing boundary in the project tree, and then click **Reassign** on the shortcut menu.

Related Topics

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

Deleting Boundaries

Reprioritizing Boundaries

Each boundary you assign overwrites any existing boundary which it overlaps. You can change the priority of a previously assigned boundary to be greater than a more recently assigned boundary.

The order of boundaries is important because, for any given triangle of the mesh, only one boundary or excitation can be visible to the solvers. When two boundary definitions overlap, the one with the higher priority is visible to the solvers.

1. Click **HFSS>Boundaries>Reprioritize** to reprioritize boundaries.

   The **Reprioritize Boundaries** window appears. The order the boundaries and excitations appear in the list indicates the order in which they were defined. The lowest priority assignment appears at the top of the list.

   Ports are automatically placed at the bottom (highest priority) of the list; you cannot move a boundary to a higher priority than a port. Magnetic Bias Excitations (if any) have the lowest priority. Other boundaries and excitations appear between these two extremes.

2. Drag the boundary you want to change to the desired order of priority.

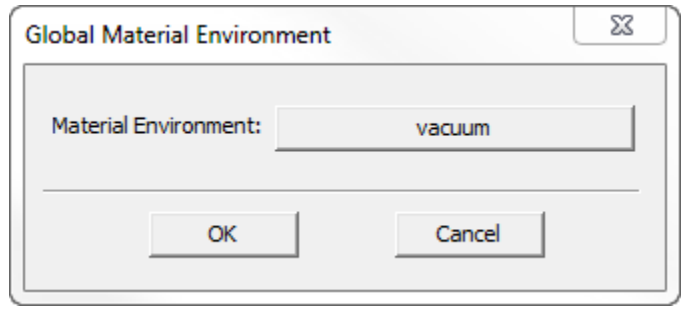
Note The order of boundaries and excitations in the project tree is alphabetical. The order does not correspond to the order of boundaries and excitations visible to the solvers.

Related Topics

Reviewing Boundaries and Excitations in the Solver View

Editing The Global Material Environment

For both HFSS and HFSS-IE, the **Boundaries>Edit Global Material Environment** command displays the **Global Material Environment** dialog. You can access the command via both the **Tools** menu and via right-click on **Boundaries** in the Project window for the shortcut menu.



By clicking the **Material** button, you can access the Select Definition dialog. This lets you work with the materials library.

This setting tells HFSS and HFSS-IE what material properties to use when calculating far fields. The default setting is vacuum. If you simulate an antenna underwater, for example, you should set the **GlobalMaterial Environment** to water.

Selecting anisotropic material is disabled because the solver doesn't support that.

For HFSS-IE, a validation checks if the selected material is dielectric. If the selected material is conducting, then an error message (the selected material is not dielectric. Please select a Dielectric Material) will be displayed.

See additional discussion of the effect this setting has on calculations in Global Material Environment.

Related Topics

Viewing and Editing Material Attributes

Far Field Wave

Duplicating Boundaries and Excitations with Geometry

To duplicate a boundary or excitation when its geometry is pasted or duplicated:

1. Click **Tools**>**Options**>**HFSS Options**.
2. Under Boundaries, select **Duplicate boundaries with geometry**.

   All boundaries and excitations will be duplicated with their associated geometries until you choose to clear this option.

Hint Use this option to copy and paste boundaries. For example:

1. Select the face to which you want to assign the boundary.
2. Click **Modeler**>**Surface**>**Create Object From Face**.
3. Assign the boundary to the new face object.
4. Copy and paste the new face object to copy and paste the boundary.

Related Topics

Copying and Pasting Objects

Showing and Hiding Boundaries and Excitations

You can choose to show or hide a boundary or excitation's geometry, name, or vectors, in the active view window or in all view windows.

What do you want to do?

Show or hide a boundary or excitation in the active view window.

Show or hide a boundary or excitation in every view window.

Showing and Hiding Boundaries and Excitations in the Active View Window

1. On the **View** menu, click **Active View Visibility**  or select the **Active View Visibility** icon in the toolbar.

   The **Active View Visibility** dialog box appears.
2. Select the tab for the objects you want to show or hide. The dialog contains tabs for 3D Modeler objects, Color Key objects, Boundaries, Excitations, and Fields Reporter objects.
3. Under the tab you need, select the **Visibility** option for the objects you want to show in the active view window.
4. Click the **Boundaries** tab if you want to show or hide boundaries.

   Note Hiding boundaries also turns off a check for boundary overlaps during boundary assignment. In the case of very large models with many boundaries, hiding boundaries can prevent delays during boundary assignment. Full model validation will subsequently check for boundary overlaps.

   Click the **Excitations** tab if you want to show or hide excitations.

   - For designs with large numbers of objects, you can resize the dialog for easier selection.
   - By default, objects are listed in alphabetical order. You can invert the order by clicking the Name bar above the Name fields. A triangle in the bar indicates the direction of the listing.
   - You can also use the **Name** field to type in an object name and apply the visibility via the **Show** and **Hide** buttons.

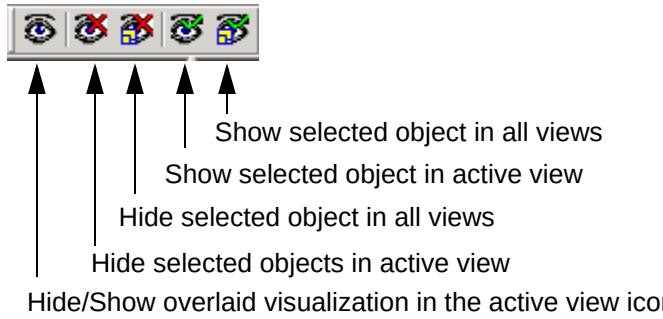
   The objects you select and designate as Visible (by selecting the property or using Show) appear.
5. Clear the **Visibility** selection of boundaries or excitations that you want to hide from view.

   The boundary or excitation will only be visible in the active view window if it is selected.
6. Select the **Visibility** option for boundaries or excitations that you want to show in the active view window.

   The boundary or excitation will be visible in the active view window when it is selected or when it is not selected.

   You can also use the toolbar icons to **Show/Hide selected objects in all views** and **Show/Hide**

selected objects in active views.



Showing and Hiding Boundaries and Excitations in Every View Window

1. Click **HFSS>Boundaries>Visualization** if you want to show or hide boundaries.

   Click **HFSS>Excitations>Visualization** if you want to show or hide excitations.
2. Clear the **View Geometry**, **View Name**, or **View Vector** selection of boundaries and excitations that you want to hide from view. Select the options you want to show.

   The options affect all view windows.

Note Hiding boundaries also turns off a check for boundary overlaps during boundary assignment. In the case of very large models with many boundaries, hiding boundaries can prevent delays during boundary assignment. Full model validation will subsequently check for boundary overlaps.

Reviewing Boundaries and Excitations in the Solver View

After you have assigned all the necessary boundaries and excitations to a model, you should review their order of priority according to the HFSS solver. Reviewing the solver's view of the model's boundaries and excitations enables you to verify that their order during the solution process will be as you intended.

To check the solver's view of boundaries and excitations:

1. On the **HFSS** menu, click **Boundary Display (Solver View)**.

   HFSS generates an initial mesh and determines the locations of the boundaries and excitations on the model.

   The **Solver View of Boundaries** window appears, which lists all the boundaries and excitations for the active model in the order specified in the **Reprioritize Boundaries and Excitations** dialog box.

2. Select the **Visibility** option for the boundary or excitation you want to review.

   The selected boundary or excitation will appear in the **3D Modeler** window in the color it has been assigned.

   - **Visible to Solver** will appear in the **Solver Visibility** column for each boundary or excitation that is valid.
   - **Overridden** will appear in the **Solver Visibility** column for each boundary or excitation that will be ignored by the solver as a result of it overlapping an existing boundary or excitation with a higher priority.

3. Verify that the boundaries or excitations you assigned to the model are being displayed as you intended for solving purposes.
4. If the order of priority is not as you intended, reprioritize the boundaries and excitations.

Related Topics

Technical Notes: Default Boundary Assignments

Setting Default Values for Boundaries and Excitations

When assigning a boundary or excitation, many of the fields in the boundary and excitation dialog boxes have default values associated with them. These default values are initially set by HFSS, but can be overridden.

To modify the default values associated with a specific boundary or excitation type:

1. Assign a boundary or excitation.
2. Modify any default values.
3. Close the boundary or excitation's dialog box.
4. Re-open the new boundary or excitation's dialog box. It now includes a **Defaults** tab.
5. Under the **Defaults** tab, click **Save Defaults**.

   The values assigned to this boundary are saved as the default values and will be assigned when new boundaries of this type are created.
6. Optionally, click **Revert to Standard Defaults**.

   The default values you set for this boundary type will be cleared and will revert to the default values set by HFSS.

Note For PML boundaries, the defaults are set via a formula, rather than a value.

Related Topics

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

Deleting Boundaries

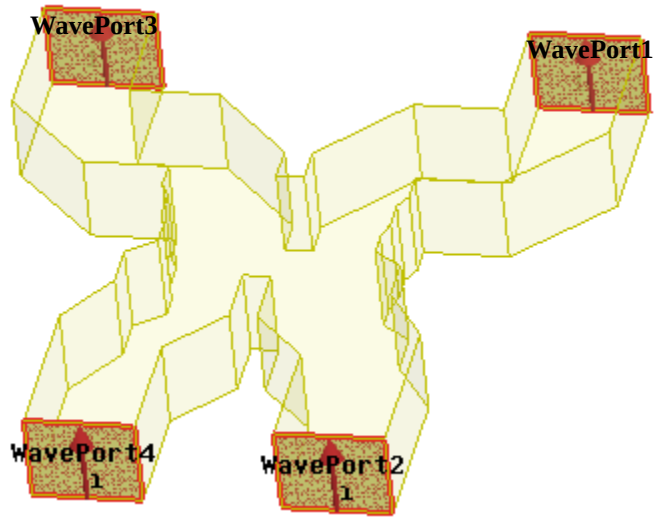
9

Assign Excitation

Excitations are sources of electromagnetic fields in the design. HFSS has various options to generate incident fields that interact with a structure to produce the total fields. Some of these excitations are local sources residing within the structure such as waveports and voltage sources, while other excitations such as plane waves are created from local sources away from the structure.The available excitations depend upon the product. They are as follows:

- HFSS Excitations
- HFSS-IE Excitations
- HFSS Transient Excitations

Excitations on a Waveguide Combiner



HFSS Excitations

You can assign the following excitations to an HFSS design:

Wave Port	Represents the *external* surface through which a signal enters or exits the geometry. It is effectively a semi-infinite waveguide attached to the model. This waveguide has the same cross-section and material properties as the port. Wave ports are placed on this interface to provide a means to link the model device to the external world.
Lumped Port	Represents an *internal* surface through which a signal enters or exits the device. It is effectively a lumped element for exciting the device and measuring S-parameters.
Terminal	A terminal is defined by one or more conductors in contact with the port. HFSS treats microwave structures as a black box that may have one or more terminals, each of which has a voltage/current pair. Terminals are assigned automatically.
Floquet Port	Floquet Ports are used exclusively with periodic structures defined by Master-Slave boundaries. They contain plane waves whose frequency, phasing, and the geometry of the periodic structure determine the propagation direction. Chief examples are planar phased arrays and frequency selective surfaces when these may be idealized as infinitely large and analyzed using a unit cell.
Incident Wave	Represents a propagating wave impacting the geometry.
Voltage Source	Represents a constant electric field across feed points.
Current Source	Represents a constant electric current across feed points.
Magnetic Bias	Used to define the net internal field that biases a saturated ferrite object.

HFSS-IE Excitations

You can assign the following types of excitation on an HFSS-IE design:

Lumped Port	Represents an *internal* surface through which a signal enters or exits the device. It is effectively a lumped element for exciting the device and measuring S-parameters.
Terminal	A terminal is defined by one or more conductors in contact with the port. HFSS-IE treats microwave structures as a black box that may have one or more terminals, each of which contains a voltage/current pair. Terminals are assigned automatically.
Plane Incident Wave	Represents a wave that propagates in one direction and is uniform in the directions perpendicular to that of its propagation.
Far Field Wave	A Far field wave is sufficiently far (usually more than a wavelength distance) from an antenna to approximate as a plane wave.
Near Field Wave	A Near Field wave is close enough to the antenna source for near field effects to occur, typically within a wavelength.

HFSS Transient Excitations

You can assign the following types of excitation. For HFSS Transient designs, the properties for each excitation include a **Transient** tab, enabling you to designate each excitation as Active or Passive.

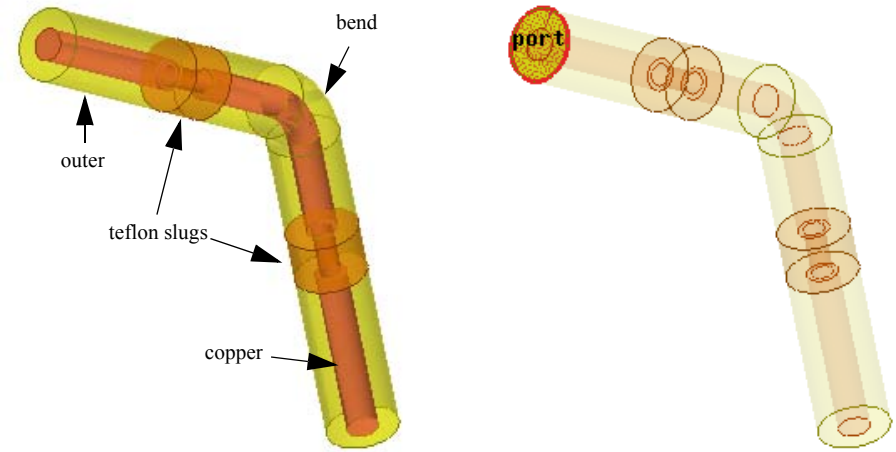
Wave Port	Represents the *external* surface through which a signal enters or exits the geometry. It is effectively a semi-infinite waveguide attached to the model. This waveguide has the same cross-section and material properties as the port. Wave ports are placed on this interface to provide a means to link the model device to the external world.
Lumped Port	Represents an *internal* surface through which a signal enters or exits the device. It is effectively a lumped element for exciting the device and measuring the S-parameters.
Terminal	A terminal is defined by one or more conductors in contact with the port. HFSS treats microwave structures as a black box that may have one or more terminals, each of which contains a voltage/current pair. Terminals are assigned either manually or automatically.
Plane Wave	Represents a wave that propagates in one direction and is uniform in the directions perpendicular to that of its propagation.
Voltage Source	Represents a constant electric field across feed points.
Current Source	Represents a constant electric current across feed points.

Related Topics

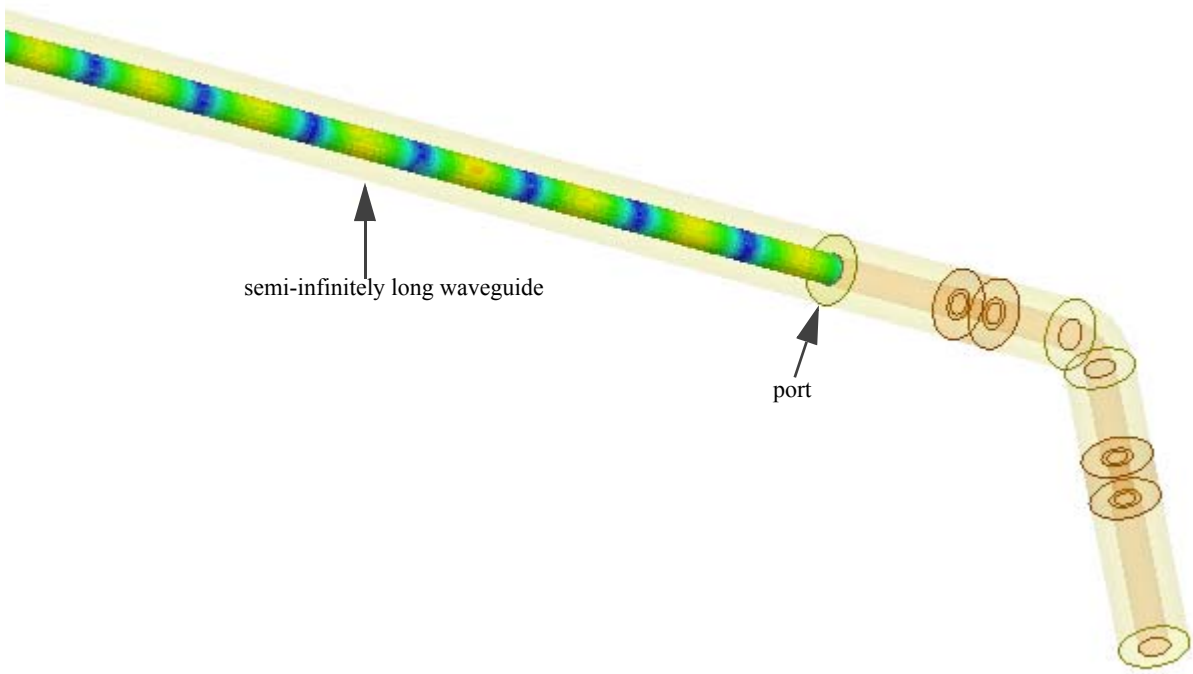
Technical Notes: Excitations in the Time Domain

Wave Ports

The objective of this section is to provide a thorough description of waveports. You must know what a wave port represents in HFSS to understand its capability. To illustrate we will use an HFSS model of a coaxial bend as an example where the waveport is assigned on the outer face.



HFSS treats this wave port as though a waveguide or a transmission line of the exact same cross-section (in this case the cross-section of the coaxial bend) and material properties, comes from infinity and ends at the port as shown in the figure below.



The properties and the cross section of the waveguide or the transmission line determine the natural field patterns called modes that excite the model and the HFSS port solver determines the propagating modes that the waveguide or transmission line will carry.

Note: A wave port can be placed internal to a model as long as it is backed by a PEC object.

Related Topics

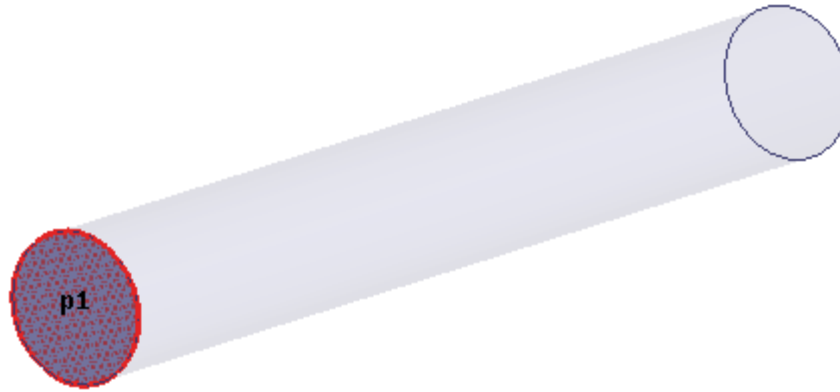
Waveport Placement

Waveport Size

Wave ports are used for exciting transmission lines and waveguide structures. Wave ports that are assigned on waveguide structures are naturally defined by the cross-section of the waveguides. However, for transmission lines (i.e. microstrip, CPW, slotline etc.) ports should be defined carefully. Sometimes transmission lines are part of a large PCB structure which makes it your responsibility to define the port size properly. This section provides guidelines for the appropriate port sizes for transmission lines. Examples of different waveguide and transmission line structures are as follows:

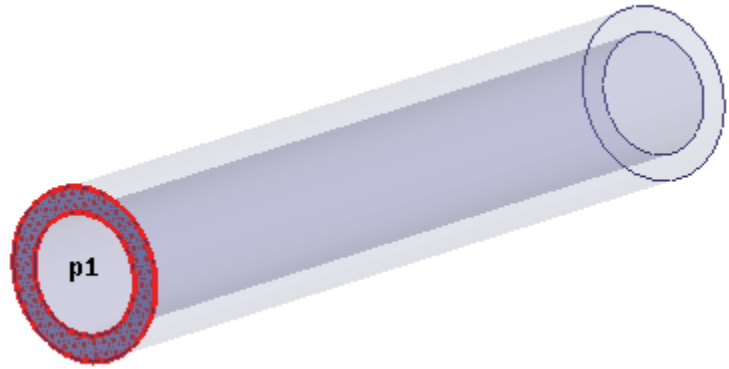
Circular Waveguide

An HFSS model of a cylindrical waveguide of uniform circular cross-section along its length excited with a waveport *p1* is shown below. You do not need to define the port size because it is naturally defined by the cross-section.



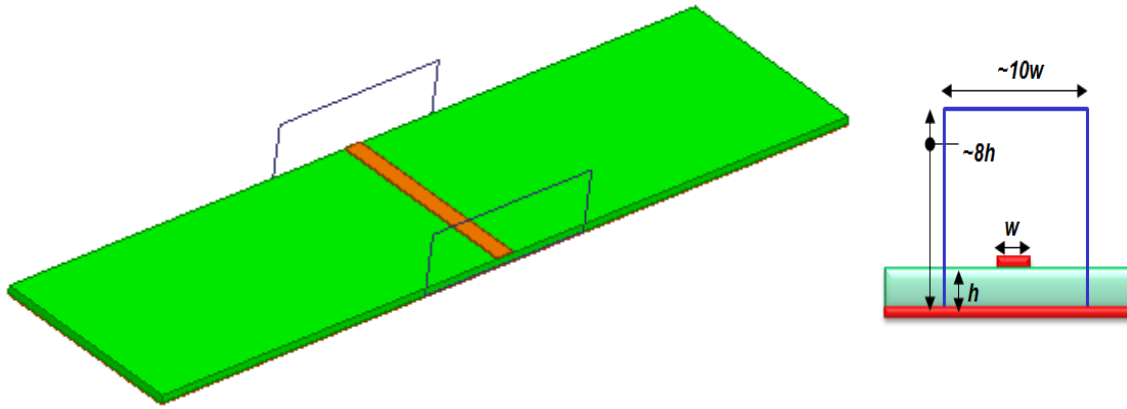
Coaxial Cable

The figure below represents a coaxial cable. You do not need to define the port size as it is naturally determined by the inner and outer radii of the shield.

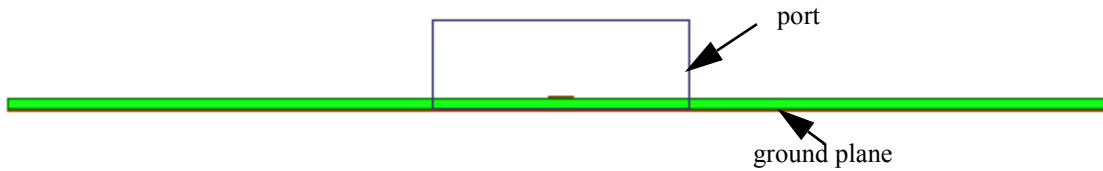


Microstrip Transmission Line

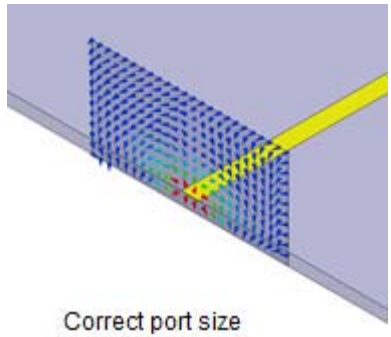
The figures below show an HFSS model of a microstrip transmission line and guidelines for setting the waveport dimensions.

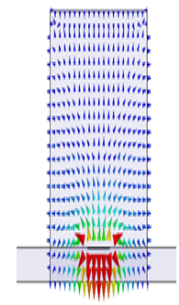


See the HFSS model at scale in the figure below. The bottom of the port touches the ground plane of the microstrip.



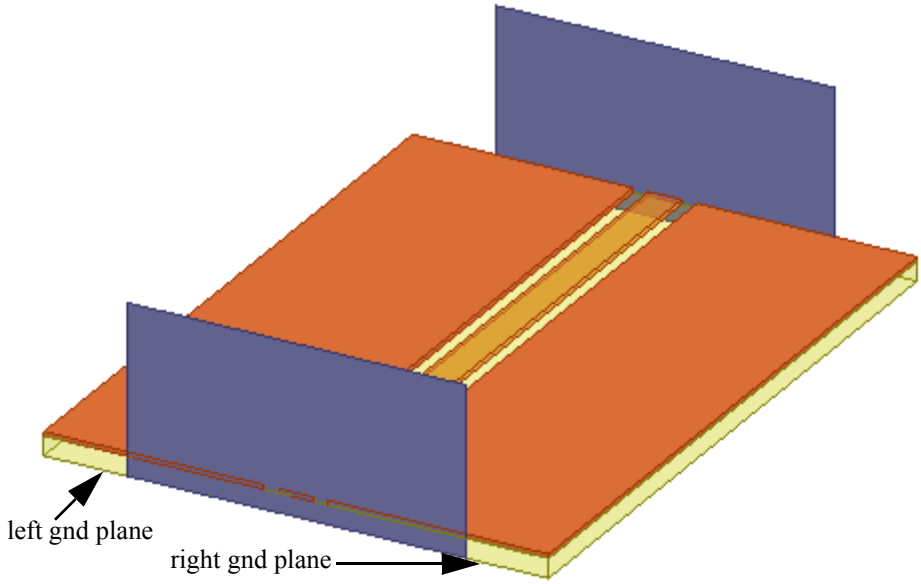
The port width affects the port impedance and the propagating modes. If the defined port is too narrow more fields will couple to the side walls. The height of the port is affected by the permittivity of the substrate. If the permittivity is too high less fields will propagate in the air, so the waveport can be made shorter.


Correct port size


Port too narrow
(fields coupled to sidewalls)

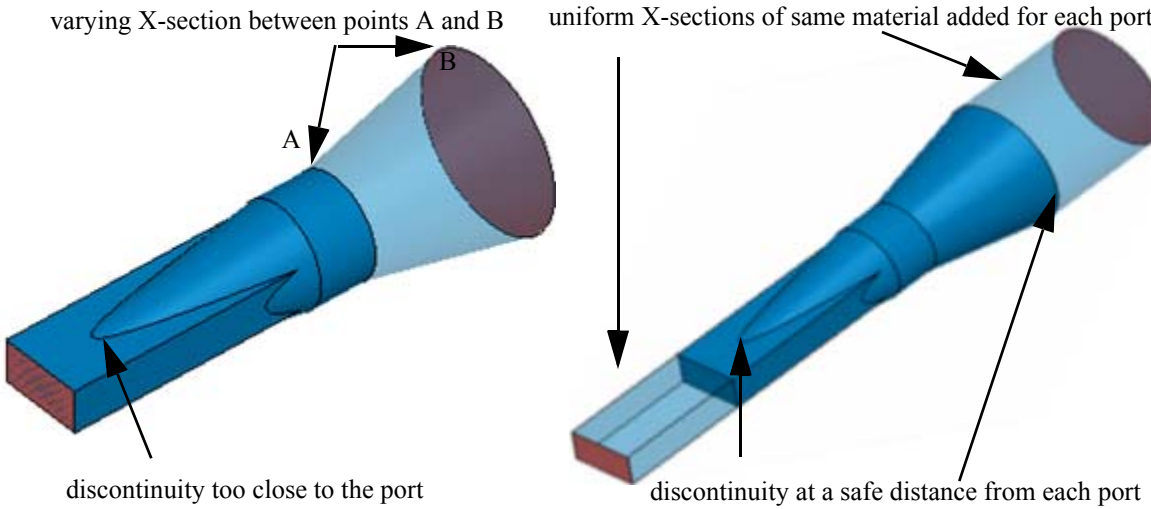
Coplanar Transmission Line

The figure below shows an HFSS model of a coplanar transmission line. The left and right edges of each port must touch the left and right ground planes. We recommend that you make the port size 8h x 10w where "w" represents the width of the trace and "h" represents the height of the substrate.



Waveport Placement

HFSS treats each port that you define to be connected to a semi-infinitely long waveguide or transmission line that has the same cross-section and properties as the wave port. The placement of the wave port is critical because it can affect the accuracy of the solutions. If the wave port is close to a discontinuity due to change in shape, dimension, or material of a structure the resulting 3D field can be a superposition of propagating and non-propagating modes.



Hence, the first discontinuity should be at an appropriate distance away from the port surface. This allows any energy reflected into an evanescent mode of the port to decay before reaching the port. Without this separation the resulting accuracy of the S-parameters will be compromised. To precisely determine a distance you can solve the 'port only' with one additional mode (for a correct port definition this mode will be evanescent) and extract the decay length of this mode from its complex gamma. Then, as a rule of thumb, place the first discontinuity at least three times this decay length. Such rigor is not usually necessary and experience over time will give you a solid understanding of typical spacings for their port geometries and frequencies of analysis.

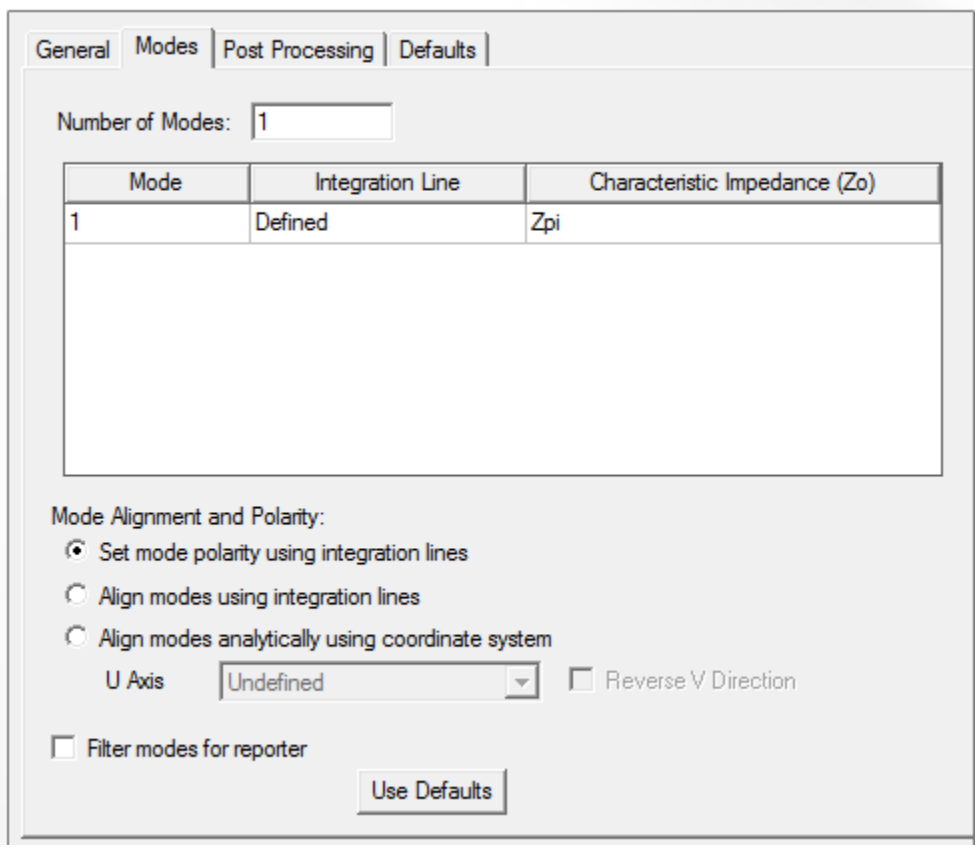
Number of Modes

The electrical size of a port at the highest frequency of interest determines the number of modes to be included in the port definitions. If the guidelines on discontinuity discussed above are followed, the number of modes equals the number of propagating modes at the highest frequency; if they are not followed then, the port as a boundary condition will not be accurate. Regardless of whether they are propagating or not, the solver will solve these modes for calculating the S-parameters.

Wave Port Dialog for Modal Solutions

A **Wave Port** dialog box with its default settings is shown below. On this dialog box you can set the number of modes, their alignment, mode polarity, the choice of characteristic impedance etc.

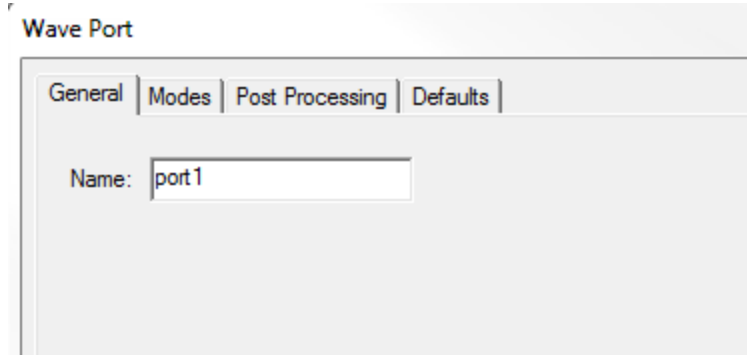
Note: The option **Use Wizards for data input when creating new boundaries** under **Tools** > **Options** > **HFSS Options** is unchecked; so the dialog box appears in the tabbed format. For more information about the appearance of the **Wave Port** dialog box, see **HFSS Options**.



The different tabs and their sub-panels are described in the following subsections.

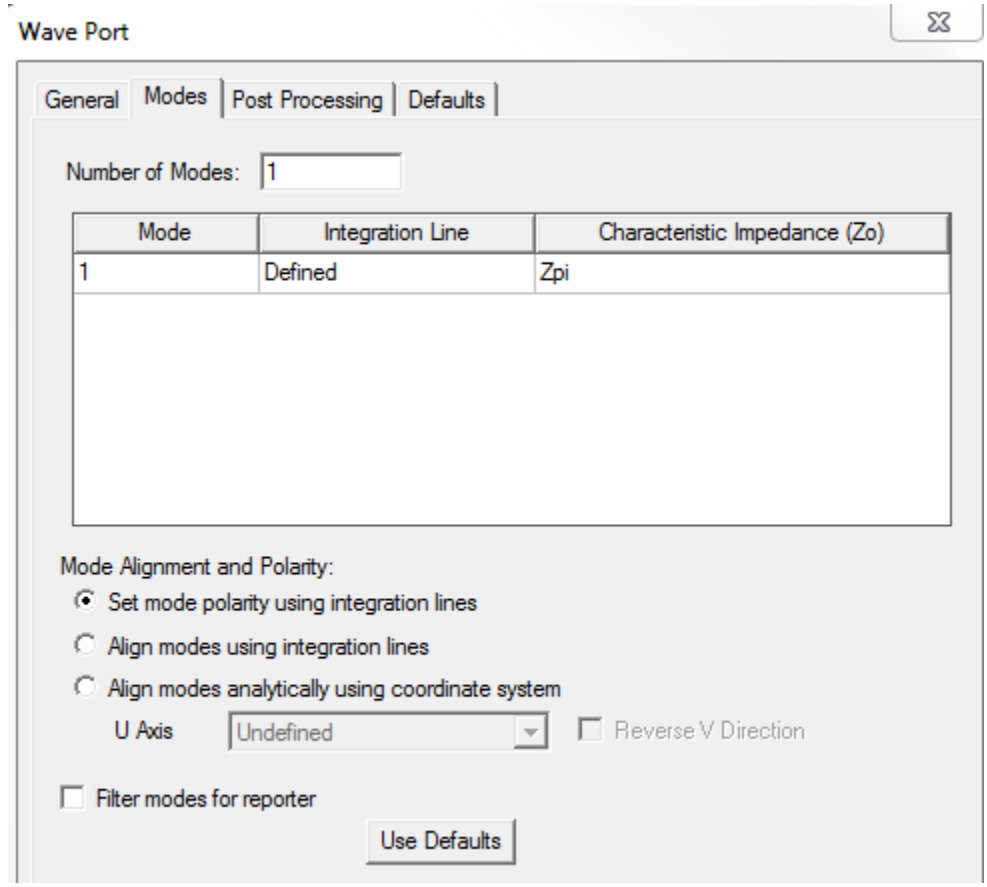
General Tab : Wave Ports

This panel contains a field where you can enter the name of the wave port.



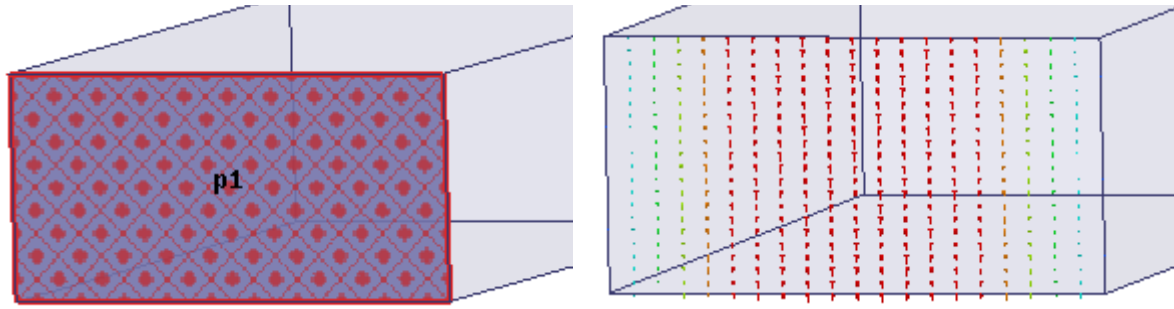
Modes Tab : Wave Ports

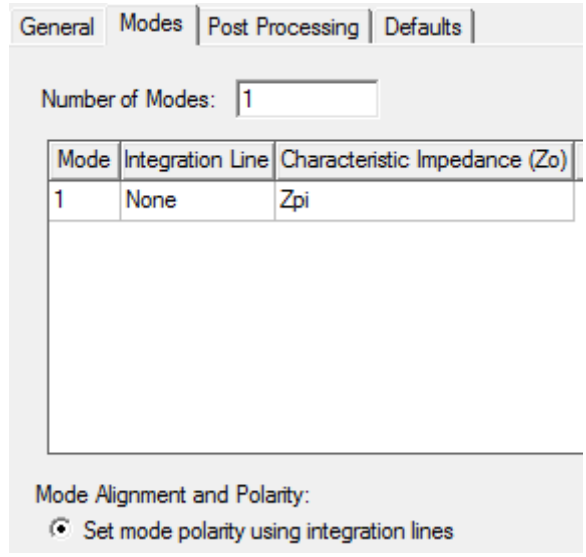
On this panel you can enter the number of modes, set integration lines, and define mode alignment, polarity, and characteristic impedance. Subsequent sections describe all these options and the effect they have on the port field.



Modes with Default Settings

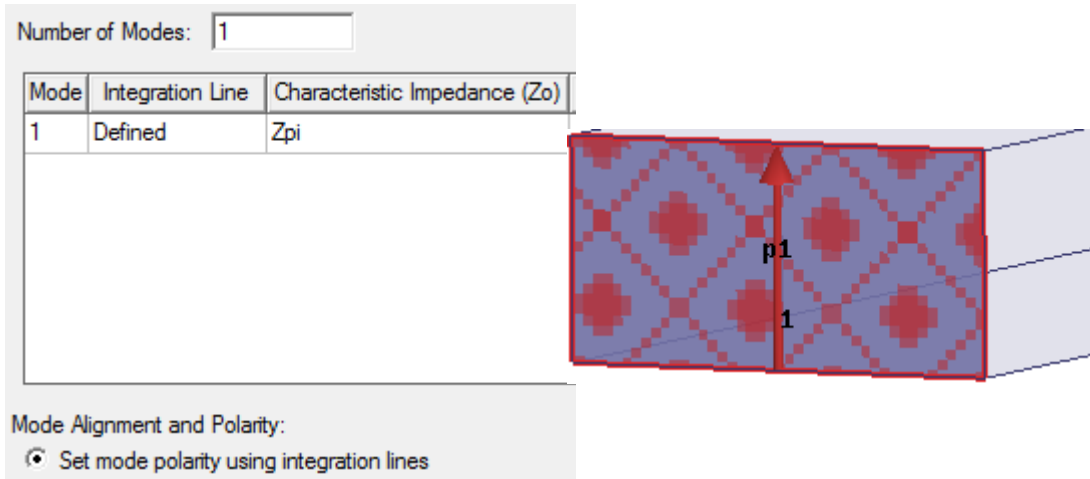
The figure below shows a wave port assigned on the face of a rectangular waveguide. The default settings in the **Modes** tab of the **Wave Port** dialog box are used and the port is named *p1*. No integration line has been defined. From the **Project Tree** if you select the option **Mode1** under **Port Field Display** you can see the electric field patterns for that mode. The direction of mode propagation for the corresponding wave port is shown below. Notice that the intensity is highest in the middle (indicated by the red arrows) and tapers to zero on the sides for the TE_{01} mode for the rectangular waveguide. Since no integration line has been defined, the polarity is arbitrarily up or down.





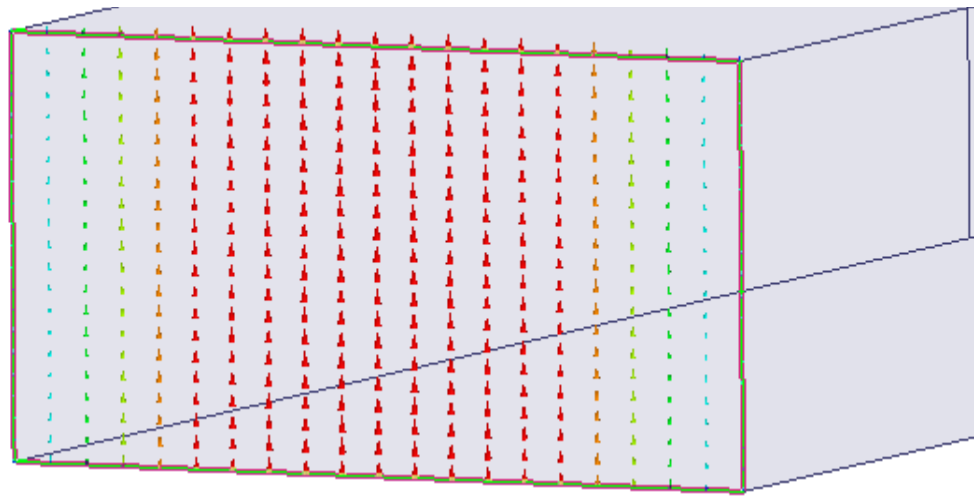
Set Mode Polarity Using Integration Line

By drawing integration lines you can set mode polarity. The settings in the **Modes** tab of the **Wave Port** dialog box for a single mode and the integration line drawn for a WR-90 rectangular waveguide are shown below.

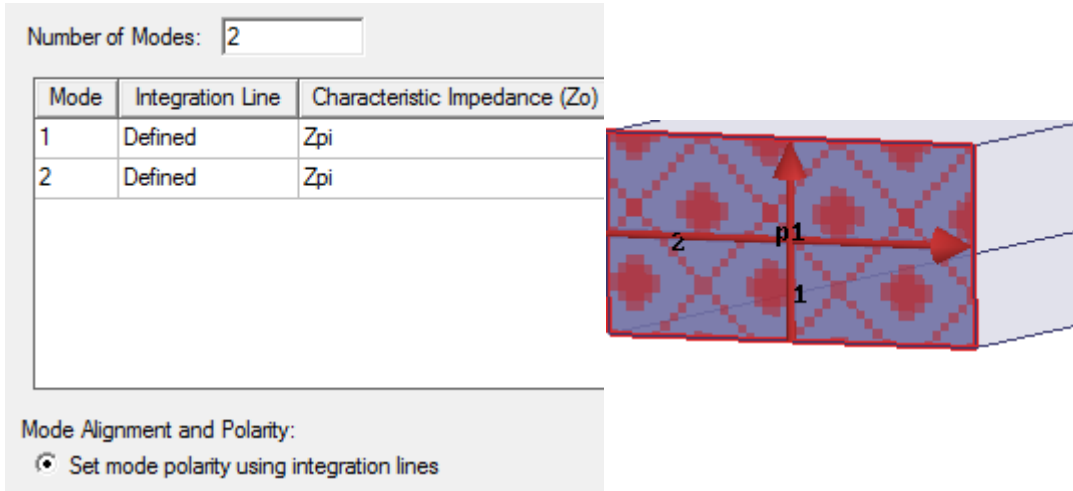


The mode alignment can also be set using the integration lines. In this case the direction is flipped upward as set by the direction of the integration line.

.

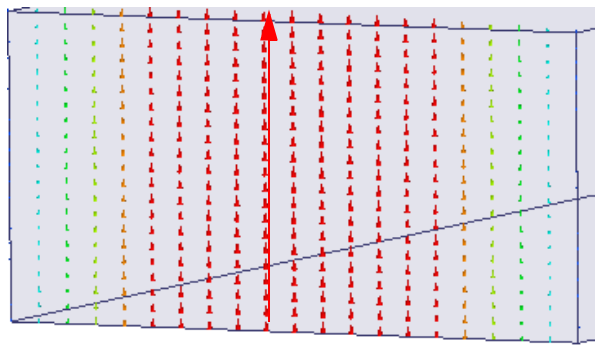


See the settings for another WR-90 waveguide with the integration lines 1 and 2 defined for two modes.

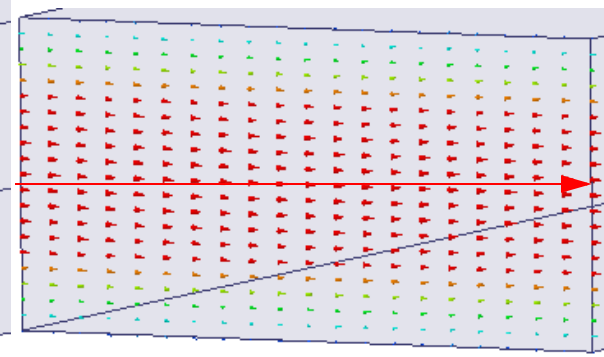


The corresponding port field display are as follows:

Mode 1 Port field display



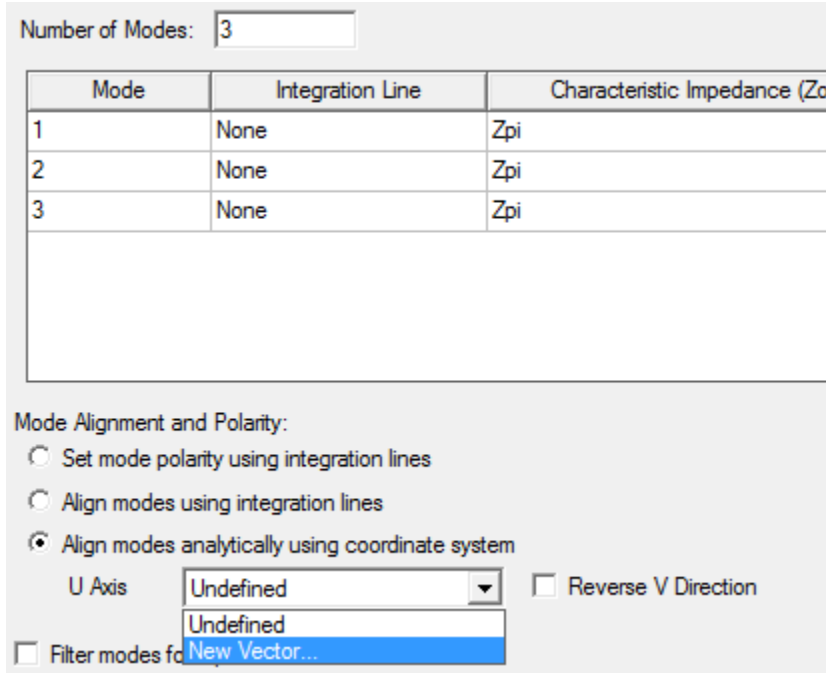
Mode 2 Port field display



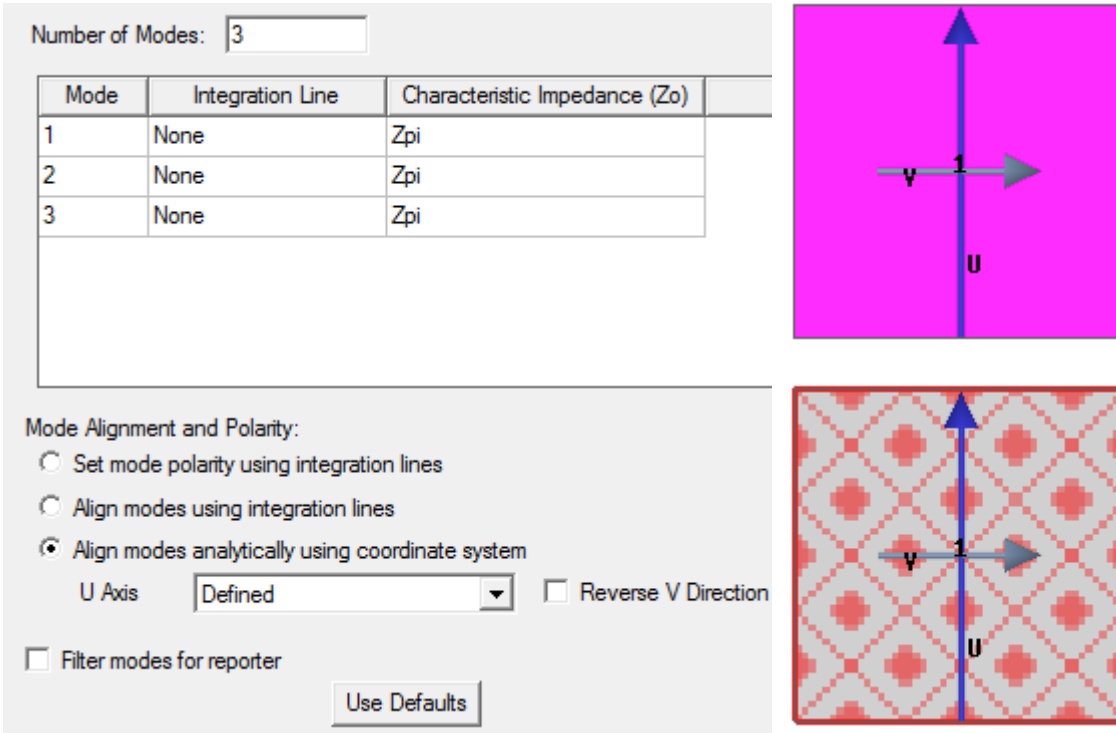
Note: The voltage along integration line 1 is positive for Mode 1 and the voltage along the integration line 2 is positive for Mode 2.

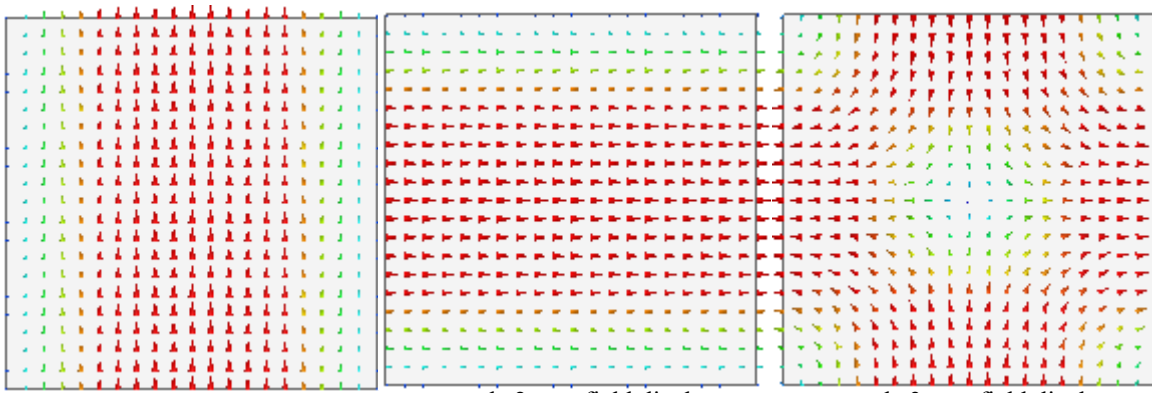
Align Modes Analytically Using Coordinate System

This option is used to align modes for standard waveguides such as square waveguides, rectangular waveguides, coaxial waveguides etc. (For a complete list, see the section Analytic Port Types). On the **Modes** tab of the **Wave Port** dialog box, select the last radio button and **New Vector** from the drop-down menu. Draw the U-V axes to determine the mode alignment. The U axis must lie on the center of the port plane and split the port into equal haves. Otherwise an error message will pop up.



The figures below show the cross section of a square wave guide for which the U-V axes are set and no integration lines defined for 3 modes. The U-axis must lie along the port plane and split the port into two equal halves. After you define the **Solution Setup** and run the simulation, the **Port Field Display** for the 3 modes are also shown in the figures below.



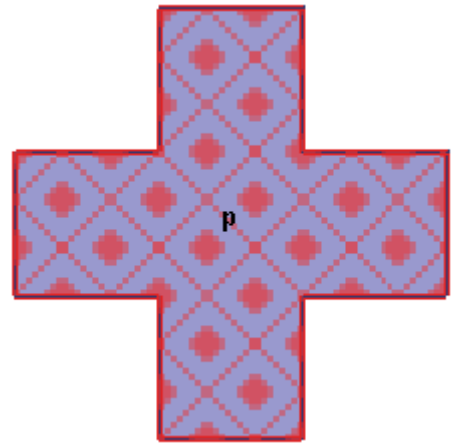


mode 1 port field display mode 2 port field display mode 3 port field display

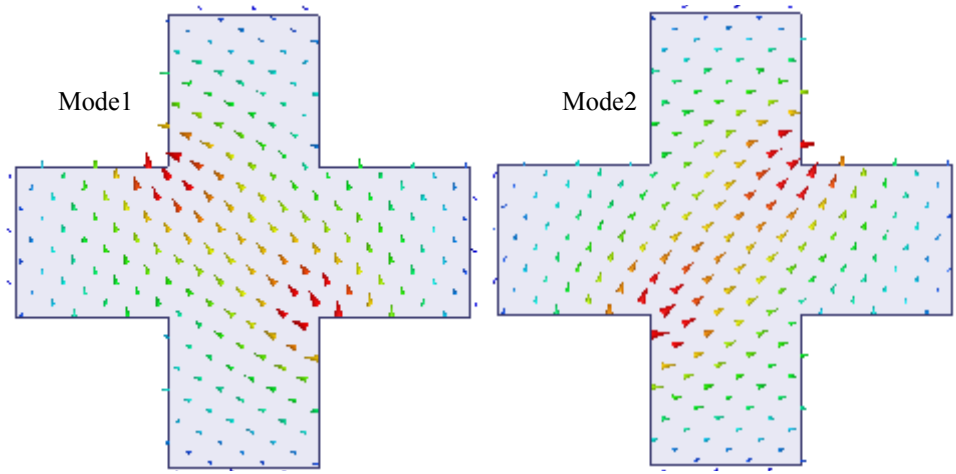
Align Modes Using Integration Lines

This option is for advanced users working with non-standard waveguides which may have degenerate modes. First we will investigate the fields for a waveport assigned on the surface of a symmetric

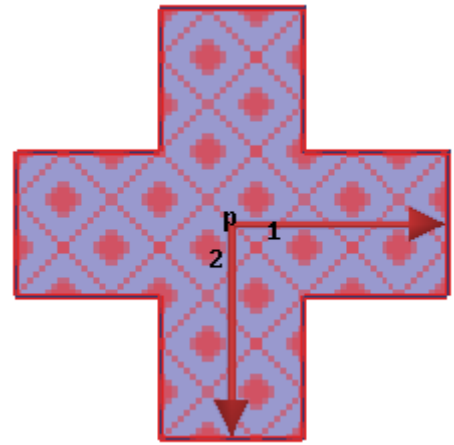
cross-waveguide (shown below) using the default settings on the **Modes** tab of the **Wave Port** dialog box. (**Set mode polarity using integration lines**) for 2 modes. No integration line is defined.

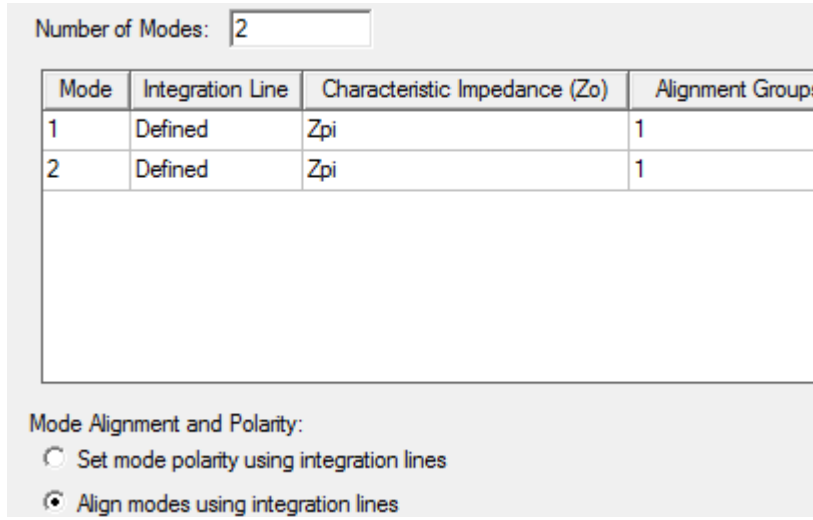


The solver arbitrarily assigns the polarity for the modes 1 and 2.



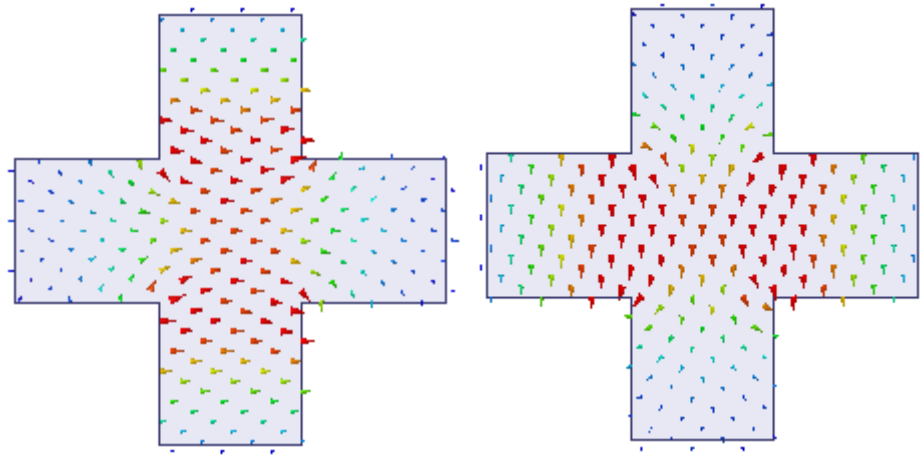
Any combination of degenerate modes results in just another combination of degenerate modes. There is no uniqueness in the orientation of the modes. To get a unique orientation advanced users can select this option **Align modes using integration lines** and define the integration lines appropriately. Prior knowledge of the mode pattern is necessary before you define the Integration Lines for mode alignment. You cannot draw the integration lines randomly. They are extremely important to get the mode alignment that you want. See the integration lines 1 and 2 defined on a port face shown below and the settings in the **Modes** sub-panel.



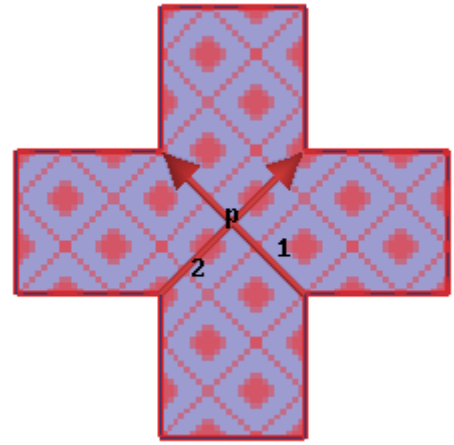


Alignment Groups

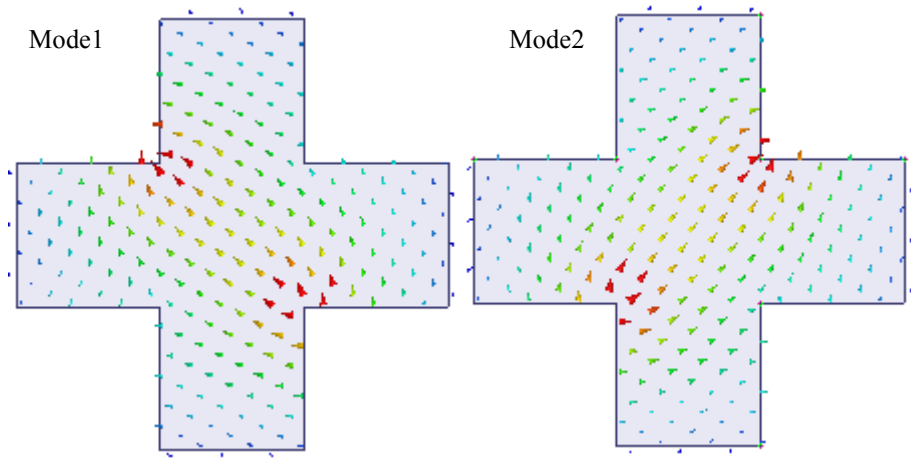
This column appears only when you select the option **Align modes using integration lines**. The modes that have the same **Alignment Group** number are degenerate. To help the solver choose which modes are degenerate you must assign the **Alignment Groups.** When you set the alignment group numbers the solver accumulates the modes of a specific group number and causes each mode within that group to have a positive voltage along its own integration line and zero voltage along all other integration lines. For example, defining integration lines 1 and 2 at right angles to each other for the port face (shown above) will cause the modes to be aligned as illustrated in the following figures.



Suppose you want the modes to be aligned along the integration lines 1 and 2 defined diagonally as shown in the figure below.



According to this definition of the integration lines, the solver aligns the modes within a given group (1 in this case) such that Mode *n* has a positive voltage along integration line *n* and zero voltage along all other integration lines.



Note: For more information, see Mode Alignment.

Characteristic Impedance Column

The **Characteristic Impedance Column (Zo)** lists three options (**Zpi, Zpv**, and **Zvi)** in its drop down menu. The last two options **Zpv** and **Zvi** will appear only if you provide a voltage through defining an integration line for a specific mode such that a unique voltage for that mode can be computed. These quantities **Zpi, Zpv**, and **Zvi** need not be equal to each other. Other than TEM modes, these three quantities will give different results since the voltage is not unique and the result depends on the path used to compute the voltage.

Mode	Integration Line	Characteristic Impedance (Zo)
1	Defined	Zpi
2	None	

Zpi
Zpv
Zvi

In case you do not define an integration line only **Zpi** is shown.

Mode	Integration Line	Characteristic Impedance (Zo)
1	None	Zpi
2	None	Zpi

Note: For more information, see calculate the characteristic impedance.

Electromagnetic power **P** = **E** x **H** (with magnitude of **P** i.e. the Poynting vector as 1 watt) is

injected into the 3D model to calculate the fields, voltages, and currents. The basic equations for characteristic impedance are as follows:

$$Zo = \frac{V}{I} \quad ; \quad P = V \cdot I$$

The expressions for characteristic impedance in terms of power vary as follows:

When using **Zpi:**

$$Zpi = \frac{P}{I^2}$$

When using **Zpv:**

$$Zpv = \frac{V^2}{P}$$

Finally, for **Zvi** the following expression holds.

$$Zvi = \sqrt{Zpi \cdot Zpv}$$

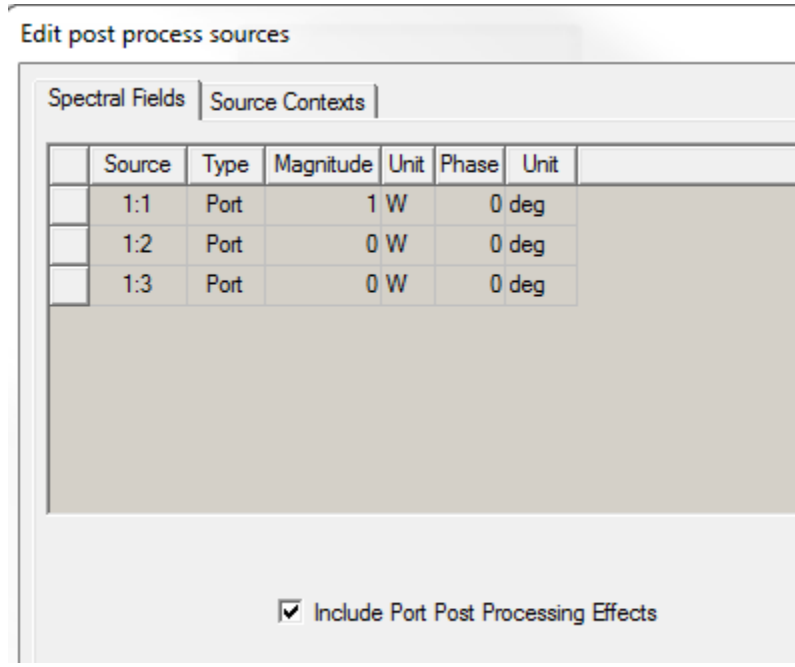
Note For more information about how HFSS defines these values, see Calculating the Z_{PI} Impedance, and Calculating the Z_{PV} Impedance.

Port Post Processing Tab : Modal Solutions

De-embedding and renormalization are the two post processing options available in HFSS. De-embedding changes the reference plane where the port is placed rather than literally defining a new port on a different reference plane. Reference plane is the location where the S-parameters are calculated. De-embedding helps to calculate the S-parameters near or on the plane of a discontinuity or when a long transmission line is attached to the port plane instead of explicitly modeling it in HFSS.

Renormalization changes the load impedance of the port.

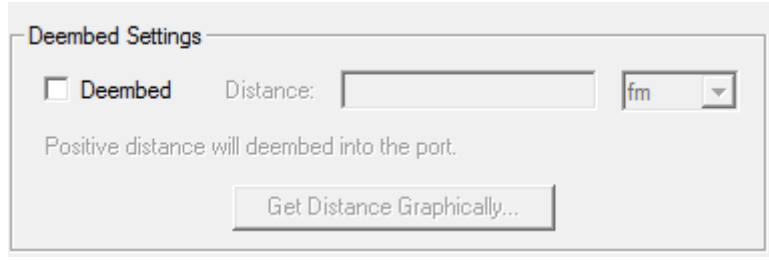
Both renormalization and deembedding affect the network parameters. They will also affect the fields if the **Include Port Post Processing Effects** option is selected in the **Edit Sources** dialog box.



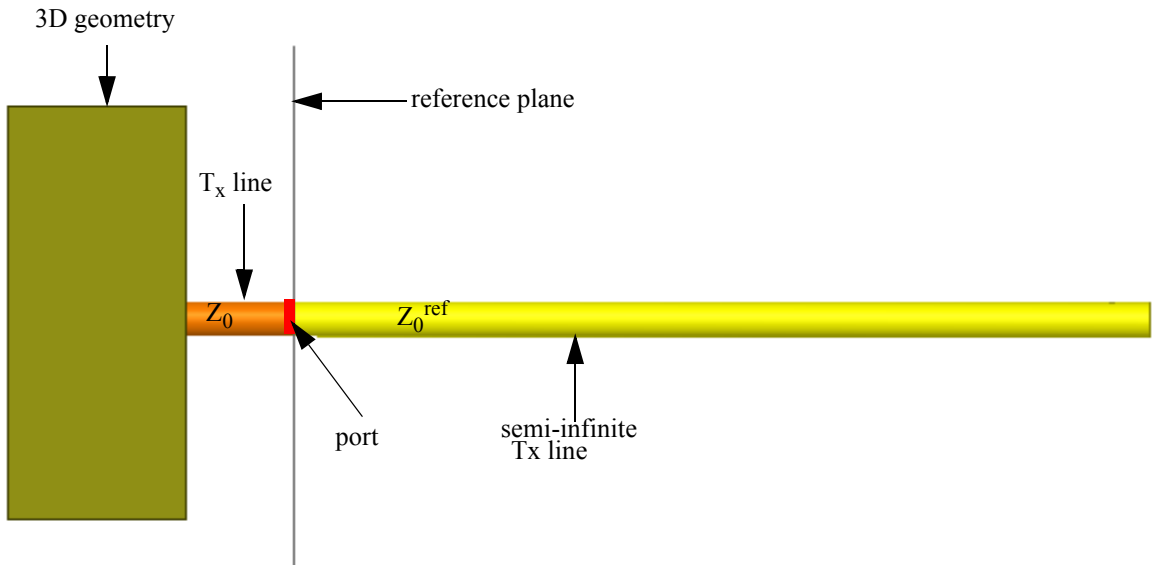
We will now explain the concept of **De-embedding** and **Renormalization**.

Deembed Settings

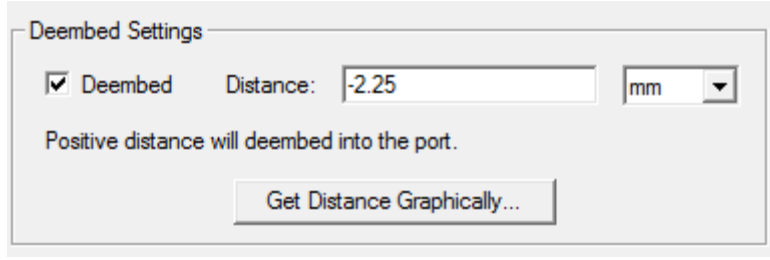
First we will not employ deembedding as shown under **Deembed Settings** of the wave port post processing dialog box.



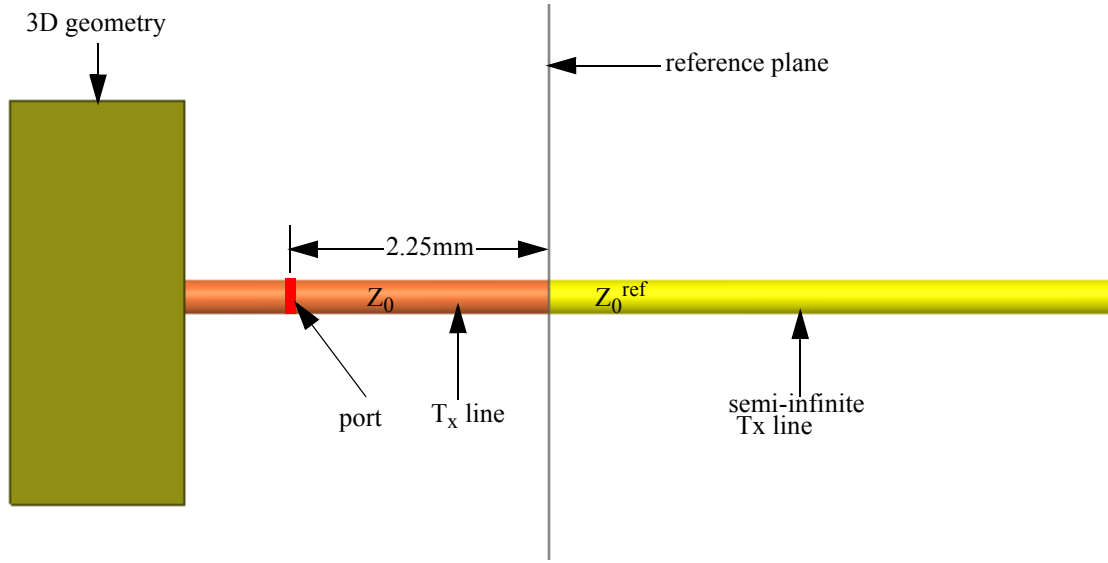
In this situation, by default the reference plane will be along the port surface as shown below.



We will now employ deembedding.



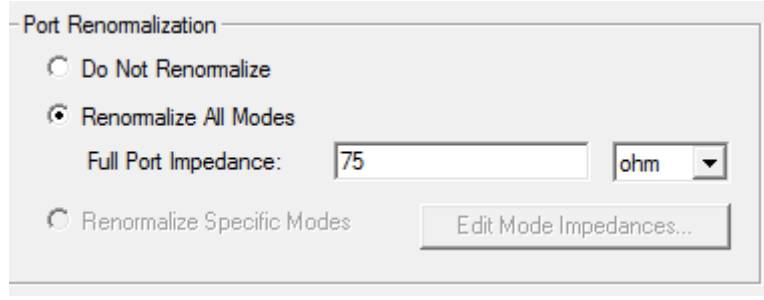
HFSS shifts the reference plane when you deembed.



Note

Renormalization

This option lets you vary the value of Z_0^{ref}. When you do not renormalize Z_0^{ref} is Z_0 . When you renormalize as shown in the figure below, Z_0^{ref} will no longer be Z_0 but it will be assigned the value that you enter in the **Full Port Impedance** field. The settings in the figure below makes Z_0^{ref} = 75 ohm.



Note: You do not need to re-run a simulation in order to renormalize a port. Post-processing reports are automatically updated to reflect the renormalized S-matrix.

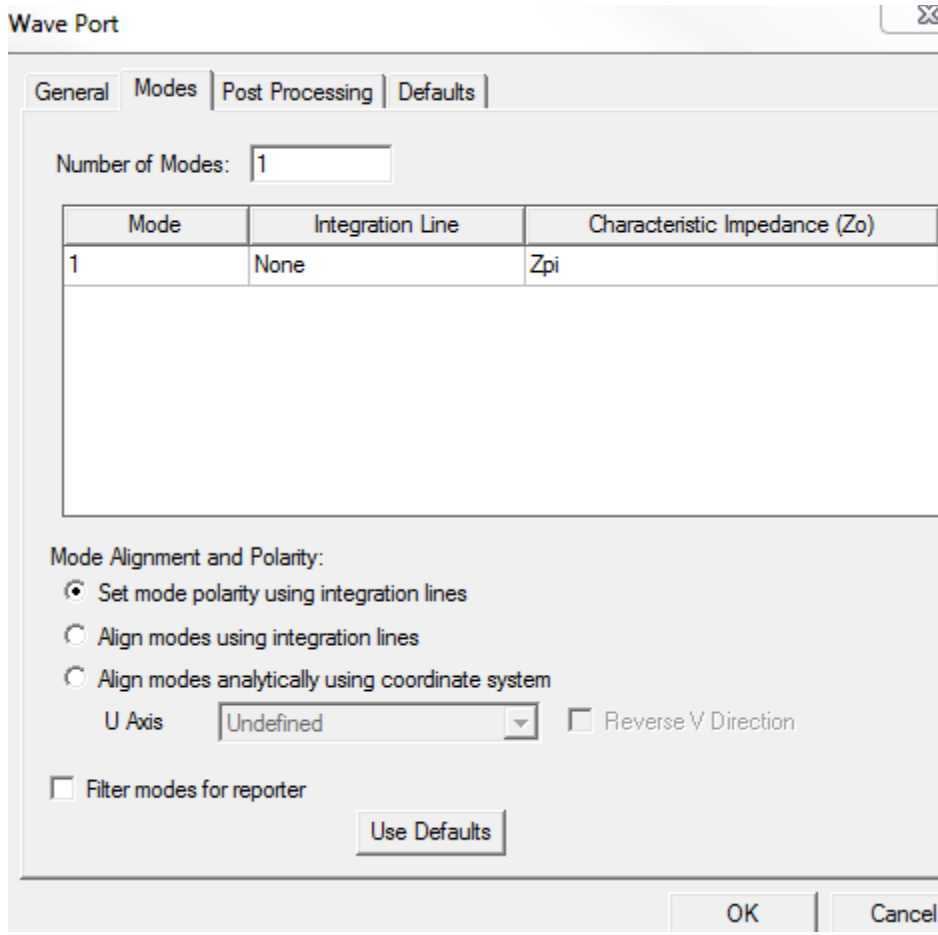
Assign Wave Ports for Modal Solutions

This section outlines the steps for defining excitations for modal solutions.

1. Select coplanar face(s) that you want to excite.
2. Click **HFSS>Excitations>Assign>Wave Port**

   The **Wave Port** dialog menu appears with the **General** tab selected.
3. Enter the name of the waveport and click **Next**.
4. Specify the number of modes for a port.
5. If necessary define the integration lines for each mode, set the characteristic impedance, and specify whether to filter modes for the reporter.

   Note For more information about the **Integration Lines**, see Defining an integration line.



Define Mode Alignment and Polarity

This section discusses the options under Modes alignment and Polarity that appears on the **Wave Port** dialog box. First bring up the **Wave Port** dialog box as follows:

1. Select the face that you want to excite, right click, and select **Assign Excitations**>**Wave Port**.
2. Name the port, and click **Modes** if the dialog appears in tabbed format.

   Notes: If the **Wave Port** dialog box appears in the wizard format, then, click **Next** to access the **Modes** panel.

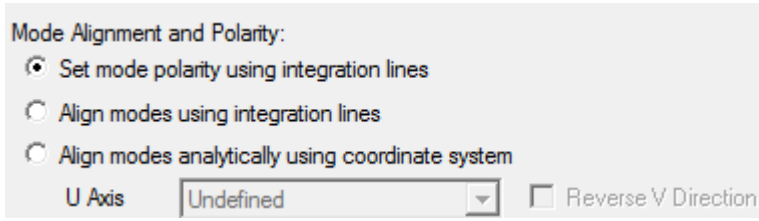
The three options under **Mode Alignment and Polarity** are described below.

Use Set mode polarity using integration lines

This is the default. Use this option as follows.

1. Bring up the **Wave Port** dialog box, access the Modes panel.
2. Enter the number of modes, select the radio button **Set mode polarity using integration lines** to control the polarity.

   Note: HFSS will arbitrarily align the E fields of the modes.

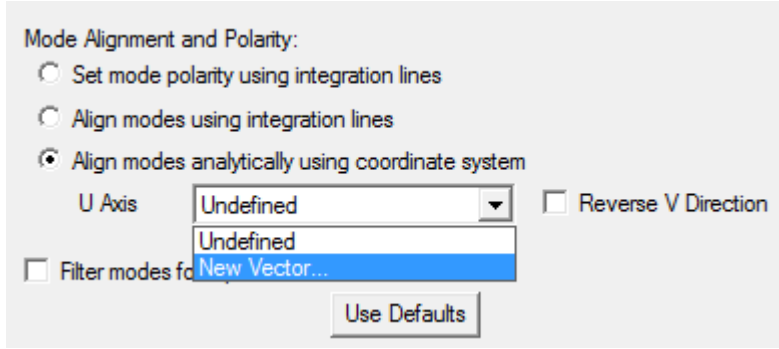


Note: For more information, see Modes with Default Settings and Set Mode Polarity Using Integration Lines**.**

Use Align modes analytically using coordinate system

Use this option if the port corresponds to an analytic port type.

1. Bring up the **Wave Port** dialog box and click the **Modes** tab and select this option.



2. Select **New Vector** from the U Axis drop-down menu.
3. Draw the U axis to split the port symmetrically.

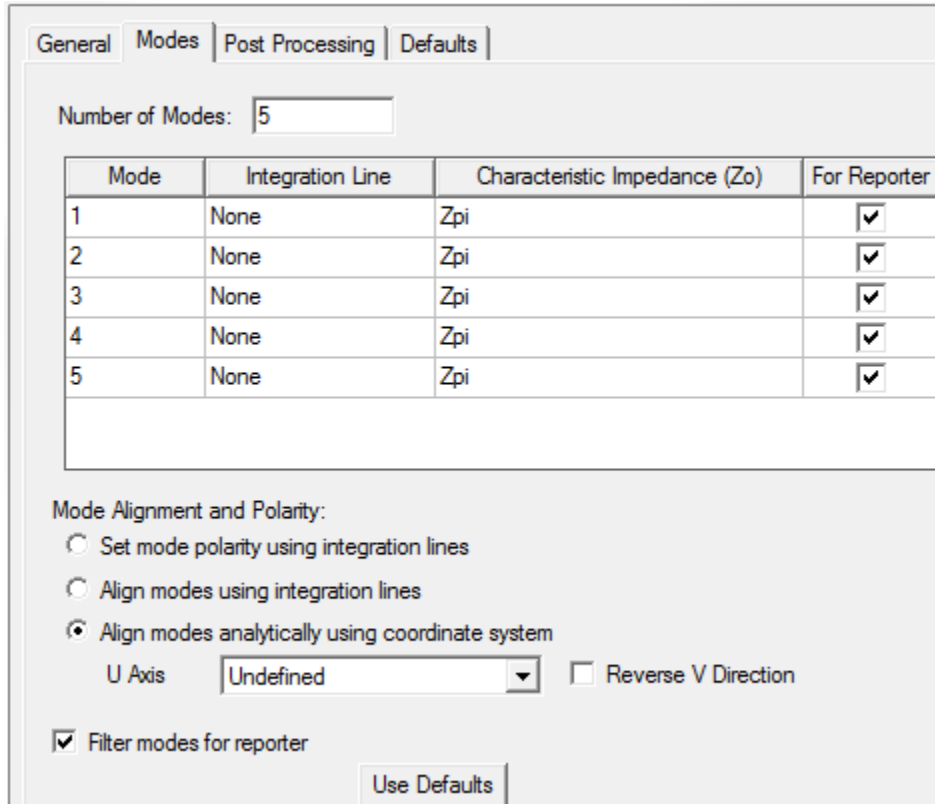
The V direction is computed automatically and can be flipped using the **Reverse V Direction** checkbox. The solver polarizes the fields by aligning them with analytic mode patterns that are generated on the U-V coordinate system.

Note: For legacy projects, when reading in a port, if alignment is requested but only one mode exists, HFSS turns alignment OFF. This prevents an error in validation, and preserves the behavior of having the fields polarized but not aligned.

If desired, you can check the **Filter Modes for Reporter** checkbox.

This adds a new column to the Mode table, which lets you use a checkbox to designate a mode **For Reporter**. For designs with multiple modes, this function will simplify your selections when you create traces for reports.

For more information, see Align Modes Analytically Using Coordinate System.



Use Align modes using integration lines

This option meant for advanced users causes polarization for non-analytic ports. There is no restriction on the port geometry, materials, or integration lines.

1. Bring up the **Wave Port** dialog box and click the **Modes** tab.
2. Specify more than one mode.
3. Select **Align modes using integration lines**.

A column **Alignment Groups** is added in the Modes table.

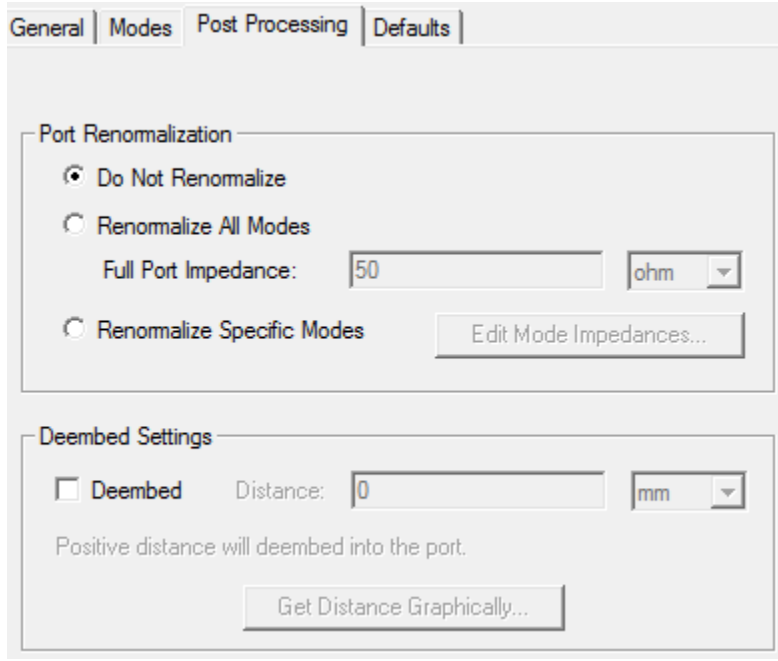
4. Select **Alignment Groups** from the corresponding drop down menu.
5. Define the **Integration Lines** on the port surface as needed.

For more information about this option and the effect of **Integration Lines**, see Align Modes Using Integration Lines.

Post Processing: Wave Ports

The **Post Processing** sub-panel has options to renormalize the port to 50 ohms (or any other desired value) or just accept the default option **Do Not Renormalize** and also the **Distance** field to de-embed a port into or out of the model as needed.

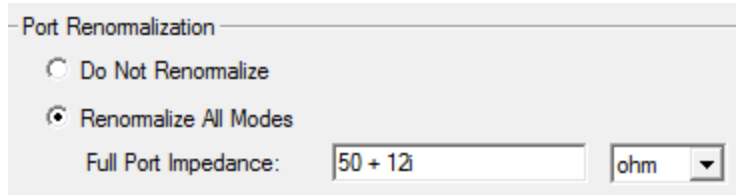
A **Wave Port: Post Processing** panel is shown below.



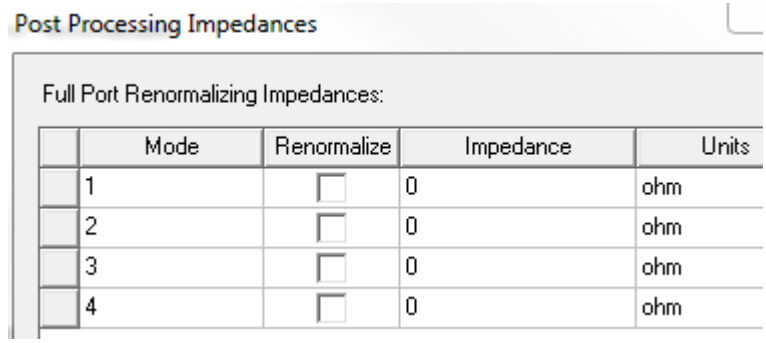
The Port Renormalization choices include:

- **Do Not Renormalize:** This is the default.
- **Renormalize All Modes**: Enables the **Full Port Impedance** text box. The default impedance for re-normalization of each mode is 50 ohms.

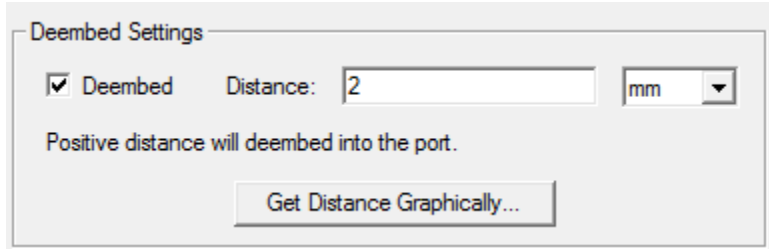
  Complex impedance should be entered in the form: *<re> + <im>j*. See figure below.



- For multiple modes, the **Renormalize Specific Modes** is enabled, causing the **Edit Mode Impedances** button to be active. Click this button to renormalize the impedance of desired modes.



6. Enter the **Deembed** distance in the field or draw a line on the model to represent the distance and get it graphically. You can also assign a variable in the **Distance** field.



Important: The orientation of the **Deembed** arrows in the modeler is determined by the modeler direction of the normal of the face where the port is defined. From the standpoint of the S-parameters what is correct is the sign of the deembed distance.

Note: A positive distance value will deembed the port into the structure. A negative distance value will deembed the port out of the structure.

Note: Make sure you are not creating a non physical situation when you deembed cutoff modes. For example, when you enter a positive distance to deembed a port with a cutoff mode into the model the resulting field values for that mode can be very large.

Related Topics

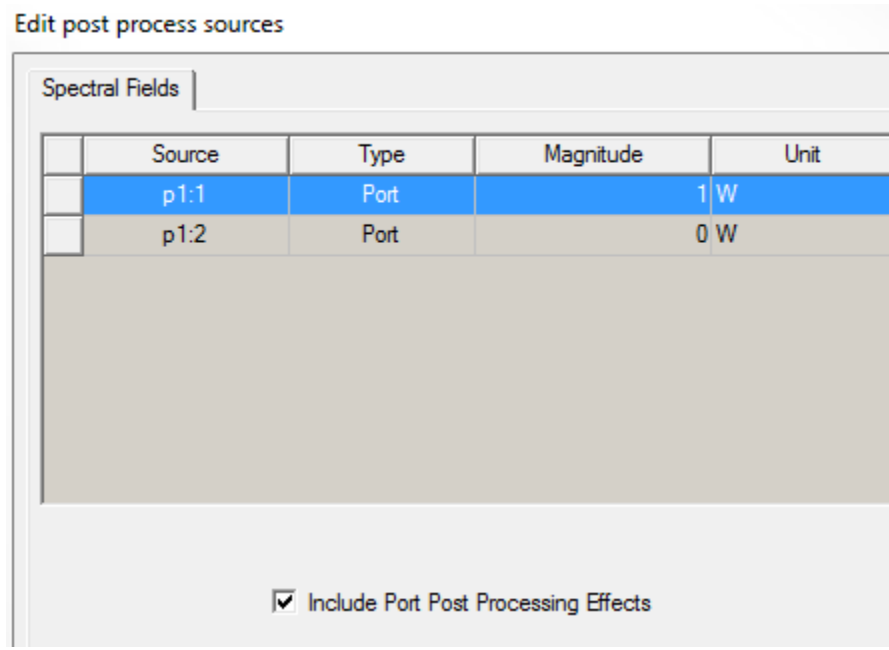
Port Post Processing Tab : Modal Solutions

Applications for Deembedding

Post Processing and Edit Sources

The settings in the **Post Processing** dialog box affect S-Parameters. However, on the **Edit post process sources** dialog box if you check the option **Include Port Post Processing Effects**, then the post processing operations i.e. renormalization and/or deembedding will also affect the Spectral Fields.

Note: For more information, see Scaling a Source Magnitude and Phase.



Related Topics

Port Post Processing Tab : Modal Solutions

Applications for Deembedding

Technical Notes: Deembedding

Wave Ports for Terminal Solutions

HFSS treats a wave port as a semi-infinitely long waveguide or transmission line. For terminal solutions, we deal with transmission lines only. Terminals are the ends of signal traces that intersect the plane of a port.

To fit circuit theory HFSS has options for assigning terminals and expressing the relationship between currents and voltages. Terminals are especially useful when you are dealing with circuit simulators since the S-parameters obtained in terms of currents and voltages are in a format that is compatible with most circuit simulators.

Terminals and Modes

Example of Terminals

Reference Conductors

Related Topics

Assign Wave Ports for Terminal Solutions

Terminals and Modes

A terminal is assigned on a conductor in contact with the port. A port with *n+1* distinct conductors that do not touch each other, will support *n* terminals where the remaining conductor is the local reference ground. The solver generates an error if the number of terminals exceeds *n* conductors.

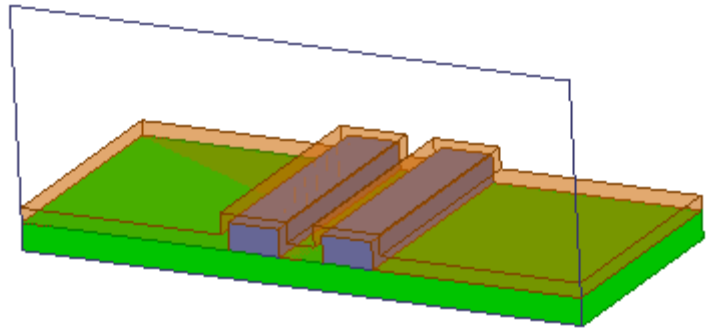
The solver uses as many modes as terminals. These modes are not eigen modes and sometimes referred to as terminal modes. The solver computes terminal modes where all the terminal modes are a superposition of eigen modes such that the S-parameters computed by the solver constitute the corresponding terminal S-Matrix instead of the eigen S-Matrix.

For terminal projects we will deal only with transmission lines and assume that the currents and voltages correspond to quasi-TEM modes. The number of terminal modes is dictated by the number of conductors on the port in the project. Proper port definitions should ensure that no higher order propagating modes exist in the transmission lines.

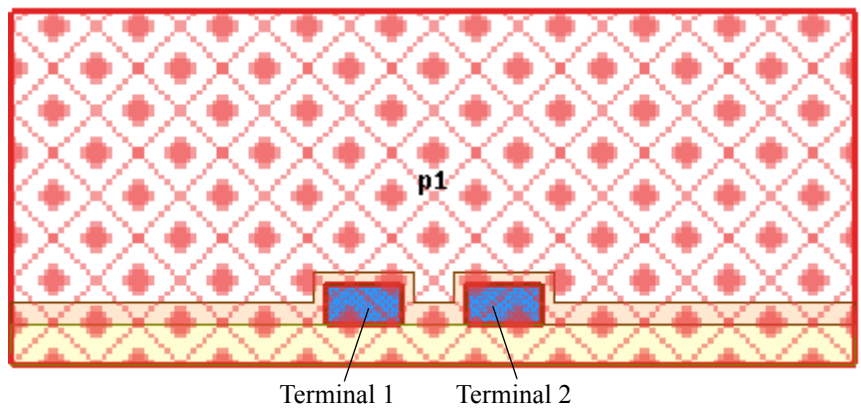
Example of Terminals

A terminal is essentially an insertion point for voltage and current on a multitrace transmission line.

An HFSS model of a coupled microstrip transmission line is shown below.

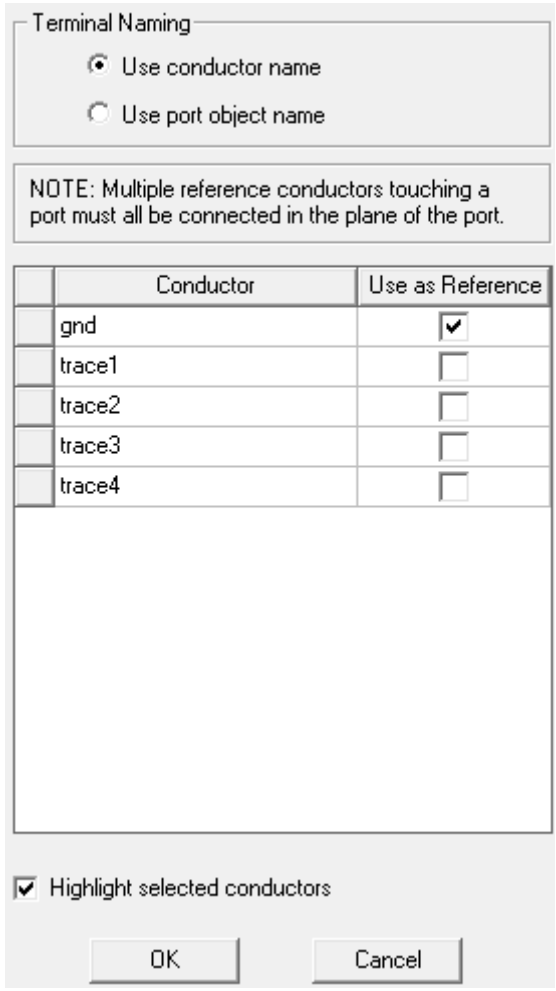


The terminals are defined at the ends of the signal traces where they intersect with the surface of the port as shown below.



Reference Conductors

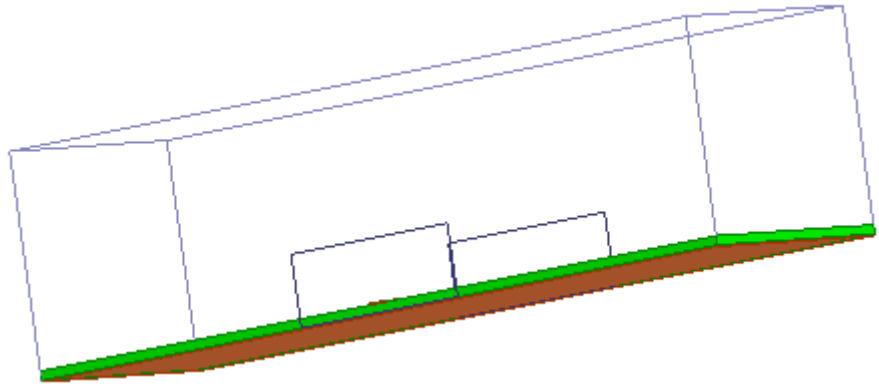
The **Reference Conductors** dialog box contains a table that lists all the conducting objects used in a model. In this dialog box you will choose the conductor(s) that will act as reference. You can name terminals based on the name of their conductors or their associated ports.



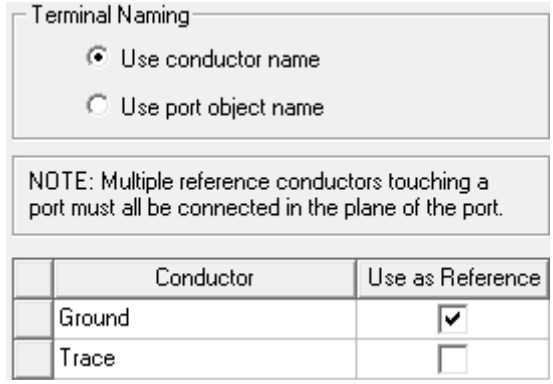
We will depict a couple of scenarios that you may encounter while assigning terminals to conductors in your model. For both scenarios we will consider the same microstrip line with modifications to the model.

Scenario 1: Explicit Reference Conductor

The figure below shows a microstrip line enclosed by an air box. The model has an explicit ground plane that is orange in color.



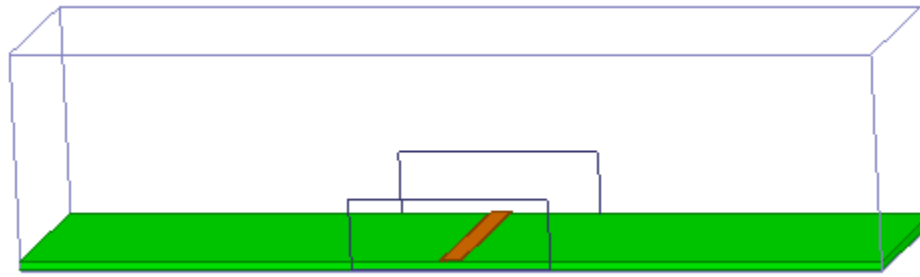
The ground is an explicit object and so it will appear as a conductor in the **Reference Conductors for Terminals** dialog box when assigning terminals to the trace.



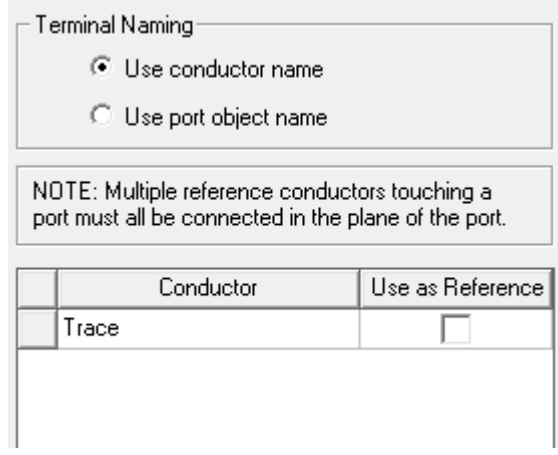
Select **Ground** as reference so that the terminal appears only on the *Trace* and not on the *Ground*.

Scenario 2: Implicit Reference Conductor

The figure below illustrates the same microstrip line enclosed by an air box without the radiation boundary. This model does not have the explicit ground plane.



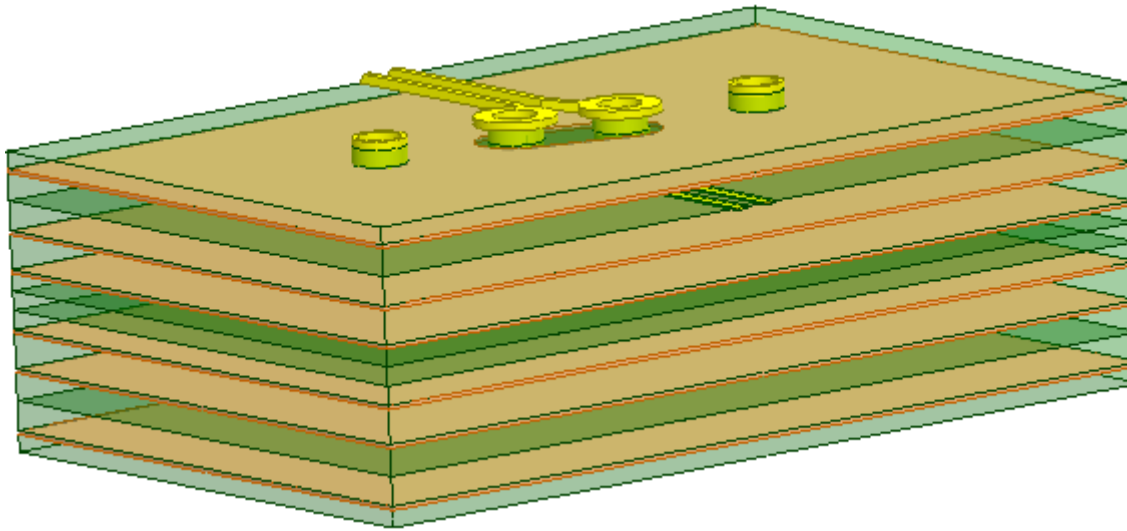
The solver treats the boundary of the air box as PEC because no other explicit boundary condition is applied to it. The air box acts as a reference conductor but will be treated as an implicit conductor that does not appear in the dialog box. In this case do not define the reference conductor as it is already assumed by the solver.



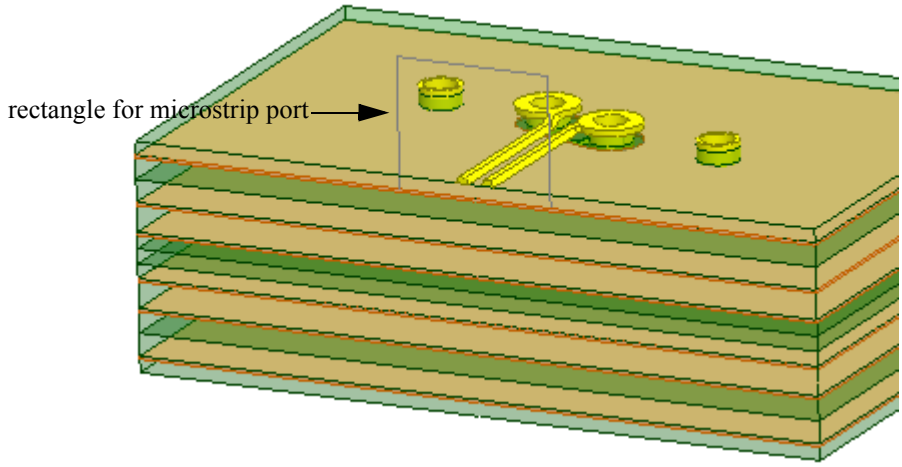
Assign Wave Ports for Terminal Solutions

A terminal is a geometry intersection of a conducting object with a port face. The intersection can be a face or an edge in a conducting geometry. The edge can even extend outside the port and though it can be defined as a terminal in the user interface, only the portion of the edge that overlaps with the port is used to define the terminal edge for the solver.

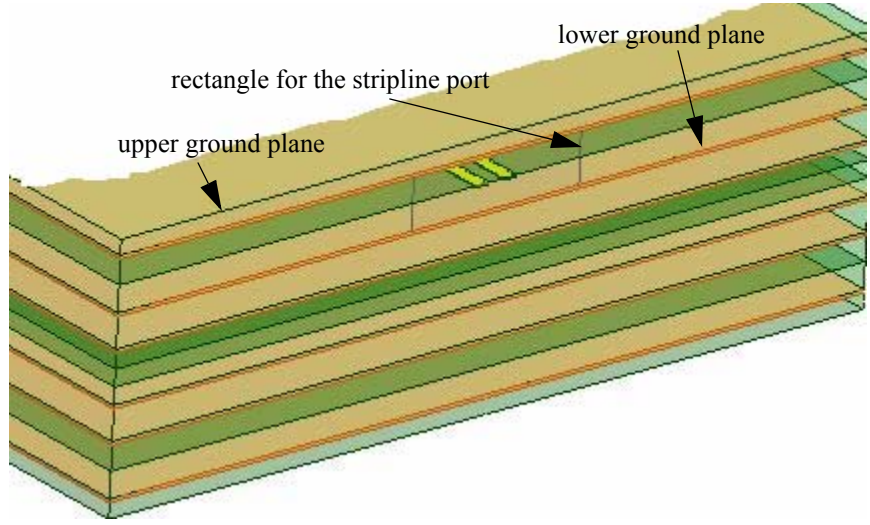
In this section we will assign wave ports on a differential pair via model shown below.



We will assign a microstrip port and a stripline port. The guidelines for defining port size introduced in the Microstrip Waveguide section are also applicable for the differential pair model. The microstrip port size is defined as (10w+s + 2w) x 8h where w = microstrip trace width, s = separation between the traces, and h = height of the substrate.

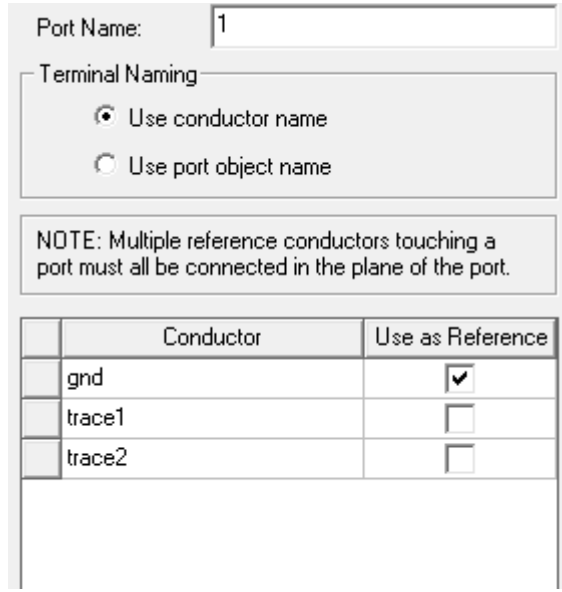


The stripline port is defined as (10b + c + b) x d where b = stripline trace width, c = separation between the traces and d = the distance between the upper and the lower ground plane.



Perform the following steps for terminal assignment.

1. Right-click the rectangle for the microstrip port and select **Assign Excitations**>**Wave Port.** The **Reference Conductors** dialog box appears.



2. Select the **gnd** conductor as reference and click **OK**.

   Note: Number of terminals = Number of conductors - 1 = 3-1 = 2.

   The port gets assigned and the terminals appear automatically on the microstrip.

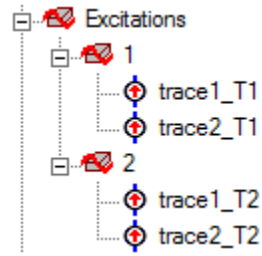
3. Similarly assign a waveport on the stripline rectangle.

   The port gets assigned and the terminals appear automatically on the stripline.

   Note: In rare cases when automatic terminal assignment fails, you can manually assign a terminal by selecting the face of the conductor.

   Note: Identification of conductors depends on a threshold conductivity value. The threshold is based on the material assignment or the boundary assignment if a conducting boundary is assigned to the object.

   Note: The project manager window gets populated as shown in the figure below.

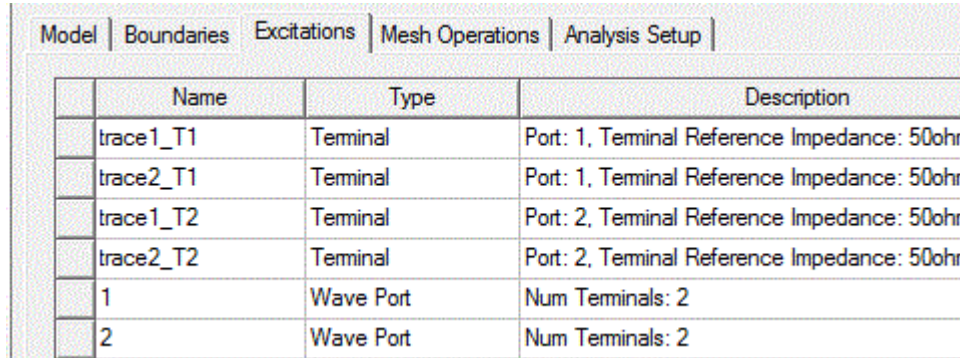


Note: Terminal naming conventions are based by default on the first geometry in the assignment selection for the terminal. You can change the names from the Set Default Boundary/Excitation Base Name dialog box. For auto assign terminals, you can also specify whether naming uses the conductor or the port object name.

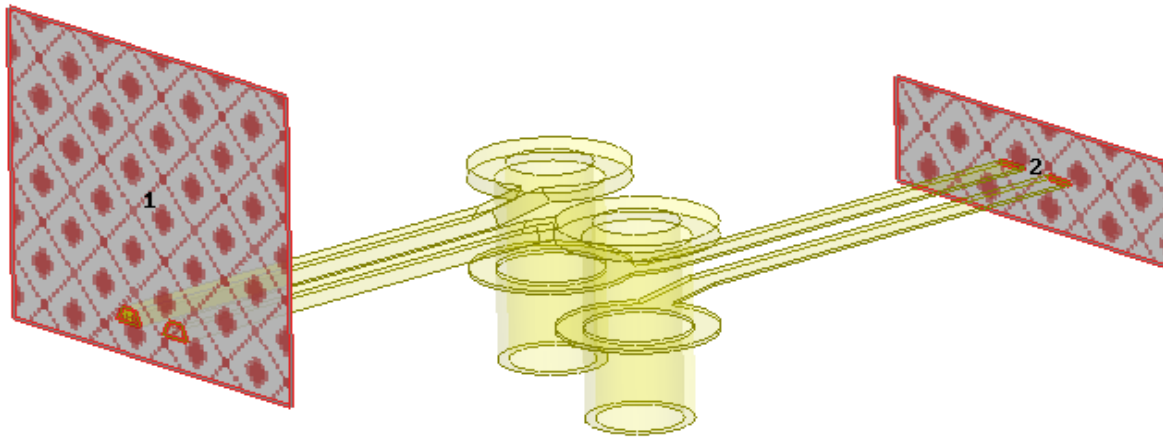
Multi-Select Ports and Terminals

To view all of the assigned excitations at the same time, perform the following steps:

1. Double-click **Excitations** from the **Project Manager** window.
   This brings up the **Design List** dialog box.



2. Press **Ctrl** and select the excitations that you want to see in the model.
   The selected excitations get highlighted in the model.
3. Click **Done** to close the dialog box.
   For example, the figure below shows all the excitations on the differential pair.



Associating Terminals and Ports

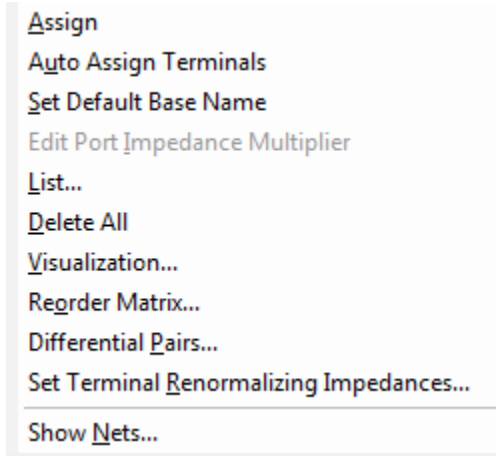
HFSS can automatically associate terminals with the ports whether you assign the ports first and then define the terminals or vice versa. You can define terminals from the **Excitations** level (in the **Project tree**) when the ports already exist. Regardless of the order in which they are assigned, a terminal is associated with the port containing the signal and reference conductors that define the terminal.

Note: At either the port level or for all excitations, you can set the renormalizing impedance for all terminals.

Check the Auto-assign terminals on ports option on the **Tools>Options>HFSS Options** dialog on the **General** tab or use the Auto Assign Terminals command.

Auto Assign Terminals

Right-click **Excitations** from the project tree to bring up the shortcut menu that contains the **Auto Assign Terminals** option.



This option assigns terminals automatically. When you change the **Solution Type** of a model from **Driven Modal** to **Driven Terminal**, you can use this option to define the terminals for the conductors on the existing ports.

In exceptional cases with complicated arrangement of conductors or geometry with slight coordinate misalignments, auto assign may create either too few or too many terminals on a port. In such cases, you can manually assign the terminals.

Related Topics

Active and Passive Excitation in HFSS Transient

Show Nets for DC Continuity for 3D Conductors

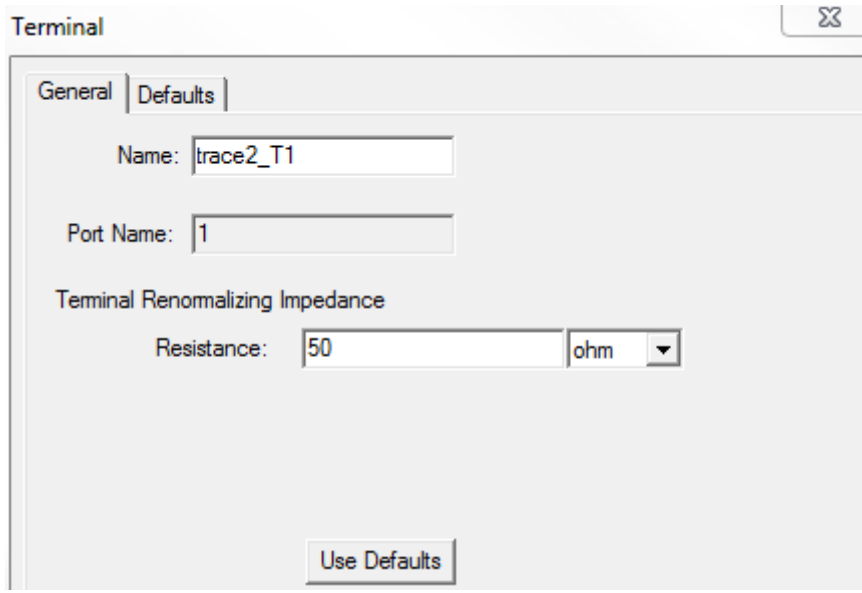
Post Processing Operations

As a post processing operation you can renormalize the impedance of the individual terminals that you assigned. This is done as follows.

1. Double-click the terminal from the **Project Tree**.

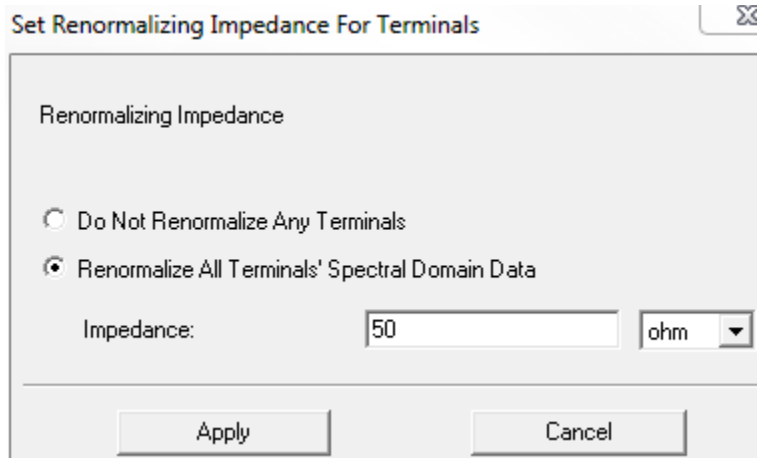
   The **Terminal** dialog box appears.

2. Enter the value in the **Resistance** and set the units from the drop-down menu and click **OK**.



Set Renormalizing Impedance for Terminals

The setup panel for each terminal includes its post processing renormalizing impedance. You can set this value either for all excitations, or for a specific port. If a design includes at least one wave port, the setup panel also includes the radio buttons **Do Not Renormalize Any Terminals**, or **Renormalize All Terminals' Spectral Domain Data**.



To set the renormalizing impedance for **all excitations**:

1. Right click **Excitations** in the Project tree and select **Set Terminal Renormalizing Impedances**.

   The **Set Renormalizing Impedance for Terminals** dialog box appears.
2. In the field for **Impedance**, set the value, and select the units from the pull down.

   This value can be a variable. This variable can be dependent on the frequency, which allows use of a dataset for frequency dependent impedance.
3. Click the **Apply** button to close the dialog and apply the change.

To Set the Reference for All terminals on a Port:

1. Right click the port icon in the Project tree and click **Set Terminal Renormalizing Impedances**.

   The **Set Renormalizing Impedance for Terminals** dialog appears. It differs from the related command for all excitations by specifying that the **Renormalizing Impedance** is for terminals on the selected port.
2. In the field for **Impedance**, set the value, and select the units from the pull down.

   This value can be a variable. This variable can be dependent on the frequency, which allows use of a dataset for frequency dependent impedance.
3. Click the **Apply** button to assign the impedance value.

   Note: For more information, see Scaling a Source Magnitude and Phase.

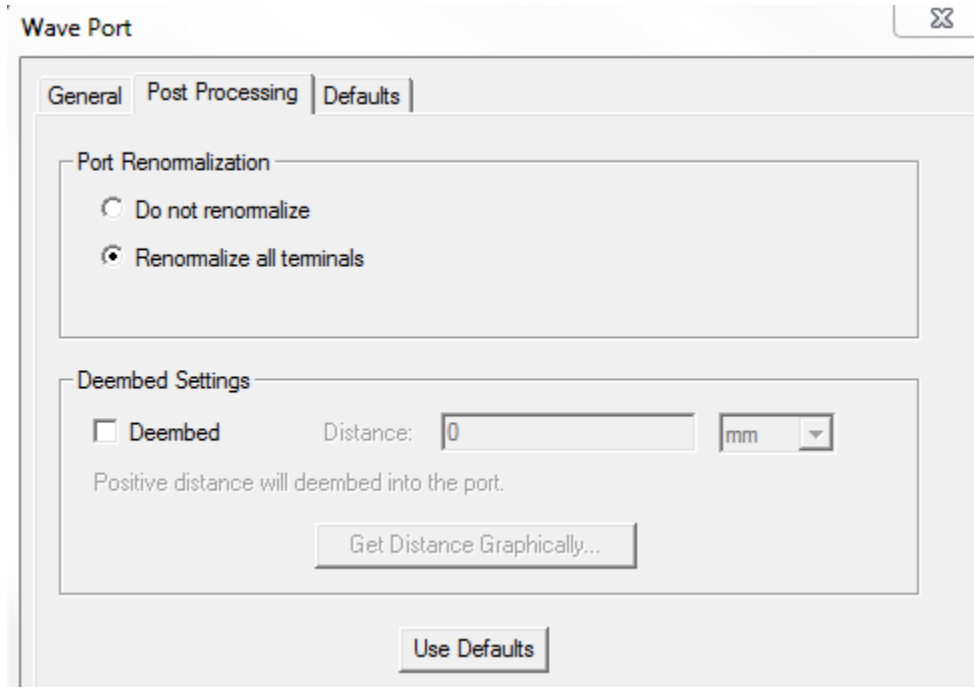
You can also set the Terminal Reference Impedance on a port by selecting the port and editing the value in the **Properties** dialog.

In designs with at least one wave port, where you want to view un-renormalized Zo impedance and/or the corresponding S parameters in either the Matrix data or in a report, you can select the **Do Not Renormalize Any Terminals** radio button.

Renormalize Terminals from the Port Level

1. Double click the wave port in the **Project** tree to display the **Properties** window for the wave port and select the **Post Processing** tab.

.



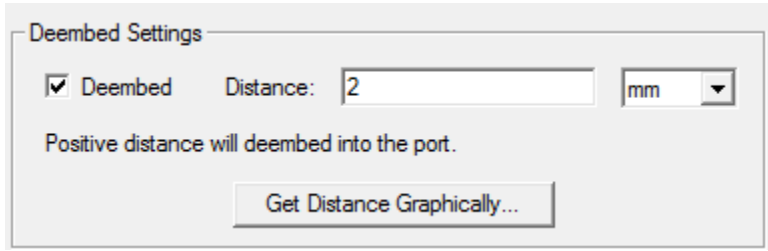
Values here affect S-Parameters only. Port Renormalization choices include:

- **Do Not Renormalize** (the default). Selecting this disables the Impedance fields for the port and terminals.
- **Renormalize All Terminals**. The default impedance for re-normalization of each port is 50 ohms. To specify a different impedance, you must open the Properties for the terminal and specify a value in the **Terminal Renormalizing Impedance** field.

  If you want to enter a complex impedance, enter it in the following form: *<re> + <im>j*
- If there are multiple modes, **Renormalizing Impedance for Specific Terminals** is enabled. Click this to enable the **Edit Terminal Impedances** button. This opens an editable table with the impedances for each terminal.

2. Enter the **Deembed** distance in the **Distance** field or draw a line on the model to represent the distance and get it graphically. You can also assign a variable in the **Distance** field

.



Important: After you enter the value, a blue arrow depicts the deembedding distance in the graphics window when the port is selected. When a waveport has been defined on a sheet, the inward or the outward normal will dictate the direction of the arrows. The orientation of the deembedding arrows in the modeler is determined by the direction of the normal of the face where the port is defined upon. From the standpoint of the S-parameters what is correct is the sign of the deembed distance.

Note: A positive distance value will de-embed the port into the model. A negative distance value will de-embed the port out of the model.

Note: Make sure you are not creating a non physical situation when you deembed cutoff modes. For example, when you enter a positive distance to deembed a port with a cutoff mode into the model the resulting field values for that mode can be very large.

3. For HFSS Transient solutions, you can designate a port as Active or Passive.

Related Topics

Set Renormalizing Impedance for Terminals

Technical Notes: Port Solution Theory

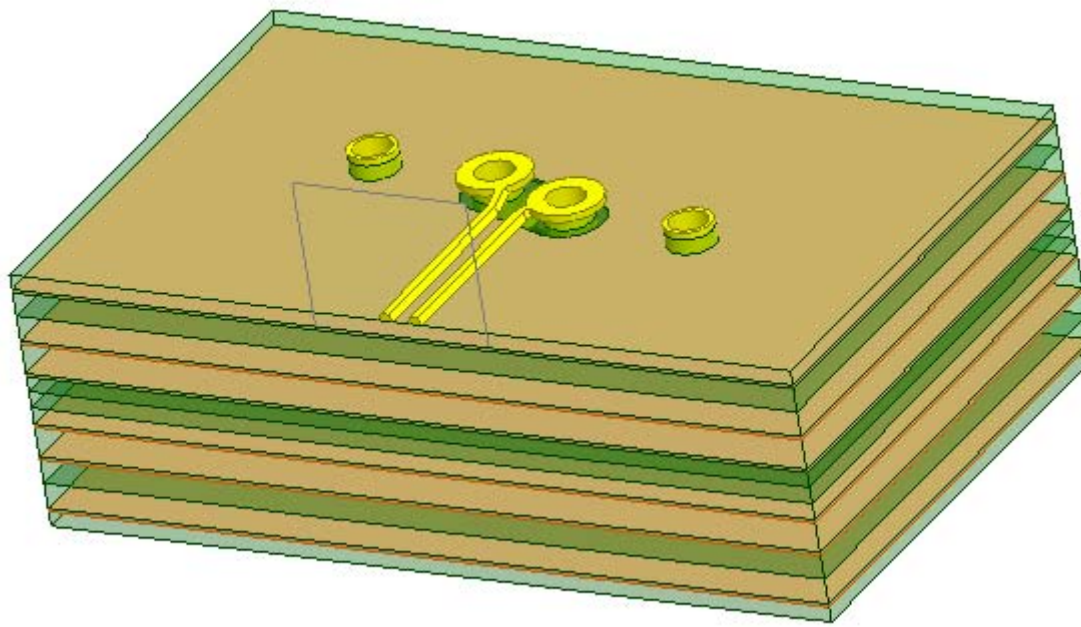
Technical Notes: Deembedding

Set Differential Pairs

A differential pair represents two circuits, one positive and one negative routed close together so they will pick up nearly the same amount of noise. The two signals are subtracted from each other by a receiver, yielding a near "noise-free" version of the signal.

You can define one or more differential pairs from terminal excitations assigned on existing wave ports. Differential pairs can span ports, use lumped ports, and be enabled or disabled. To allow automated calculation of differential S-parameters from lumped ports, you can select terminals from two arbitrary ports, whether wave ports or lumped ports, for use in a differential pair.

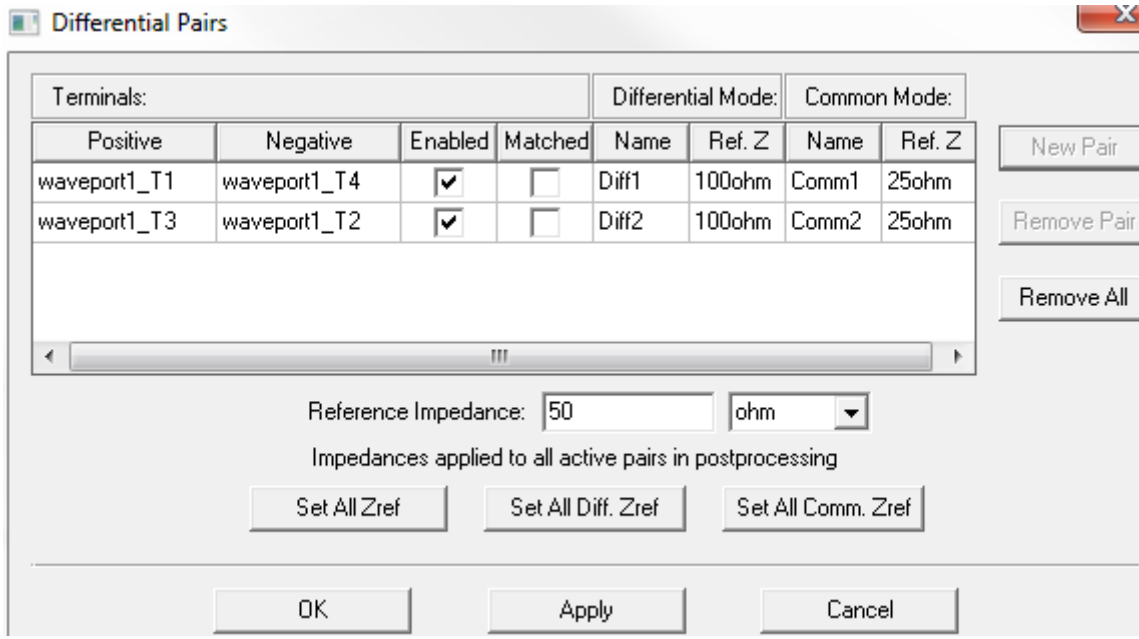
Because differential pairs can span ports or occur within a port, the **Differential Pairs** command is accessible at corresponding levels in the Project tree via the right click menu both at the **Excitations** level, and at the port name level. If a differential pair involves terminals from two different ports, the **Differential Pairs** command for those ports can only be accessed at the **Excitations** level. If an individual wave port has multiple terminals defined, the **Differential Pairs** command is enabled when you select that port and right click to display the shortcut menu. In order to combine differential pairs across ports, both ports must have the same renormalization setting; that is, either ports have **Do not Renormalize** on, or both have it off. For Transient Network solutions, differential pairs cannot include passive terminals. We will use a differential pair via model to assign the differential pairs. See Figure below.



To set up a differential pair:

1. Right-click **Excitations** in the **Project** tree and select **Differential Pairs** on the shortcut menu. **Note:** For a multi terminal wave port, select that port in the **Project** tree and click **Differential Pairs** on the shortcut menu.

   This displays the **Differential Pairs** dialog box. (For Transient Network designs, the Active Column is replaced by Enabled and Matched columns.) This contains table headers for the rows of values defined for each pair. It also contains a field for the renormalizing impedance value and units.



2. Click **New Pair**.

   This adds existing pairs to the Terminals list, and sets default values for the Differential Mode and Common mode. All values can be edited.

   It also lists which terminal is Positive, which is Negative. By selecting the drop-down menus in these fields, you can reassign these values.

   The table row shows the checkbox for the newly defined pair as **Enabled**. Unchecking the box disables the definition for that pair. This can be useful if you later want to redefine terminal normalization, without having to remove the defined pair altogether.

3. If other pairs can be created from the existing Terminals, the **New Pair** button remains enabled.
4. Under **Differential Mode** headers in the table, do the following:

a. If desired, type a new name for the differential mode in the **Name** text box. The default base name is Diff. To specify a new default see: Setting Default Boundary/Excitation Base Names.

b. Unless the **Post Processing** tab selection for the Port is set to **Do Not Renormalize**, you can edit the renormalize impedance value here. You can either specify a real valued renormalizing impedance for the differential mode in the **Ref. Z** text box or use the **Full Port Renormalizing Impedance** text box and the **Set All Diff. Zref.** button or the **Set All Zref** button to set the values.

5. Under **Common Mode** headers in the table, do the following:

a. If desired, type a name for the common mode in the **Name** text box. The default base name is Comm. To specify a new default name, see Setting Default Boundary/Excitation Base Names.

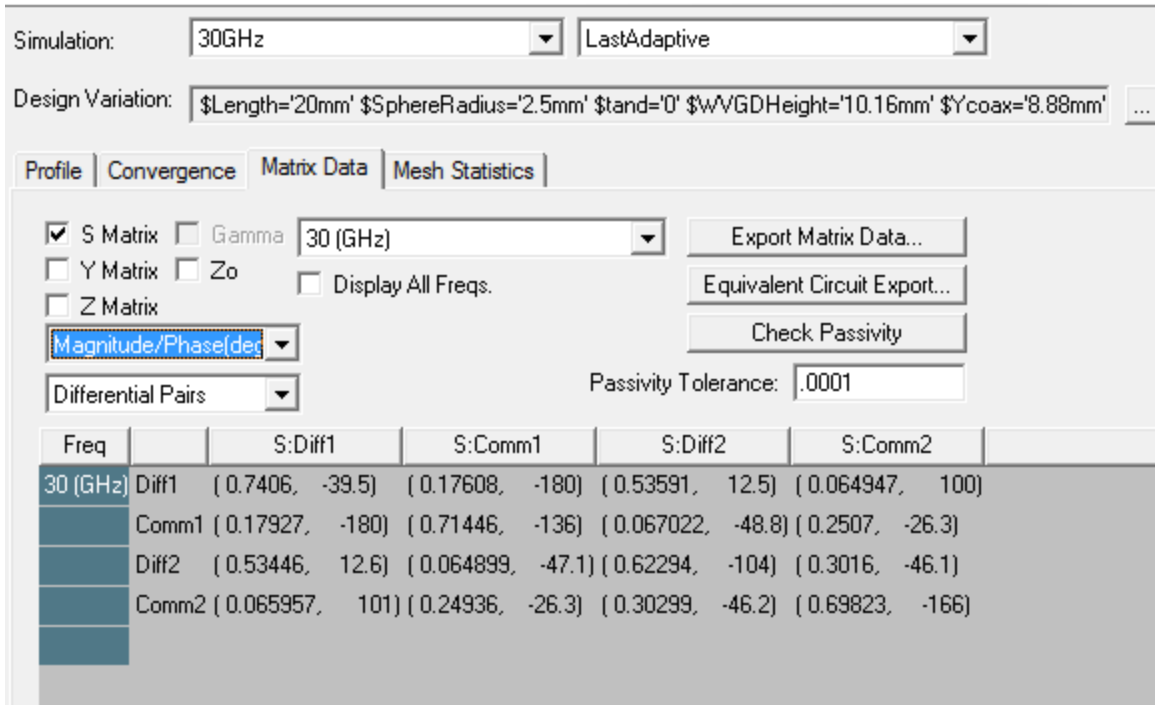
b. Either specify a real valued renormalizing impedance for the common mode in the **Ref. Z** text box, or use the **Full Port Renormalizing Impedance** text box and the **Set All Comm. Zref.** button or the **Set All Zref** button to set the values.

6. If the **New Pair** button is enabled, you can define additional differential pairs.

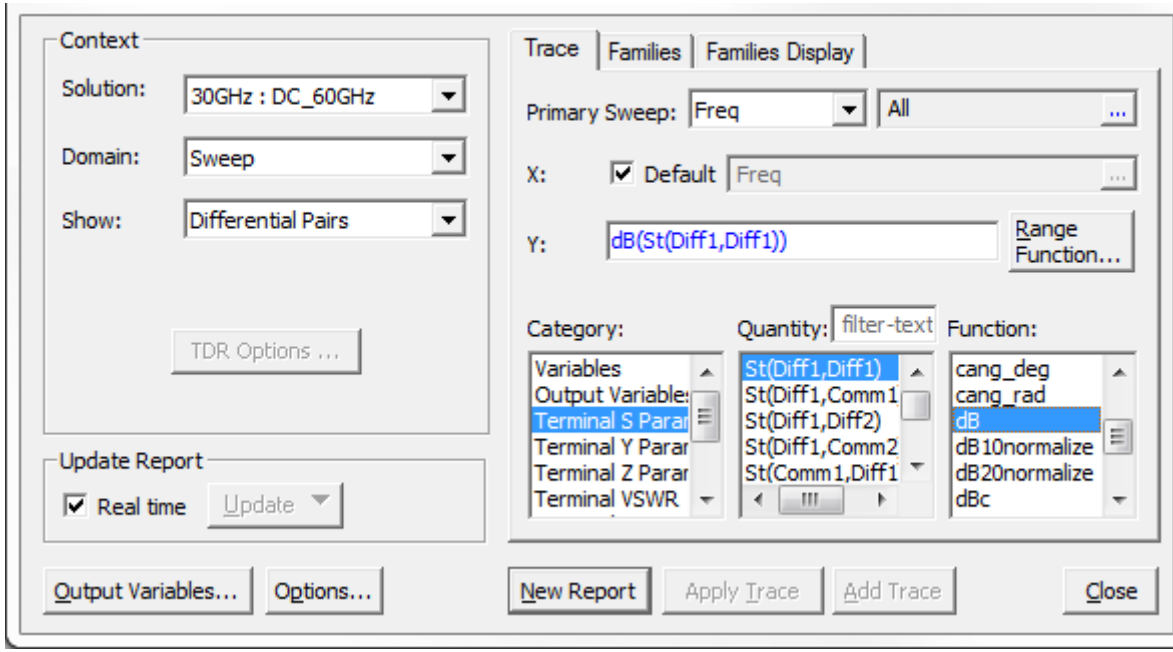
   You can use the command buttons in the Differential Pairs window to **Remove Pair**, or **Remove All Pairs**.

7. To accept the assignments, click **OK** to close the **Differential Pairs** dialog box.

After HFSS has generated a solution, view the common and differential quantities of the differential pair under the **Matrix** tab of the **Solution Data** window as shown below.



When the design has differential pairs (link), the reporter can display quantities for the defined pairs or for the single-ended terminals upon which they are based. A pull-down menu will appear in the **Context** area of the **Report** creation dialog which allows the user to select which quantities will be displayed.



You can freely mix differential and single-ended terminal quantities. However, single ended quantities are computed as if no differential pairs existed. So, in the unlikely case of several terminals where only a subset are combined into pairs, the results may not be as expected.

Related Topics

Creating a New Report

Context Section for Reports.

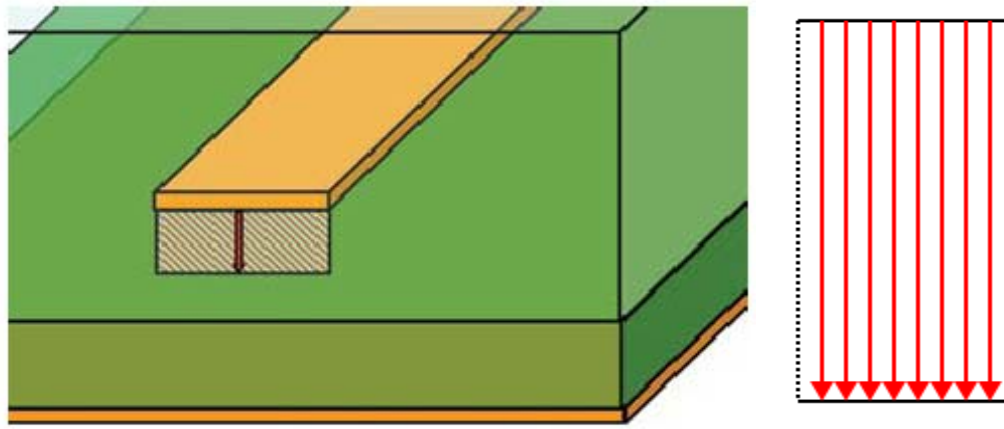
Differential Pairs in HFSS Transient Network

Technical Notes: Computing Differential Pairs

Lumped Ports

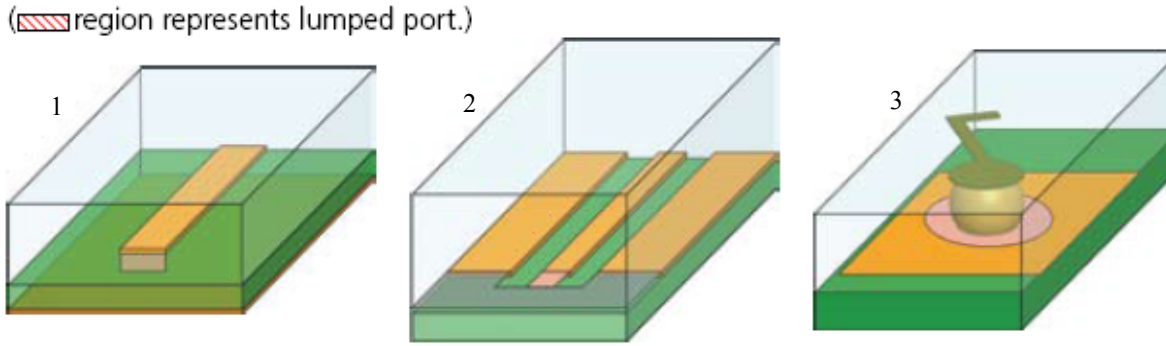
Lumped ports support single mode excitations when S-parameters have to be extracted at internal locations of a model. It can also be used to represent a terminal of a passive component to be subsequently optimized in a circuit simulator using S-matrix description of the model.

For lumped ports all edges that do not touch metal are treated as perfect H boundaries. From this definition the resulting field distribution on the lumped port geometry is solved with the wave port solver. For a rectangular lumped port this results in electric fields orientated parallel to these perfect H sides. See figures below. The physical geometry of the rectangular lumped port carries current with the corresponding H fields resulting in parasitic inductance. For these same rectangular lumped ports the parasitic inductance can be calibrated out of the s-parameter response with the deembedding option for lumped ports.



Examples of Lumped Port

Lumped ports can be used in a variety of ways as shown below.



They vary as follows:

- For 1: The 2D port rectangle touches the signal trace with one edge while the opposite edge touches the ground plane.
- For 2: The 2D port rectangle touches the signal trace with one edge while the opposite edge touches the PEC objects (drawn in grey).
- For 3: the port is an annular ring around the BGA ball where the resulting field patterns closely resemble those for a coaxial TEM mode.

Note: Use lumped ports only for those surfaces that are internal to the model and wave ports to model exterior surfaces through which a signal enters or exits the geometry.

Related Topics

Assigning Lumped Ports for Modal Solutions

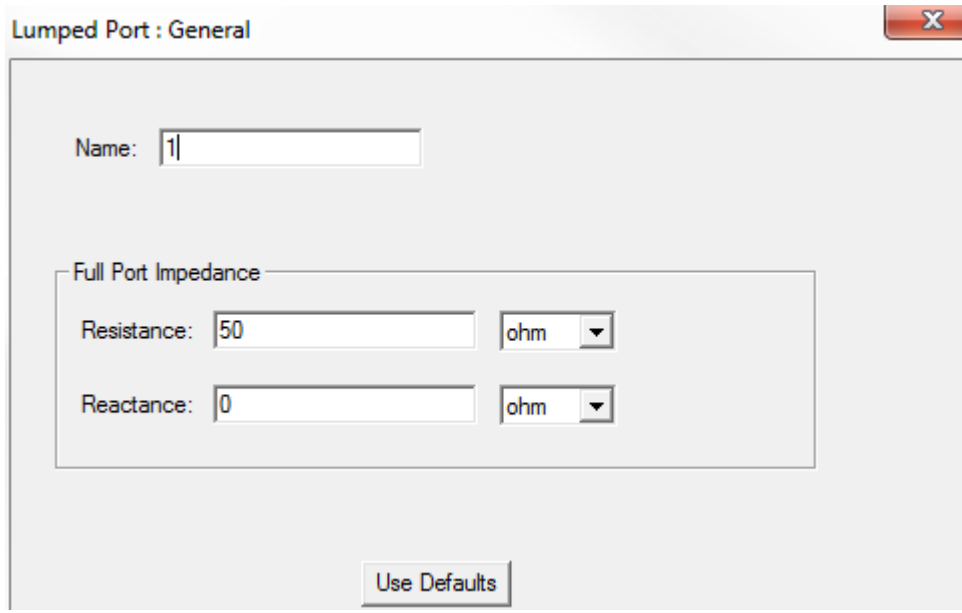
Technical Notes: Lumped Ports

Technical Notes: Calculating Characteristic Impedance

Assign Lumped Ports for Modal Solutions

While assigning lumped ports for modal solutions, you will be prompted to set the complex full port impedance which must be non-zero, set the non-negative resistance, and finally define the integration line for the single mode. The steps are as follows:

1. Select a surface to which you want to assign the port and click **HFSS** or **HFSS-IE>Excitations>Assign>Lumped Port** to bring up the **Lumped Port: General** dialog box.

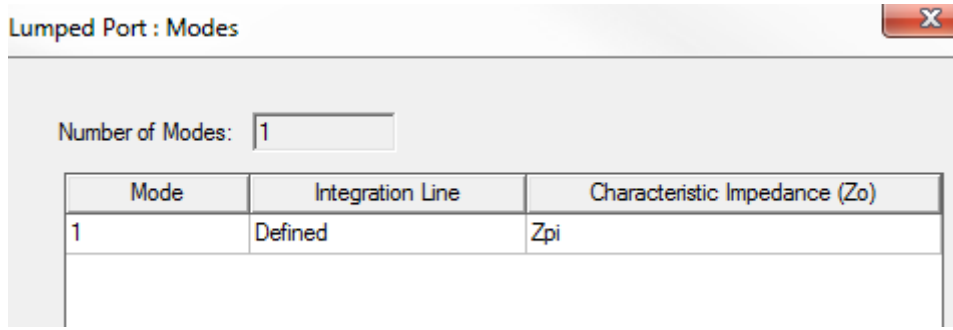


2. Define the complex **Full Port Impedance** in the **Resistance** and the **Reactance** fields.

   Note: The reference impedance is meant to represent the component modeled by the lumped port. You can assign a variable to these values. This variable can be dependent on the frequency, which allows use of a dataset for frequency dependent impedance.

3. Click **Next** to display the **Lumped Port: Modes** dialog box.

   Note: The field **Number of Modes** is not editable because only one mode is allowed.



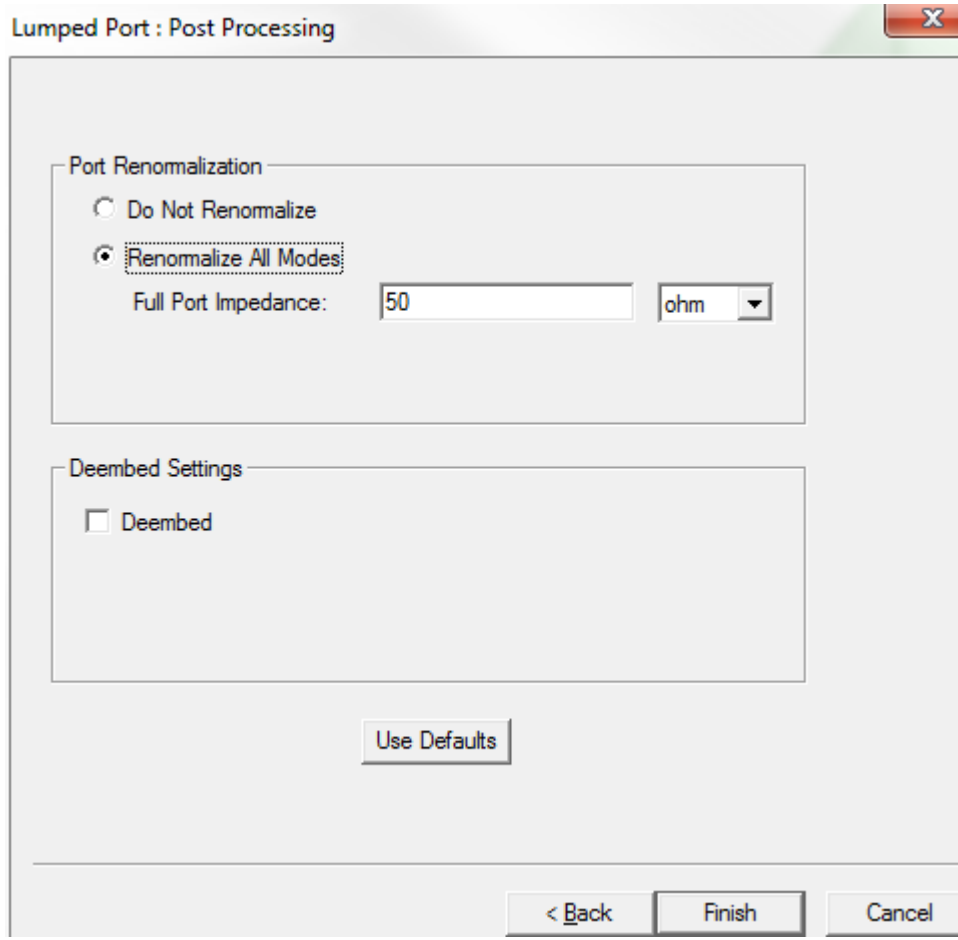
4. Integration line must be drawn for a lumped port.

   The **Characteristic Impedance (Zo)** column shows the Zpi method usually used to calculate the characteristic impedance. If Zpi is zero, HFSS uses Zpv.

   For definitions of how HFSS defines these values, see Calculating the Zpi , and Calculating the Zpv.

Lumped Port: Post Processing

The parasitic inductance can be calibrated out of the s-parameter response with the deembed option for lumped ports. Click **Next** on the **Lumped Port: Modes** dialog box, to bring up the **Lumped Port: Post Processing** dialog.



Values here affect S-Parameters. They will also affect the spectral fields if you selected the **Include Post Processing Effects** on the **Edit post process sources** dialog box. By default, lumped ports are renormalized to a 50 Ohm full port impedance. You can override the default by entering a value of your choice in the **Full Port Impedance** field. If you want to enter a complex impedance, enter it in the form *<re> + <im>j*.

For a lumped port defined on a rectangular geometry HFSS provides an option to deembed the par

Note: For more information about the port calibration technique, see the topic When HFSS Needs Port Calibration (Deembedding).

The Deembed option can be turned on or off as a post processing operation without invalidating solutions. If multiple lumped ports are selected in the Excitations List panel, changes to the common Deembed property change the setting for all lumped ports at once, but only for those that support calibration.

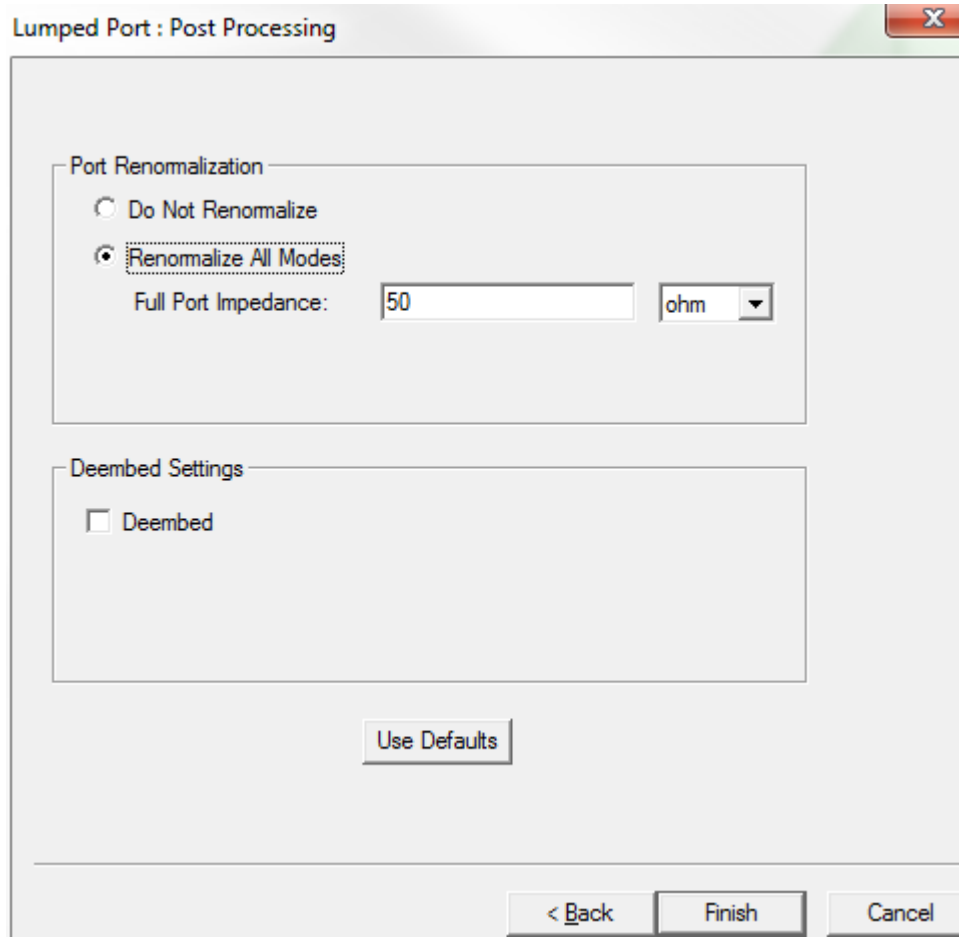
If a port is selected for calibration but the validation checks fail for some variation (which could happen with a parametric solve), then the matrix data will not load for that variation. Also, if the fields are to include port post processing effects, then the fields do not load.

Related Topics

Technical Notes: Lumped Ports

Assign Lumped Ports for Terminal Solutions

As with lumped ports for modal solutions, while assigning lumped ports for terminal solutions, you will be prompted to set the complex full port impedance which must be non-zero, set the non-negative resistance, and finally define the integration line for the single mode. The steps are as follows:



1. Select the object face to which you want to assign the port.
2. Click **HFSS>** or **HFSS-IE>Excitations>Assign>Lumped Port**.
3. Define the complex **Full Port Impedance** of the port in the **Resistance** and the **Reactance** text boxes. You can assign a variable as these values. This variable can be dependent on the frequency, which allows use of a dataset for frequency dependent impedance.

Related Topics

Technical Notes*:* Lumped Ports

Technical Notes: When HFSS Needs Port Calibration (Deembedding)

HFSS-IE Lumped Ports

The lumped ports in HFSS-IE ports are different than those in HFSS. The HFSS-IE lumped ports impress a one volt difference between the terminal and its reference while an HFSS lumped port impresses an electric field between the terminal and its reference.

To ensure a valid port, the maximum distance from the terminal to the reference should be less than a twentieth of a wavelength. If this condition is violated, a warning occurs.

For HFSS-IE:

- For auto assignment to work you must enable **Auto-assign terminals on ports** on the HFSS-IE Options: General Options Tab.

Otherwise, for either HFSS or HFSS-IE:

- Manually assign a terminal. Select any connected edge(s) and/or face(s) of conductors that touch the port.
- Select **Excitations>Auto Assign Terminals**.

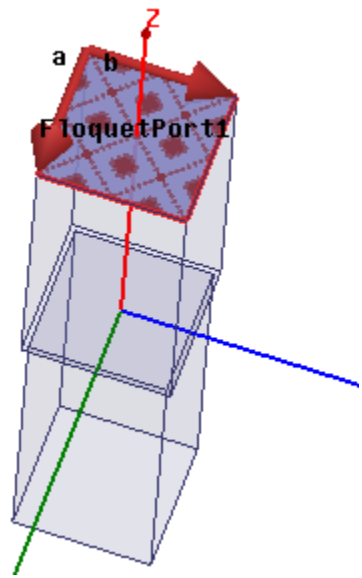
This will bring up a dialog box for you to select the objects used for "reference conductors."

Floquet Ports

The Floquet port in HFSS is used exclusively with planar-periodic structures. Chief examples are planar phased arrays and frequency selective surfaces when these may be idealized as infinitely large. The analysis of the infinite structure is then accomplished by analyzing a unit cell. Linked boundaries most often form the side walls of a unit cell, but in addition, a boundary condition is required to account for the infinite space above. The Floquet port is closely related to a Wave port in that a set of modes ("Floquet modes") represents the fields on the port boundary. Fundamentally, Floquet modes are plane waves with propagation direction set by the frequency, phasing, and the geometry of the periodic structure. Just like Wave modes, Floquet modes too have propagation constants and experience cut-off at low frequency. When a Floquet port is present, the HFSS solution includes a modal decomposition that gives additional information on the performance of the radiating structure. As in the case of a Wave port, this information is cast in the form of an S-matrix interrelating the Floquet modes. In fact, if Floquet ports and Wave ports are simultaneously present, the S-matrix will interrelate all Wave modes and all Floquet modes in the project.

Example of Floquet Ports

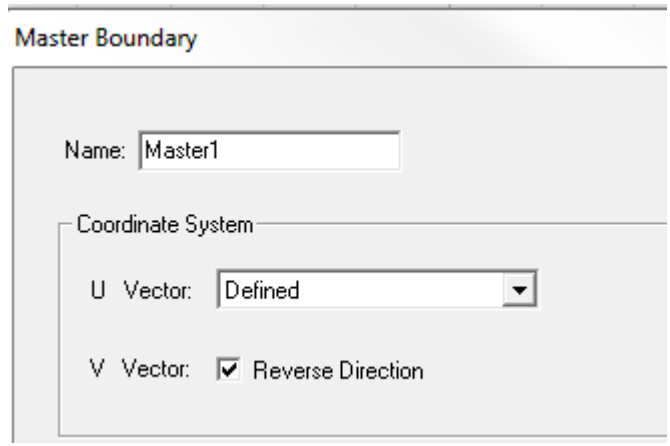
As a simple example, consider an infinite array of radiating rectangular apertures in a ground plane. The figure below depicts a simple HFSS model for the unit cell of the infinite array. The model consists of two boxes. The lower box represents the feeding waveguide and the upper box is the unit cell for the region above the plane. The dimensions and geometry of the unit cell reflect the lattice vectors of the array. Linked boundaries are defined on the cell walls and a Wave port provides the array element excitation.



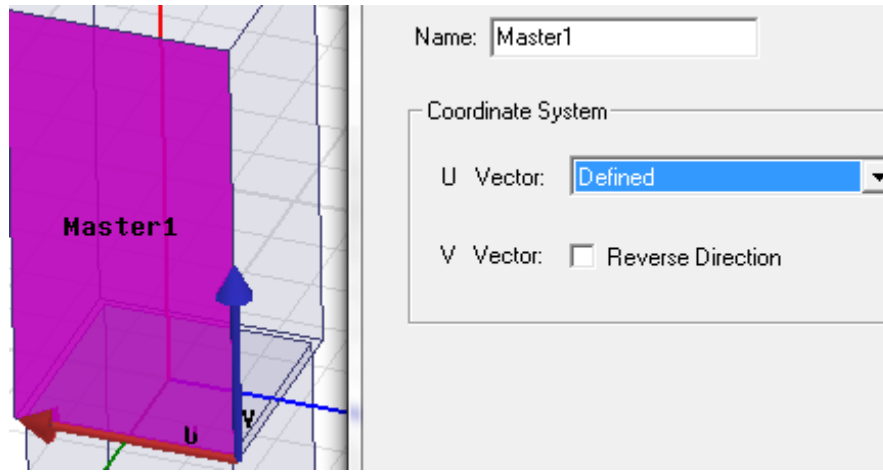
Assign Master and Slave Boundaries

In this section we will illustrate a key requirement for setting up a unit cell--the perimeter of a Floquet port must be covered by Master and Slave boundaries. You will assign the boundaries as follows:

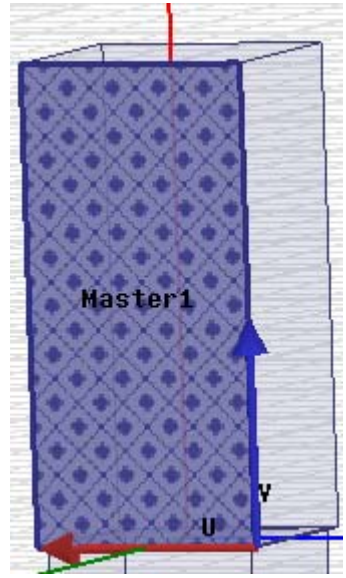
1. Select the face of the box and click **HFSS**>**Boundaries**>**Assign**>**Master**.



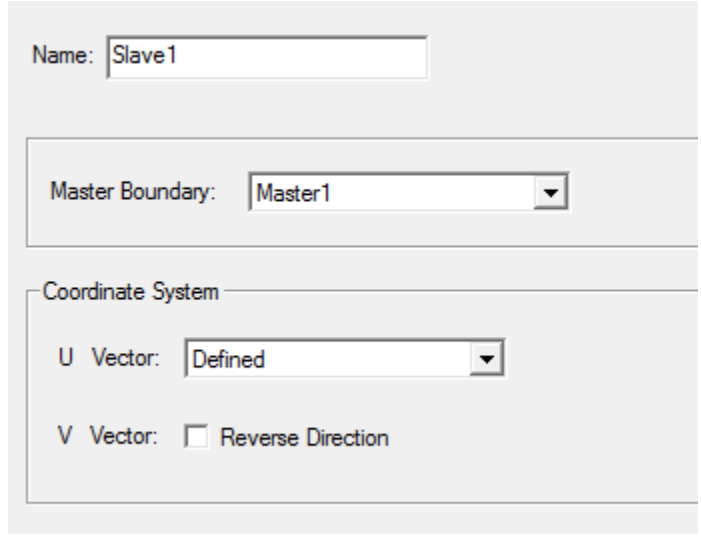
2. Click the corner of the box and draw the U-V vectors as shown below.



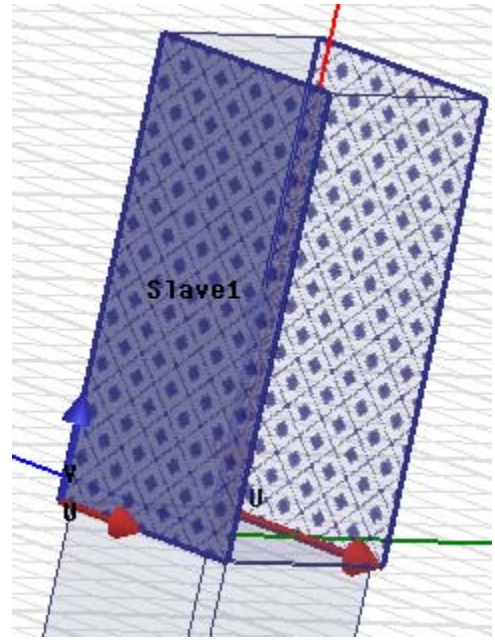
The Master1 boundary gets assigned as shown in the figure below.



3. Access the face opposite to Master1 and click **HFSS**>**Boundaries**>**Assign**>**Slave**.



4. Assign the slave boundary as shown in the figure below.



Direction of the U-V Vectors

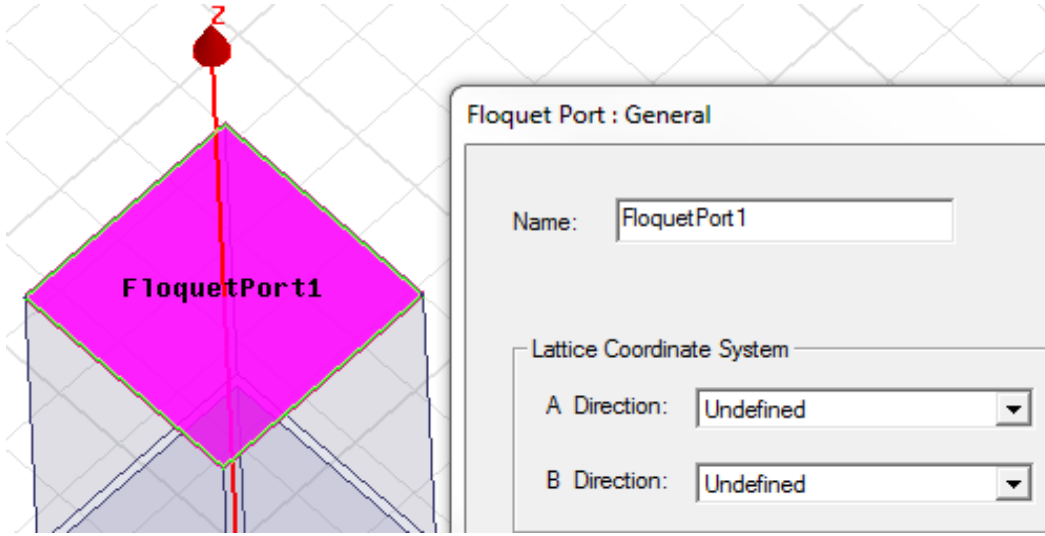
HFSS uses the U-V vectors to set up a local co-ordinate system. A point on the Master boundary must correspond to that on the Slave. A point on the Slave needs to be paired on the Master so that a one-to-one correspondence can be established. Such a co-ordinate system should be constructed and aligned properly in the same direction so that the mapping is accomplished successfully. Therefore, when the V-vector is directed downwards (i.e. when it is below the U vector), you must check the **Reverse Direction** option to keep it on the face of the box and not pointing outside of it.

Assign Floquet Ports

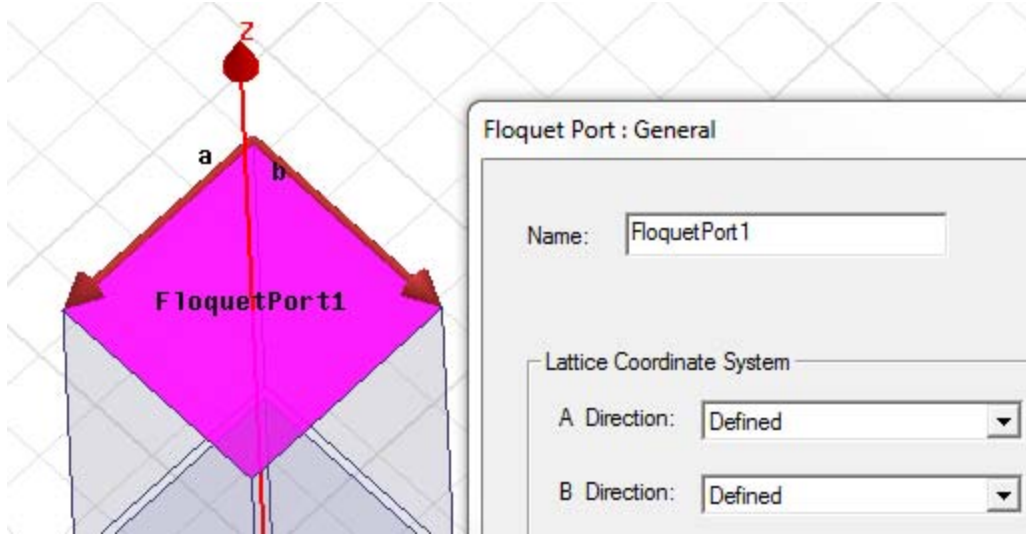
This section describes how to assign Floquet Ports on the unit cell after you define the Master and Slave boundaries. To assign the Floquet port perform the following steps.

1. Select the top face of the unit cell, right click, and select **Assign**>**Excitation**>**Floquet Port**

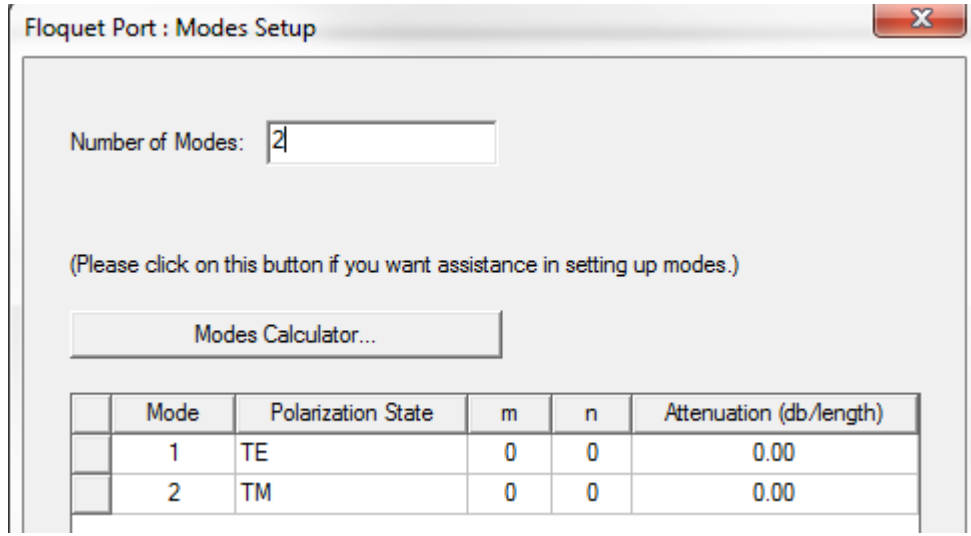
from the shortcut menu.



2. Specify the A and B directions for the Lattice coordinate system. These define the periodicity of the planar lattice. The vector arrows must start and end at points on the face of the Floquet port and must have a common initial point.



3. The **Modes Setup** window displays a field for the **Number of Modes**, a button for access to the **Modes Calculator**, and a table.



4. On the post processing panel, if needed, the deembed distance can be specified.

Floquet Port Dialog Box

This section describes the settings on each of the tabs available on the **Floquet Port** dialog box.

Floquet Port: Mode Setup

Floquet Port: Post Processing

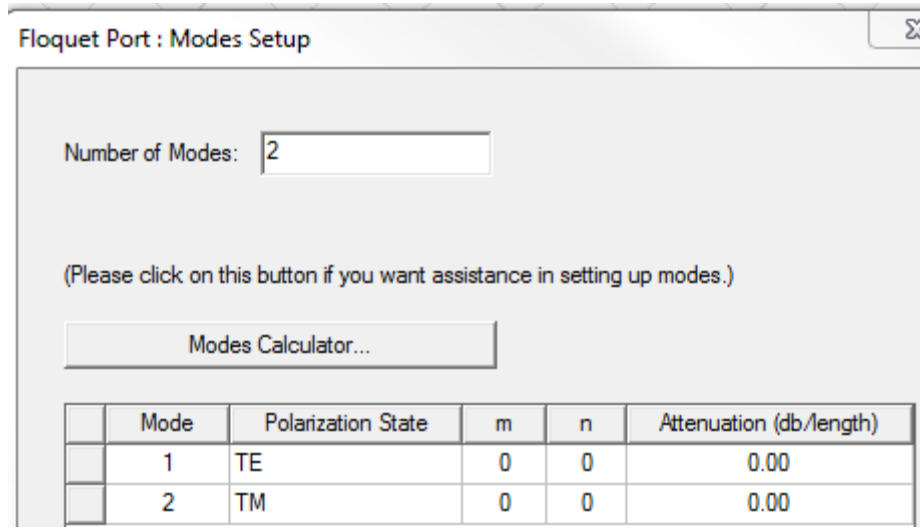
Floquet Port: 3D Refinement

Floquet Ports: Lattice Coordinate System

Floquet Port: Modes Calculator

Floquet Port: Mode Setup

In general Floquet modes are specified by two modal indices and a polarization setting. These designations resemble the textbook notation for rectangular waveguide modes, such as ``TE10."



The default mode table specifies a pair of Floquet modes. The default modes both have modal indices equal to zero and are sometimes called "specular" modes, which are always an essential part of the Floquet mode set. For general frequency and scan conditions, other higher-order Floquet modes will be needed. A modes calculator, invoked by selecting the **Modes Calculator** button, is available to set these up for the user.

The values under **Attenuation** represent the modal loss in amplitude along the direction normal to the Floquet port plane. The numbers in this column are computed by the modes calculator that will help you decide which modes to keep.

Attenuation for a mode is a function of both the frequency and the scan angle specified in the modes calculator. When the latter includes more than one scan direction, the least amount of attenuation experienced by the mode over all the specified scan directions.

Thus when the table gives a value of 0 dB, at one or more scan directions specified in the modes calculator the particular mode propagates without attenuation. Similarly, when the table displays say 60 dB, a 60 dB per unit length is the least amount of attenuation at all specified scan directions. At any given direction, only the same or larger attenuations (for example, 70 dB per length) will occur. To improve simulation efficiency and interpret the results easily eliminate any modes that are not necessary. Do this by editing the **Number of Nodes** field in the **Modes Setup** tab.

Note: The list is trimmed from the bottom.

To change the order of items in the final Modes list, drag each corresponding line by the square box at the left of each row.

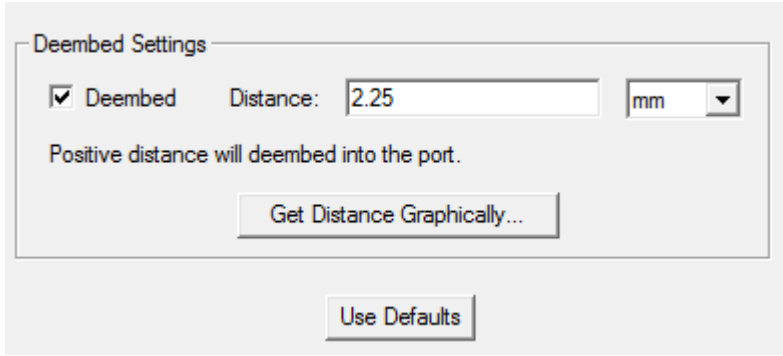
Related Topics

Floquet Ports : Modes Calculator

Floquet Port: Post Processing

This tab lets you specify a de-embedding distance. This optional post processing step is employed when you are interested in the phase of the S parameter elements. The interface for deembedding a Floquet port is the same as that for a Wave port.

This panel contains settings that affect the fields once the field solution is complete.



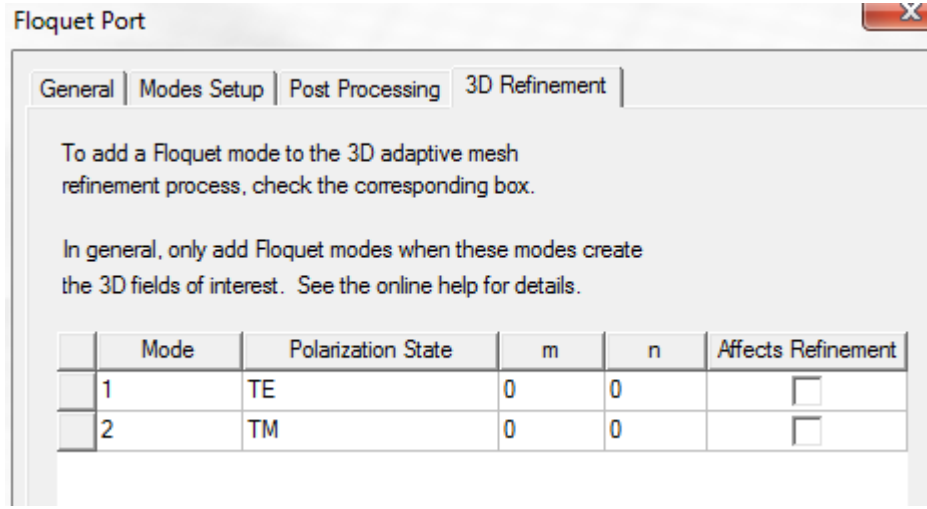
This enables the distance and units field for the positive value to deembed the port into the mode, and for a negative value to deembed the port out of the model.

Floquet Port: 3D Refinement

This panel contains the **Affects Refinement** checkboxes which allows you to specify **Floquet** modes in the 3D adaptive refinement process. Typically, you select no modes or only one or both

specular (TE00 or TM00) modes. Selecting more than this may affect the efficiency and accuracy of the solution process.

In 3D adaptive refinement the generated mesh is a compromise which simultaneously represents the 3D field patterns of every mode included in the adapt process. If the field patterns of certain modes represent the fields of interest and others do not, excluding the latter from the adapt process will result in a "targeted" mesh that better represents the excitation field pattern.



For antenna array simulations in which the active impedance or embedded-element pattern is sought, the Wave or Lumped ports modeling the feed structure provide the fields of interest. In this case, no Floquet modes should be included in the adapt process. On the other hand, if the per-cell RCS is of interest, one or both specular Floquet modes provide the fields of interest and should be included in the 3D adapt process by checking the corresponding **Affects Refinement** boxes. Similarly, for an FSS simulation with two Floquet ports, specular modes provide the fields of interest and should be selected to participate in the 3D adapt process.

In certain simulations (for example, a frequency-selective surface) you will set up a second Floquet port. Note that when you do this HFSS automatically copies the lattice vectors, modes table, and 3D refinement settings from the first Floquet port to the second.

Related Topics

Technical Notes: Master and Slave Boundaries

Technical Notes: Deembedding

Technical Notes: Floquet Ports

Floquet Ports : Modes Calculator

Floquet Ports: Lattice Coordinate System

In the **Floquet Port** dialog, **General** tab, for each Vector (A and B Direction):

1. Select the drop down menu and click **New Vector**.

   This opens a **Measure Data** dialog and causes the cursor to drag a visual marker that drops a dashed line to the reference plane, and shows a location indicator on the Floquet plane.

2. Drag the marker to select a location for the Direction vector. Click to set the origin point, and drag and click to specify the position 2 point that defines the direction from that origin. The vector arrows must start and end at points on the face of the Floquet port and must have a common initial point.

3. Clicking the second point closes the **Measure Data** dialog, and exits the New Vector mode. The drop down menus for Position A and Position B now include an entry called "Defined" along with "Undefined" and "New Vector."

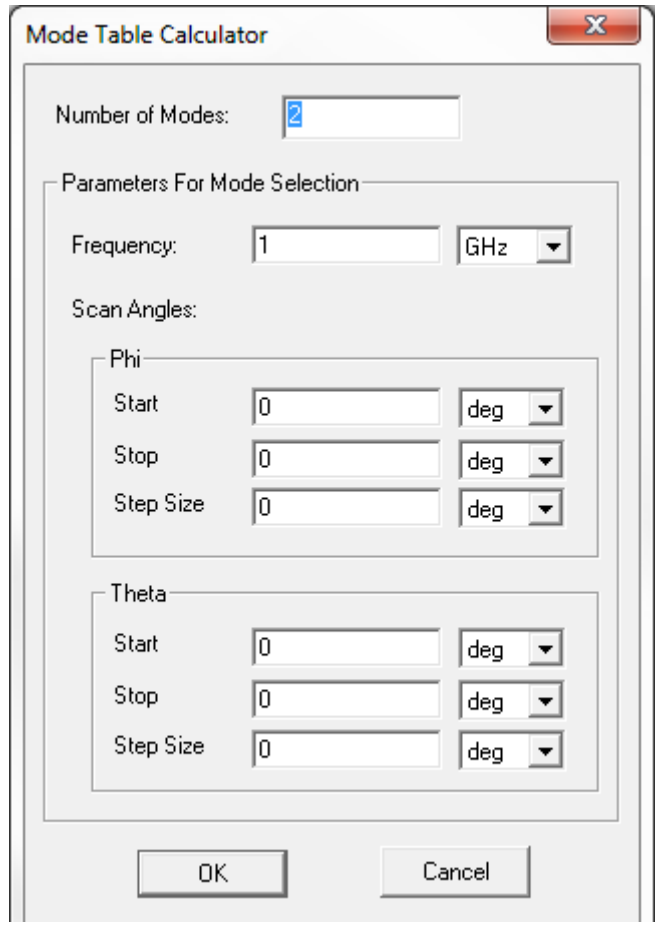
Related Topics

Assigning Floquet Ports

Floquet Port: Modes Calculator

1. In the **Floquet Port: Modes** tab, if you select the **Modes calculator** button, the **Mode Table**

Calculator window displays.



Enter the following fields:

- **Number of modes** - you can trim this value later, as you learn which modes are needed and which are not.
- **Frequency** - if the problem setup contains one or more frequency sweeps, usually you will set this value to the highest frequency.
- **Scan Angles** -enter the values and select units for Phi and Theta, including Start, Stop, and Step size. The **Mode Table Calculator** calculates a set of Floquet modes on the basis of all the angles defined.

These inputs constitute the information required to create a set of recommended modes for the Floquet port. The inputs are used by the mode selection algorithm but do not affect the problem setup.

2. Click **OK** to leave the **Mode Table Calculator** and to compute the recommended list of modes.

   The new modes table appears on the **Modes Setup** tab of the **Floquet Port** properties/setup window. The attenuation associated with a listed mode represents the minimum (or worst case) for that mode over the range of scan angles.

Related Topics

Assigning Floquet Ports

Floquet Ports : 3D Refinement

Incident Waves

An incident field is the electromagnetic field in the absence of any scatterers. We suppose that the incident field is present everywhere and it comes from a source residing in some location. The source can even be another HFSS project or an SIwave project.

Incident waves can be of the following types:

- Plane Wave
- Hertzian-Dipole Wave
- Cylindrical Wave
- Gaussian Beam
- Linear Antenna Wave
- Far Field Wave
- Near Field Wave

For incident analytical waves (plane waves through linear antenna waves) HFSS supports two basic methods: scattered field formulation and total field formulation.

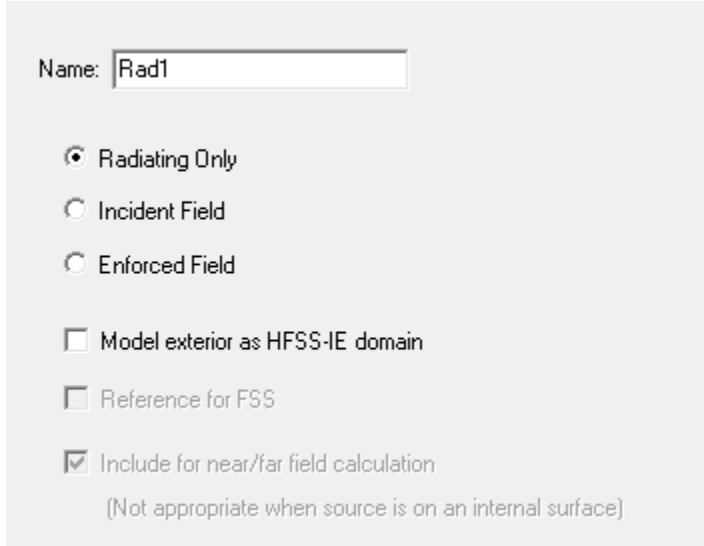
$\mathbf{E}_{total} = \mathbf{E}_{incident} + \mathbf{E}_{scattered}$

The total field formulation is useful for viewing weak total fields while scattered field formulation is useful for viewing weak scattered fields.

For near field and far field waves we are merely specifying the fields on the radiation surfaces. So, we cannot view only the total fields.

Radiation Boundary Panel

When assigning an incident wave either a radiation boundary, PML, or Finite Element Boundary Integral (FEBI) method must be defined at least on one section of the surface of the model. The Radiation Boundary panel is shown below. For the options in the Radiation Boundary panel to be active, at least one incident wave must be defined.



The options on the panel will impact the simulation as follows:

- For **Radiation Only** HFSS will use the scattered field formulation.
- For **Incident Field** HFSS will use total field formulation.
- For **Enforced Field** HFSS will use total field formulation enforcing the tangential component of the magnetic field to be equal to that of incident field.
- **Model exterior as HFSS-IE** domain is a FEBI method option.
- **Reference for FSS** can be used only if Master and Slave boundaries exist.
- **Include for near/far field calculation** can be checked if the source project resides outside the target project. This option is meant only for near or far field incident waves.

  Note: If you select this option, it is your responsibility to ensure that the source project is located outside the target project. The location can be set on the Translation of Source Origin Relative to the Design sub-panel.

For more information see Assigning Radiation Boundaries.

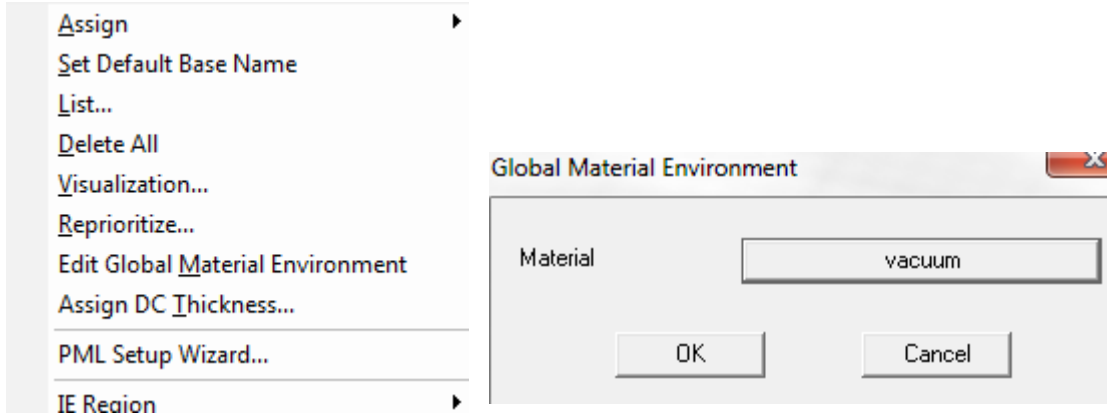
Global Material Environment

A plane wave is described by the following equation.

$$\vec{E} = \vec{E}_0 \, e^{-j\vec{k}\cdot\vec{r}}$$

The **k** vector also depends on the material of the radiation boundary. More precisely it is the wave number of the global background material.

You can set the global material environment which is otherwise vacuum by default. Right click **Boundaries** and select **Edit Global Material Environment** to bring up the dialog box.



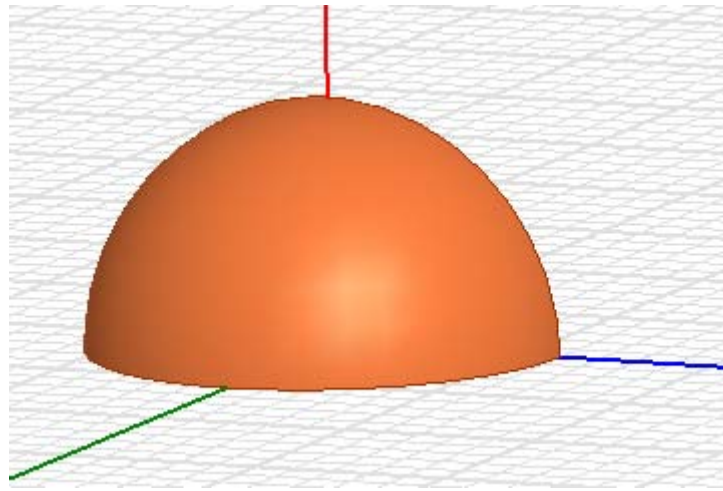
The material specified in the **Global Material Environment** panel can impact an HFSS design in the following three areas:

- All incident waves except evanescent plane waves and near and far field links are assumed to have been generated in the global background material.
- All fields outside the computational domain are obtained using the global background material.
- The currents on exterior HFSS-IE regions are computed assuming the global background material.

Note The PML and simple ABC are constructed assuming the interior material touching the radiation surface is the same as the exterior global background material. The actual material used is the interior in this case in order to allow different materials to be in contact with the radiation surface as is the case when truncating the substrate of a PCB design. To obtain consistent results when viewing fields outside the computational domain, ensure that the interior material touching the radiation surface is the same as the global background material.

Incident Plane Wave

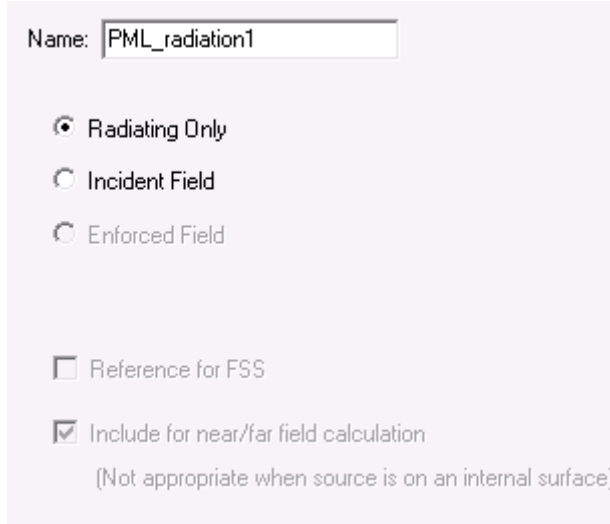
A plane wave propagates in one direction and the fields are perpendicular to the direction of propagation. A project which simulates the RCS of a dielectric sphere is shown below. We take advantage of the symmetery and model just 1/8th of the structure.



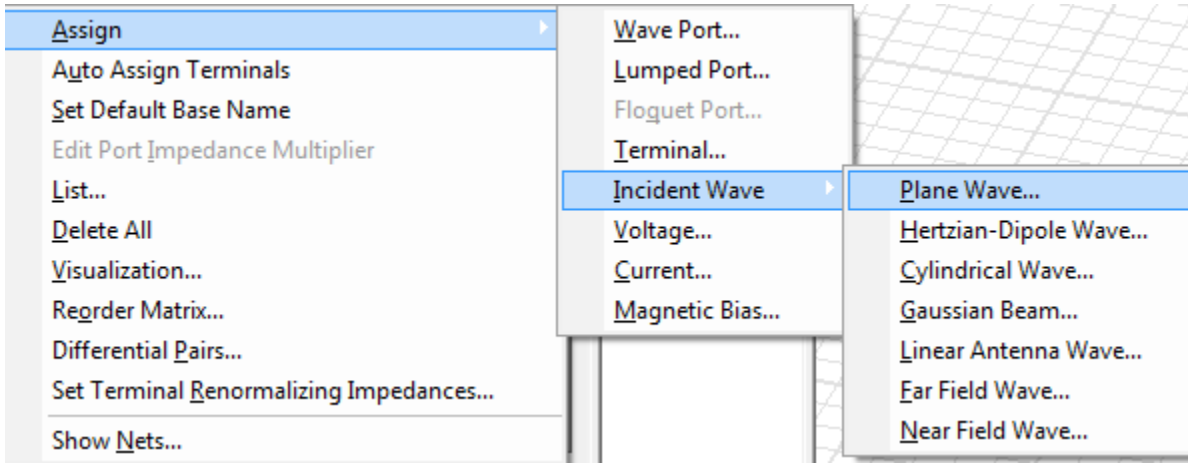
First we use the E and H symmetry boundary conditions where we cut the sphere and then, apply PML boundaries on the remaining surfaces.

Since this is a scattering problem and the scattered field is weak, use the option **Radiation Only**.

For more information about these options, see Radiation Boundary Panel.

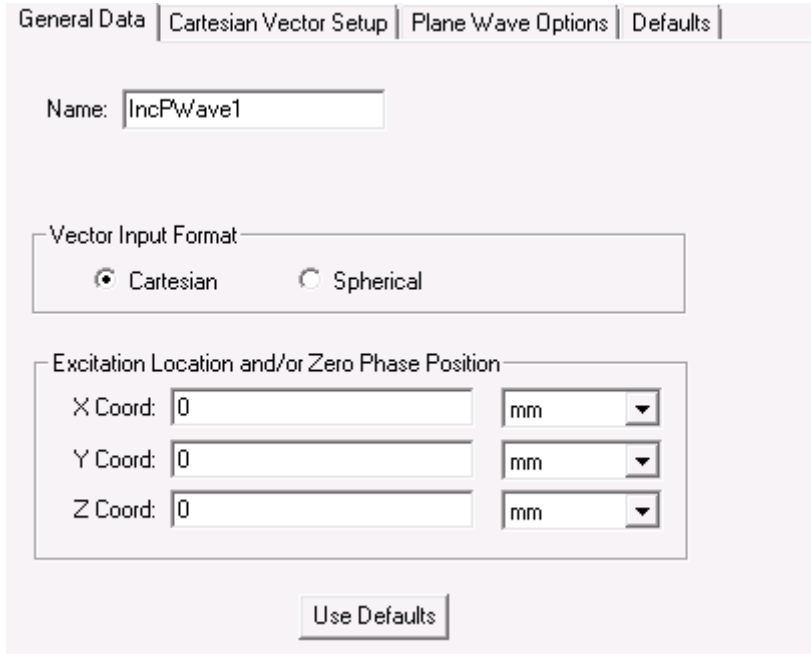


To assign a plane wave in HFSS, right click **Excitations** on the project tree and select **Assign>Incident Wave>Plane Wave.**



The **Incident Wave Source : General Data** dialog box appears as shown below. You can select either **Cartesian** or **Spherical** for the **Vector Input Format**.

First we will describe the effect of choosing **Cartesian** followed by **Spherical**.



Cartesian Vector Setup

Enter the Cartesian Co-ordinates on the **Incident Wave Source** dialog box to set the excitation location and click the **Cartesian Vector Setup** tab.

A plane wave is described by the following equation.

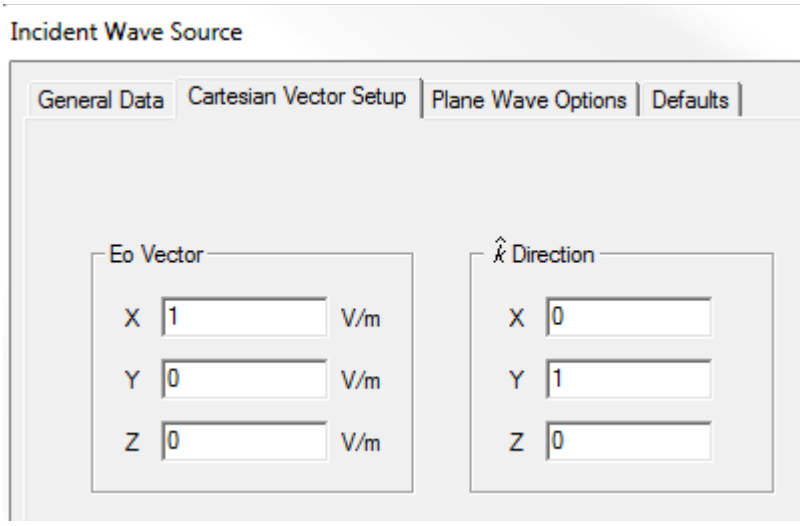
$$\vec{E} = \vec{E}_0 \, e^{-j\vec{k}\cdot\vec{r}} \qquad \text{where} \qquad \vec{k} = k\hat{k}$$

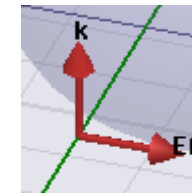
where k

= wave number of the global background material for regular/propagating plane waves.

$= (\beta + j\alpha)$ for evanescent waves since they do not depend upon the background global material.

In the **Cartesian Vector Setup** panel you can specify the $\mathbf{E_0}$ vector and the direction of the unit vector. It is your responsibility to ensure the direction of propagation of the plane wave is perpendicular to that of $\mathbf{E_0}$. Based on the panel settings notice the $\mathbf{E_0}$ and **k** vectors below.

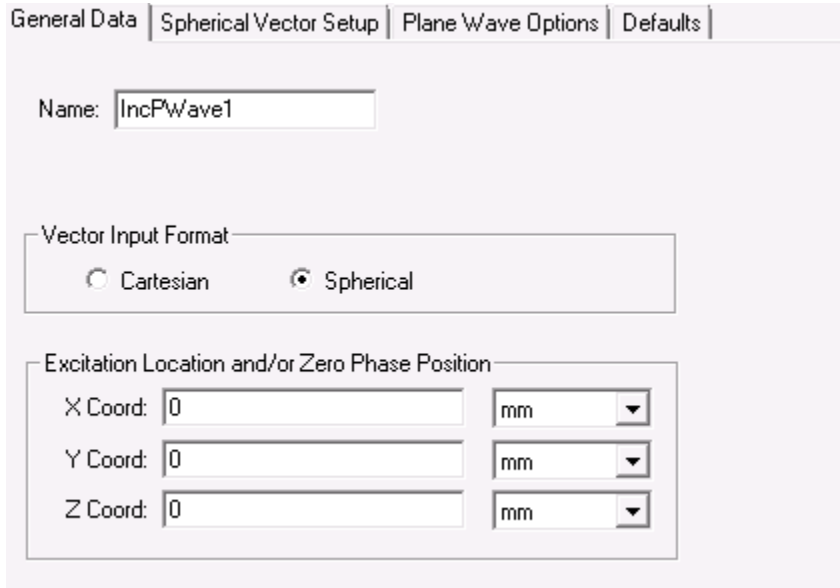




Spherical Vector Input Format

For the equivalent Spherical Setup first click the radio button Spherical on the **Incident Wave Source** dialog box so that the vectors align similarly per your settings in the Cartesian Vector Setup. See the figures below. If you use the Spherical vector setup, HFSS will ensure that the direction of the $\boldsymbol{E_0}$ **vector** is perpendicular to that of the **k** vector.

1. On the **Incident Wave Source** dialog box, select the radio button **Spherical**.

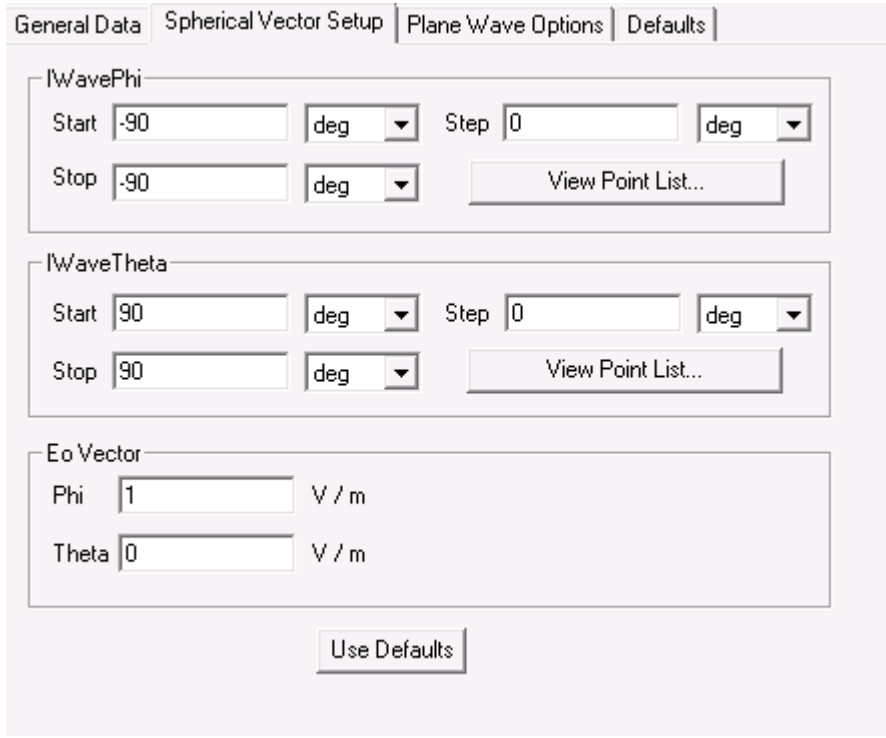


2. Click the **Spherical Vector Setup** and edit the **IWavePhi**, **IWaveTheta**, **Eo Vector** fields.

   Note: If you enter values in the **Step** fields and click the **View Point List** button, you can see all the phi or theta values.

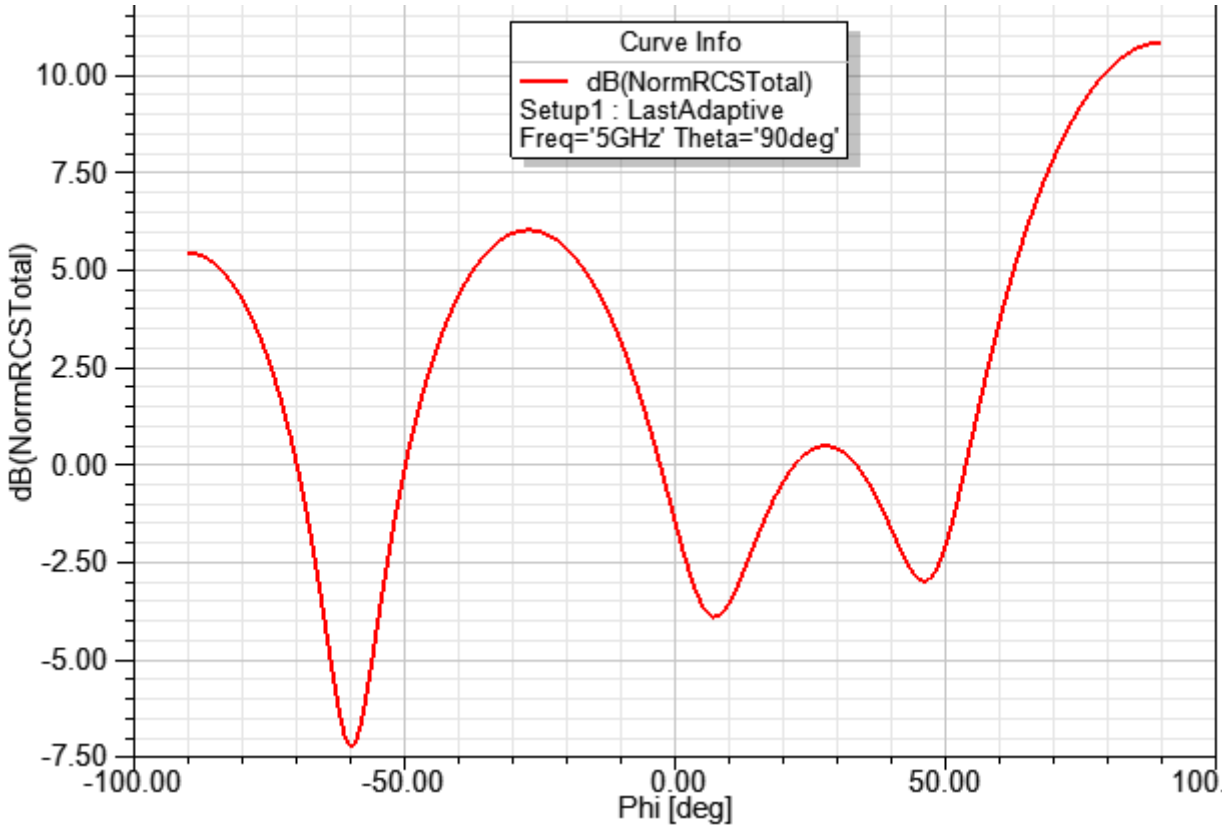
   A spherical grid is created when θ is swept through each ϕ point. At each grid point, an incident wave is present traveling towards the origin of the coordinate system for the design. The number of incident waves and grid points can be calculated by multiplying the number of ϕ points by the θ points.

   Note: Only a single incident wave angle can be defined for periodic structures which are defined with master and slave boundaries.



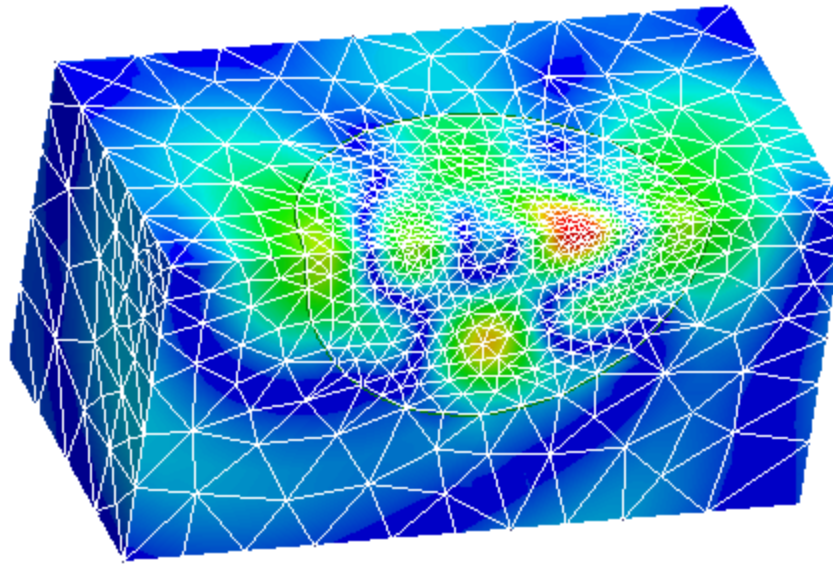
Note: Whenever additions/changes are made to incident waves that affect fields, it invalidates those solutions that can possibly have fields. Meshes are not invalidated.

The plot of the RCS of a dielectric sphere for an incident regular/propagating plane wave are shown below.



The Mesh Plot and the field are as shown in the figure below. The mesh is denser where the field is

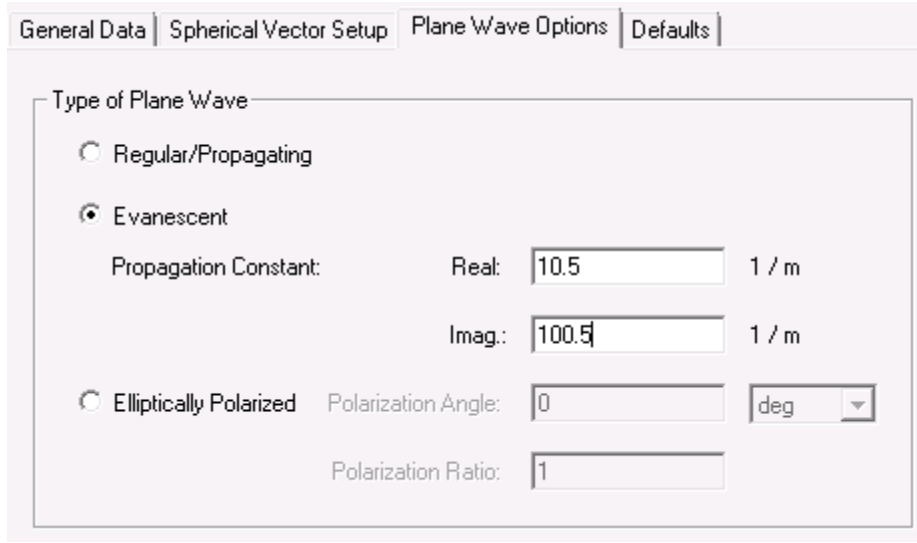
high due to the adaptive meshing feature in HFSS.



Plane Wave Options

In the example shown a Regular/Propagating plane wave set up was used. There are also other plane wave options that you can use as applicable.

On the **Incident Wave Source** dialog box click the **Plane Wave Options** tab to define the type of plane wave.



On this panel you can set the plane wave as a **Regular/Propagating** which is the default or **Evanescent**. The last option will make the E field to be elliptically polarized.

A plane wave is described by the following equation.

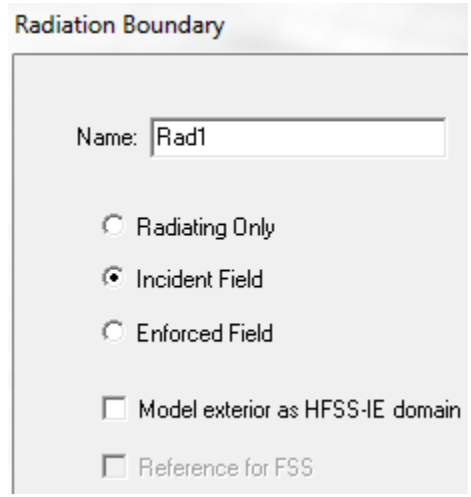
$$\vec{E} = \vec{E}_0 \, e^{-j\vec{k}\cdot\vec{r}} \qquad \text{where} \qquad \vec{k} = k\hat{k}$$

For evanescent wave k = $(\beta + j\alpha)$.

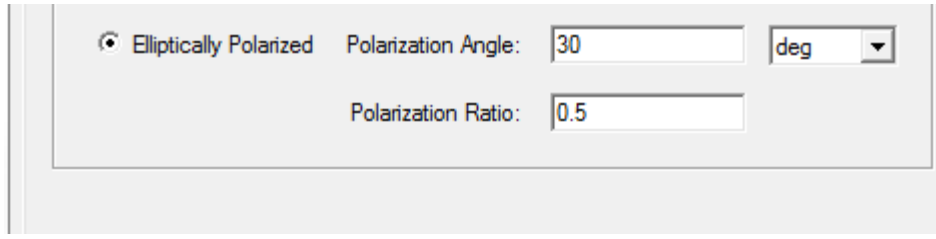
Evanescent waves do not depend upon the background global material.

Do not define an evanescent plane wave for a scattered field formulation. If you set **Radiation Only** on the **Radiation Boundary** panel for evanescent plane wave, the HFSS validation check does not pass. This is because HFSS cannot perform a scattered field formulation when the evanescent plane wave is a source and generates an error message.

For the **Evanescent** plane wave type, select the **Incident Field** option on the **Radiation Boundary** panel. Selecting the **Incident Field** option triggers HFSS to use the total field formulation.



Note: See Polarization of the Electric Field for a technical discussion of polarization angles, and a definition of Polarization Ratio.



Related Topics

Active and Passive Excitation in HFSS Transient

Technical Notes: Incident Waves

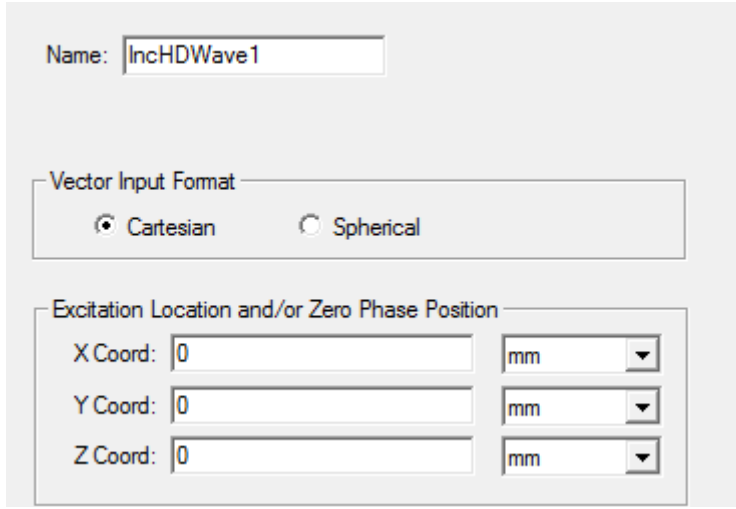
Technical Notes: Evanescent Plane Wave Equations

Hertzian-Dipole Wave

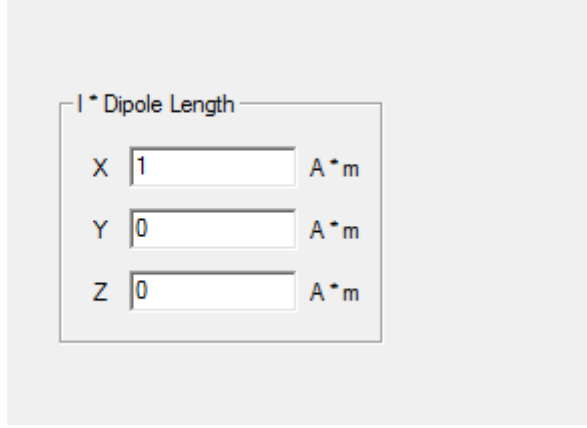
An incident Hertzian-Dipole wave can be specified as either an Electric dipole or a Magnetic dipole. The Electric dipole simulates the field of an elementary short dipole antenna placed at the origin. The Magnetic dipole is useful for EMC/EMI applications. Specify a Hertzian dipole as follows:

Cartesian Coordinates

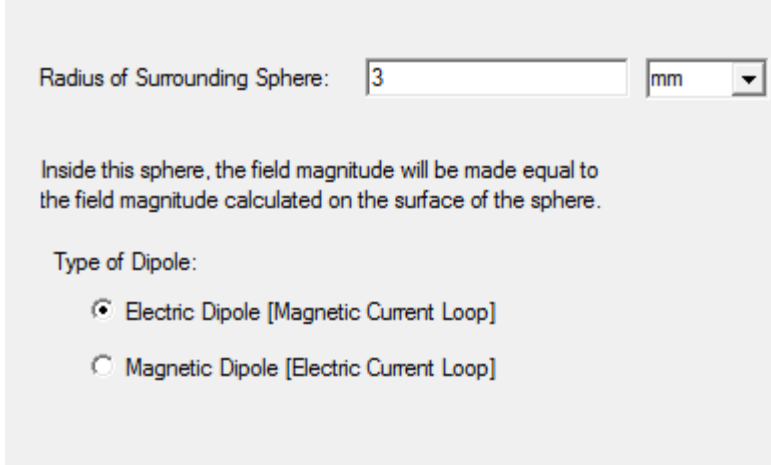
1. Click **HFSS>Excitations>Assign>Incident Wave>Hertzian-Dipole Wave**.



2. Select **Cartesian** and click **Next**, the subsequent dialog lets you define the dipole length.



3. Define the types of dipole and the radius in the **Hertzian-Dipole Wave Options** dialog box.



Related Topics

For more information, see the following topics in Technical Notes.

Incident Waves

Spherical Wave (Electric Hertzian Dipole) Equations

Spherical Wave (Magnetic Hertzian Dipole) Equations

Incident Cylindrical Wave

An incident Cylindrical wave is a wave that simulates the far field of an infinite line current placed at the origin.

1. Click **HFSS>Excitations>Assign>Incident Wave>Cylindrical Wave**.
2. Select the **Vector Input Format** as **Cartesian** coordinates.
3. Enter the X-, Y-, and Z-coordinates of the **Excitation Location and/or Zero Phase Position** (the origin for the incident wave).
4. Click **Next**.
5. If you selected **Cartesian**, the **Incident Wave Source: Cartesian Vector Setup** page appears. Enter the X-, Y-, and Z-components for the **I Vector** in the **X**, **Y**, and **Z** boxes. **I** is the current amplitude (peak value). Units are Amps (A).

   A single incident wave will be defined.

Note Only a single incident wave angle can be defined for periodic structures which are defined with master and slave boundaries

6. Click **Next**. the **Incident Wave Source: Cylindrical Wave Options** page appears.
7. Select the **Radius of Surrounding Cylinder**. Inside this cylinder, the field magnitude will be made equal to the field magnitude calculated on the surface of the cylinder. To restore the

default (10 mm), click the **Use Defaults** button.

8. Click **Finish**.The incident wave you defined is added to the **Excitations** list in the **Project**.

Related Topics

Technical Notes: Incident Waves

Technical Notes: Cylindrical Wave Equations

Gaussian Beam Wave

An incident Gaussian Beam wave propagates in one direction and is of Gaussian distribution in the directions perpendicular to its direction of propagation. The steps for assigning a Gaussian Beam Wave are the same as those for assigning Plane Wave. The only difference is that in this case **Gaussian Beam Wave Options** will appear as one of the tabs instead of **Plane Wave Options** on the **Incident Wave Source** dialog box. For more information see Plane Wave.

Related Topics

See the following topics in the Technical Notes.

Incident Waves

Gaussian Beam Equations

Linear Antenna Wave

An incident linear antenna wave is a wave that simulates the far field of a linear antenna placed at the origin. The steps for assigning a Linear Antenna Wave are the same as those for assigning a Hertzian Dipole wave. The only difference is that in this case **Linear Antenna Wave Options** will appear as one of the tabs instead of **Hertzian Dipole Wave Options** on the **Incident Wave Source** dialog box. For more information see Hertzian Dipole Wave.

Related Topics

For more information, see the following topics in Technical Notes.

Incident Waves

Linear Antenna Equations

Far Field Wave

A far field wave originates at a distance several wavelengths from the computational domain. Far field values are defined on the surface of a unit sphere.

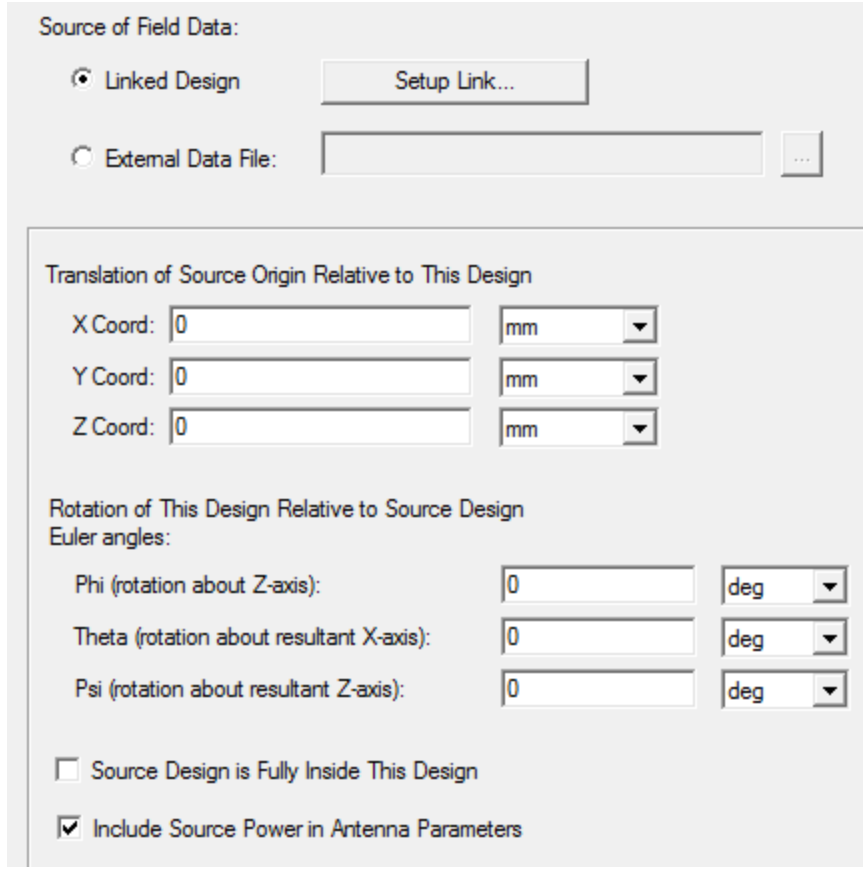
When you use a Far Field link, the origin of the global coordinate system of the source project should be in the phase center of the antenna.

Far Field Wave Options

This section describes the panel **Far Field Wave Options**. Bring up this panel on the **Incident Wave Source** dialog box as follows.

First click **HFSS>Excitations>Assign>Incident Wave >Far Field Wave** and then, click **Next** or the **Far Field Wave Options** tab.

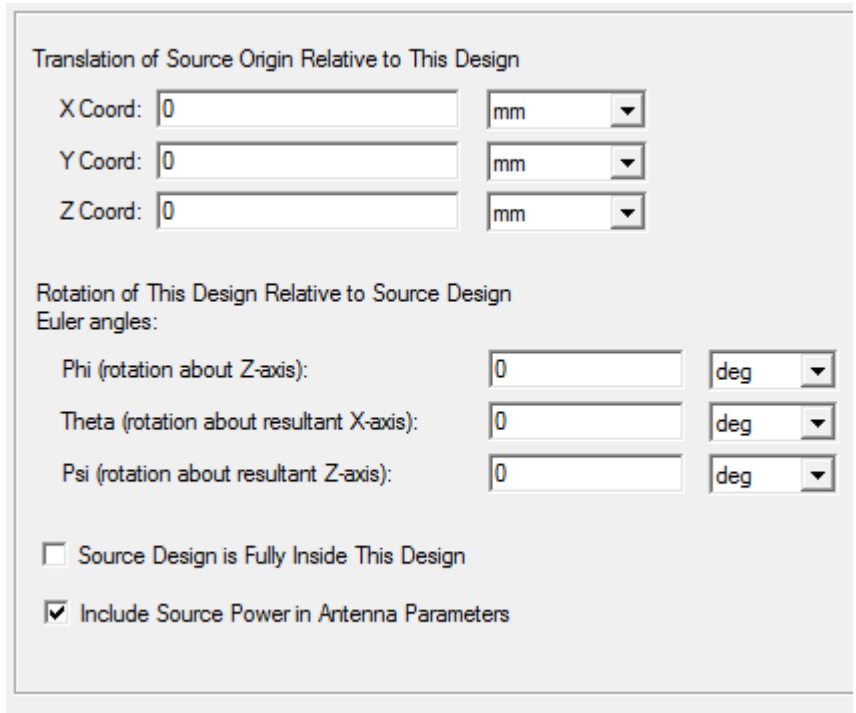
Click the **Setup Link** button on the **Incident Wave Source** dialog to browse for a source project for specifying the linked design.



Translation of Source Origin Relative to This Design

If the coordinate system you are using in the source design is different from that in the target design, you must define the relationship between those coordinate systems as a translation and a rotation. The translation is the offset between the **origins** of the two coordinate systems, and the rotation can be defined through the use of **Euler angles.**

.



The X-, Y-, and Z-coordinates of the source location and/or Zero Phase Position (the origin for the incident wave) represents the translation of the source design's origin with respect to the target design's origin. For instance, if the source design's origin is located in the target design co-ordinate system at (-2, -2, 1), then the translation between the two coordinate systems is (-2, -2, 1).

The Euler angles represent the angles through which you can rotate the target design to achieve the desired rotation angles.

Note: A figure that represents the angles and the rotation of a design is at the end of the section.

When a source project is inside of the target project, it should not be included into the near/far field calculation. If it is outside the target project, the far field calculation should integrate over the radiation surfaces of the source project as well.

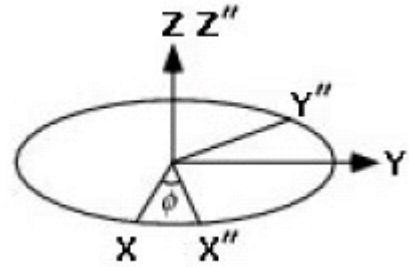
The antenna parameters include power-related entries such as incident power, accepted power, radiated power. If you check **Include Source Power in Antenna Parameters** the antenna parameters in the design that receives the fields from a source design will be based on the power set for the

excitations in the source design under **HFSS>Fields** >Edit Sources. For instance, this can be the power with which a port in a source antenna is excited. If the antenna in a source design is excited with 10 W, then the antenna parameters panel in the receiving design will show an incident power of 10 W.

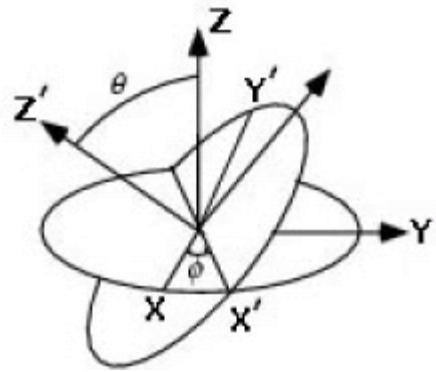
See Computing Antenna Parameters for a description of how to obtain antenna parameters during post processing.

X,Y, Z Source Coordinate System

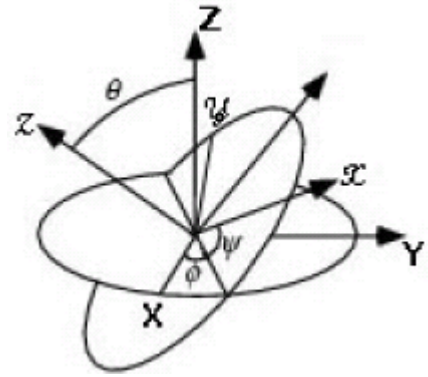
X", Y", Z" Source Coordinate System After Rotation by ϕ [phi]



X', Y', Z' Source Coordinate System After Rotation by θ [theta]



X, *Y*, *Z* Source Coordinate System After Rotation by ψ [psi]



Related Topics

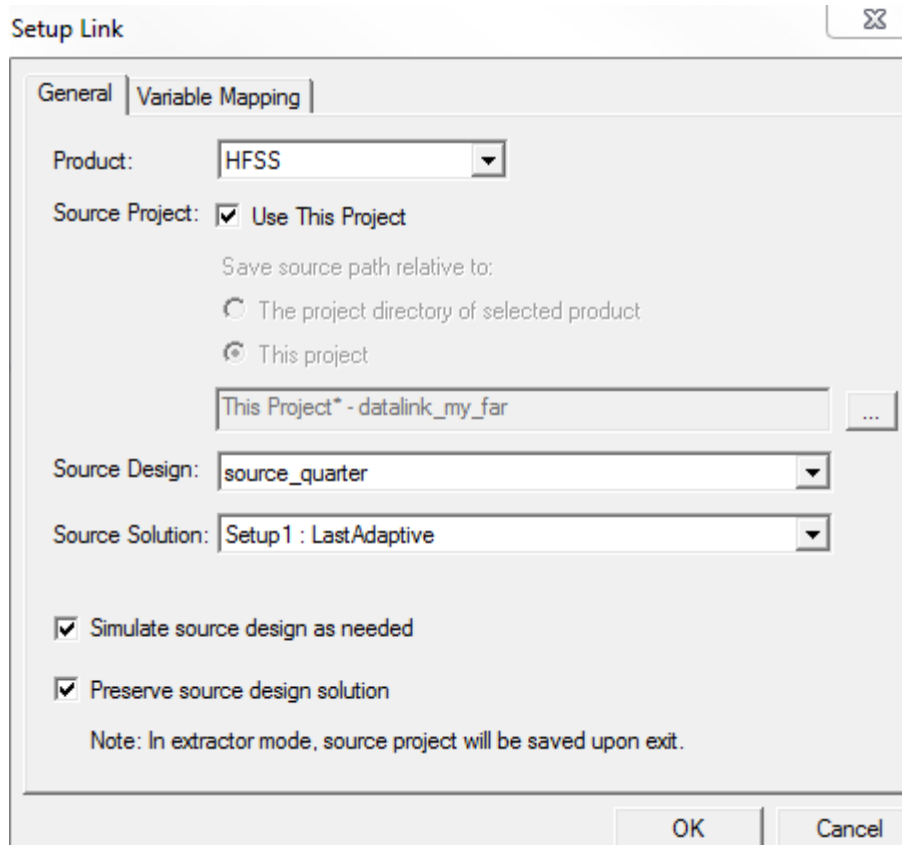
Technical Notes: Incident Waves

Clear Linked Data

Using Field Solutions from Other Simulators

Setup Link for Far Field or Near Field Wave

You can access the **Setup Link** dialog from the Far Field Wave options dialog or Near Field Wave options.



1. Select the **Edit Link** radio button to activate the options in the panel.
2. Select the type of the source project e.g. **HFSS, SIWave or Maxwell** from the **Product** drop-down menu.

   Note: An **HFSS-IE** project can be used as target or source. **Maxwell** does not appear in the drop down menu because it is not applicable for far fields.
3. Browse for the source project or check **Use This Project**. If you check the latter option**,** you must select a source design from the current project.

Note: The origin of the global coordinate system of the source project should be in the phase center of the antenna, when you use a Far Field link.

4. Specify the **Source Solution**. The first option in the Source Solution field is the default.

   Note: The solution in the source design must provide data for the target design's adaptive frequency as well as its sweeps. That is, the adaptive frequency for the target design must be included in the sweep in the source design. If necessary, you can open the source design and add an appropriate frequency point to an existing sweep.

5. Use the last two checkboxes at the bottom of the panel as needed.

   Note: Extractor mode means that the software is opened during the link solely for the purpose of solving.

6. Under the **Variable Mapping** tab, you can set the desired variable values in the source design. If the source and target designs contain same named variables, you can choose to Map Variable By Name. In this case, same named variables are mapped automatically.

External Data File for Far Field Wave

You can define a Far Field **Incident Wave Source** as a plain text data file with a .ffd suffix. The fields can be independent or dependent on the frequency.

Note: The far field data file must be a plain text file. Wordpad includes invisible characters that HFSS does not support. The code tolerates white space separators anywhere (blank, tab, newline, carriage return). Between double values in same logical grouping, it tolerates comma or semicolon separators.

Frequency Independent Far Fields

In this case, the file format is simple. No keywords are used. The first two lines contain doubles representing start, stop, and num points for the sweeps of theta and phi. The angle units are degrees. In order to have a unique coordinate system theta and phi should be in the range of $0 < \text{theta} < 180$ and $0 < \text{phi} < 360$.

```
ThetaStart ThetaStop ThetaNumPoints
PhiStart PhiStop PhiNumPoints
```

Then, the subsequent lines give the complex components of electric field in the theta and phi directions. The unit of E field is Volt. All of these are doubles. For these lines, theta will be held constant while phi is swept through all values, then theta will increment to the next value, etc. So the lines will correspond to (th1, ph1), (th1, ph2), (th1, ph3),…,(th1, phN), (th2, ph1), (th2, ph2), etc… Note that this format enforces a strict uniform grid of theta and phi samples.

```
E_theta_real   E_theta_imag   E_phi_real   E_phi_imag
E_theta_real   E_theta_imag   E_phi_real   E_phi_imag
...
```

See the following example of a data file:

```
0,180,3
0,360,5
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
```

Note: Instead of commas you can use double-space or semicolon separators.

Frequency Dependent Far Fields

For **frequency-dependent** far field links, the data is supplied in blocks. The syntax for a frequency dependent far field uses the following format:

```
ThetaStart ThetaStop ThetaNumPoints
PhiStart PhiStop PhiNumPoints
Frequencies   NumFrequencies
Frequency   FrequencyValue
E_theta_real   E_theta_imag   E_phi_real   E_phi_imag
E_theta_real   E_theta_imag   E_phi_real   E_phi_imag
E_theta_real   E_theta_imag   E_phi_real   E_phi_imag
```

… repeat for all theta and phi sweep points

```
Frequency   FrequencyValue
E_theta_real   E_theta_imag   E_phi_real   E_phi_imag
E_theta_real   E_theta_imag   E_phi_real   E_phi_imag
E_theta_real   E_theta_imag   E_phi_real   E_phi_imag
```

… repeat for all theta and phi sweep points
… repeat for a total of NumFrequencies

Data File Example

See the sample data file below.

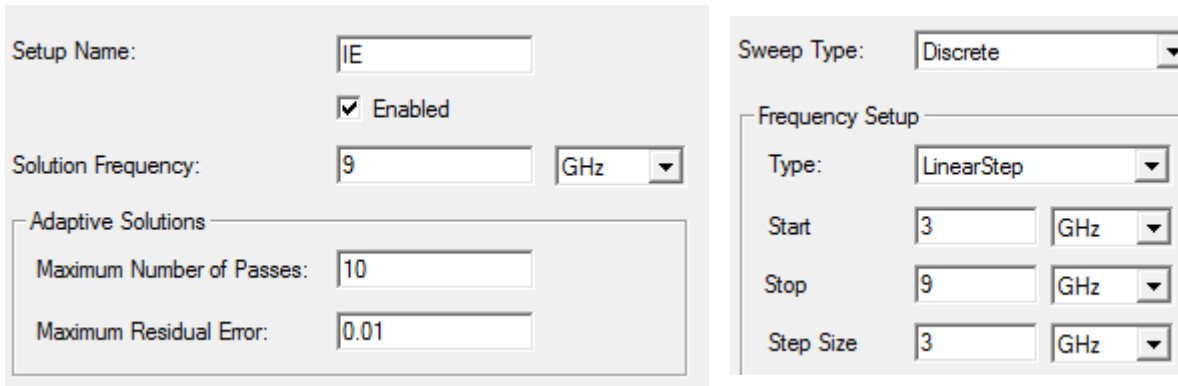
```
0,180,2
0,360,5
frequencies 3
frequency 3.0e9
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0
0.0,0.0,0.0,1.0

frequency 6.0e9
0.0,0.0,0.0,2.0
0.0,0.0,0.0,2.0
0.0,0.0,0.0,2.0
0.0,0.0,0.0,2.0
0.0,0.0,0.0,2.0
0.0,0.0,0.0,2.0
0.0,0.0,0.0,2.0
0.0,0.0,0.0,2.0
0.0,0.0,0.0,2.0
0.0,0.0,0.0,2.0

frequency 9.0e9
0.0,0.0,0.0,3.0
```

```
0.0,0.0,0.0,3.0
0.0,0.0,0.0,3.0
0.0,0.0,0.0,3.0
0.0,0.0,0.0,3.0
0.0,0.0,0.0,3.0
0.0,0.0,0.0,3.0
0.0,0.0,0.0,3.0
0.0,0.0,0.0,3.0
0.0,0.0,0.0,3.0
```

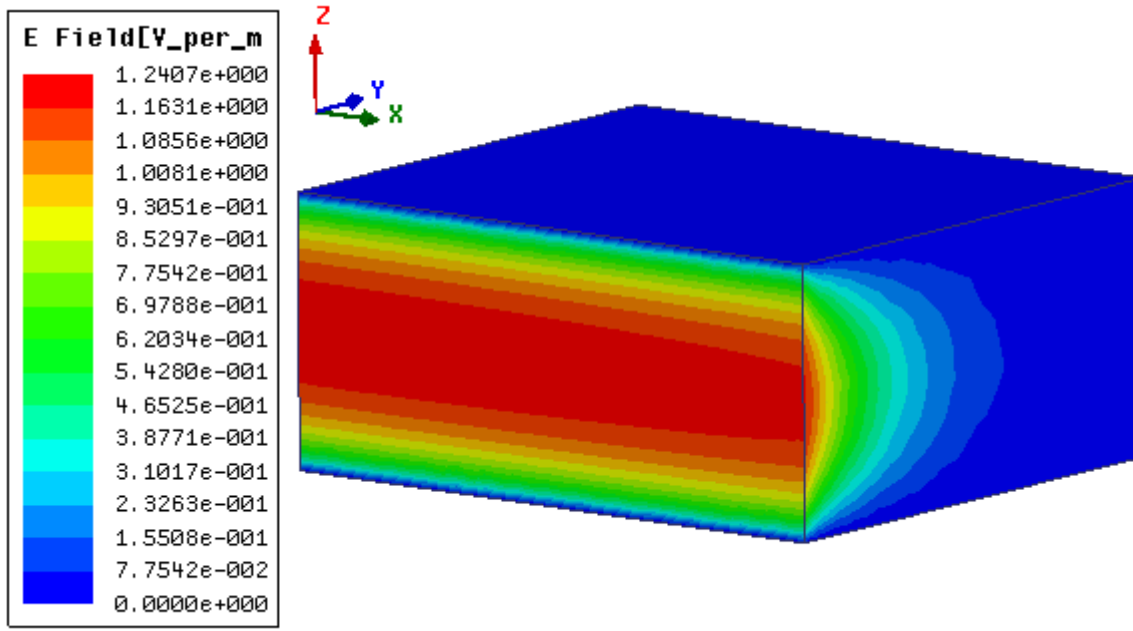
The syntax and their values are as follows:

- 0, 180, and 2 are the values of `ThetaStart`, `ThetaStop`, and `ThetaNumPoints` in the first line.
- The values 0, 360, and 5 do the same thing for `Phi`.
- `NumFrequencies` represents the number of blocks defined in the data file. In this example, 3 frequency blocks are defined.
- `FrequencyValue` represents the value of frequency for which you supply the far field data. In the Data File Example, `FrequencyValue` is as follows:
  - 3.0e9 Hz for the first block;
  - 6.0e9 Hz for the second block;
  - 9.0e9 Hz for the third block.

HFSS supports only the discrete sweep type for Far Field (and Near Field) data links. The frequencies defined in the sweep must be consistent with the frequencies specified in the data file. If not, an error message will pop up. So make sure that all the frequencies that you supply in the data file are consistent with those in the **Frequency Setup** section of the **Edit Frequency Sweep** dialog box. For example, based on the **Frequency Setup** shown below in the target project HFSS will expect to find data for 3GHz, 6 GHz, and 9 GHz in the data file. Also, the **Solution Frequency** in the **Solution Setup** should match one of the frequencies in the data file and the sweep if the project type is HFSS or HFSS-IE in the target design. Also, the frequency blocks can appear in any order in the data file.



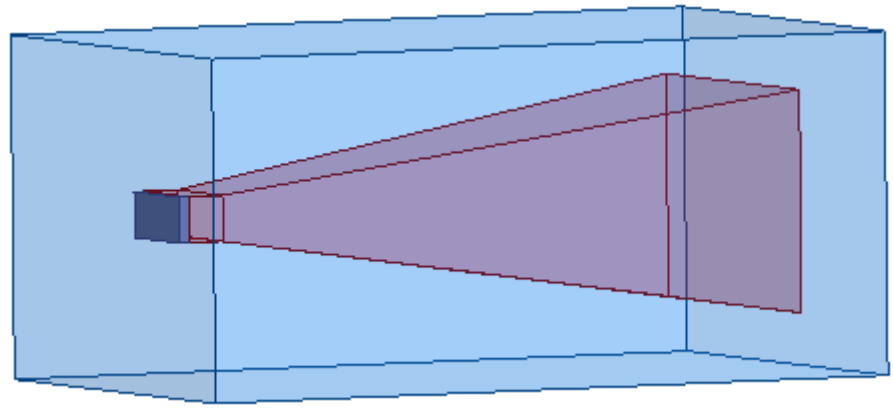
The sample frequency dependent data file was used to apply far field on an open ended WR-90 waveguide. The frequency sweep shown was used to simulate it. The results for the E-field at 9 GHz are shown below.



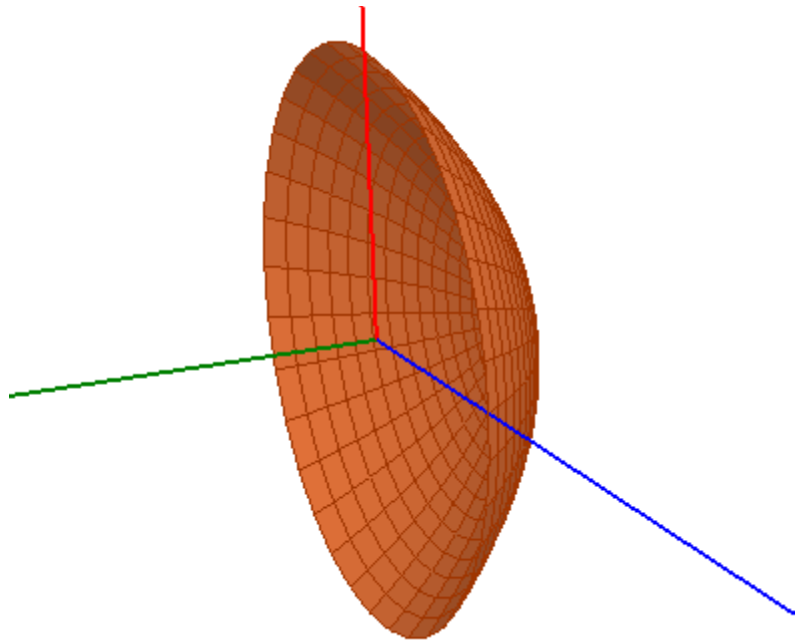
Example That Uses Far Field Links

This section describes linking an HFSS project to an HFSS-IE project while using the common feature of assigning Far Field Wave excitation.

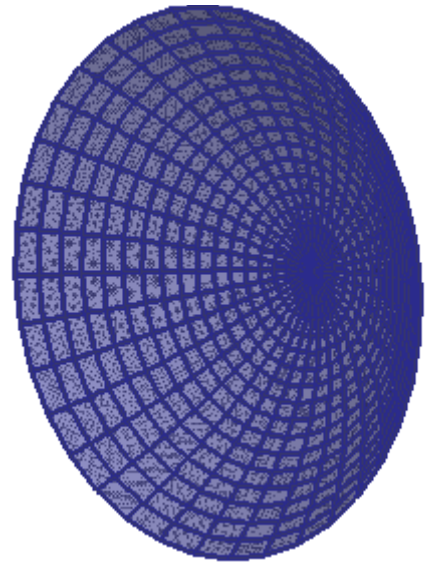
The horn antenna enclosed in an air box in the figure below represents the source project designed in HFSS.



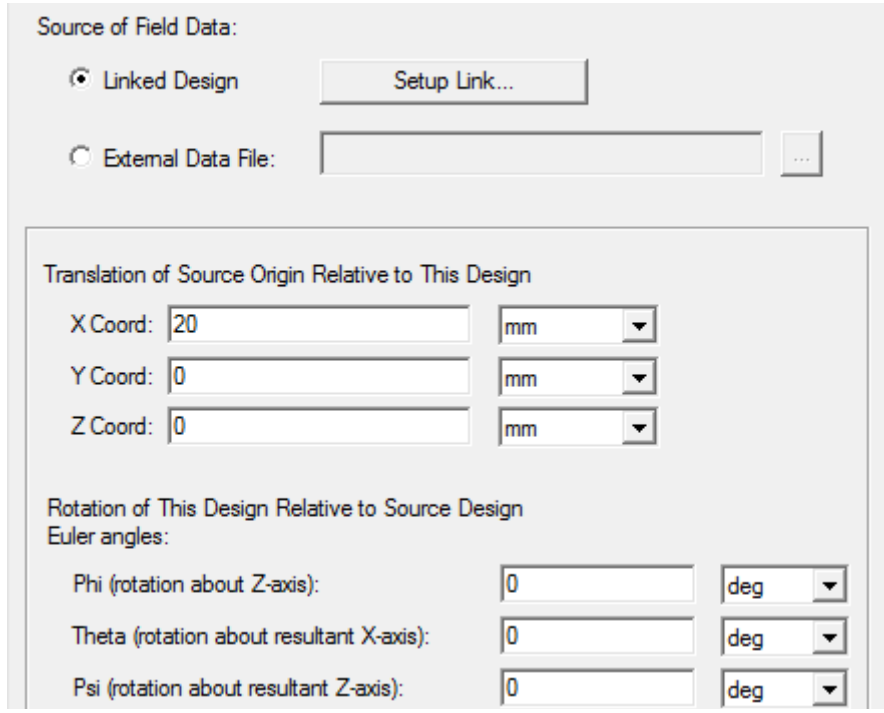
The dish/reflector in the figure below is the target project designed in HFSS-IE.



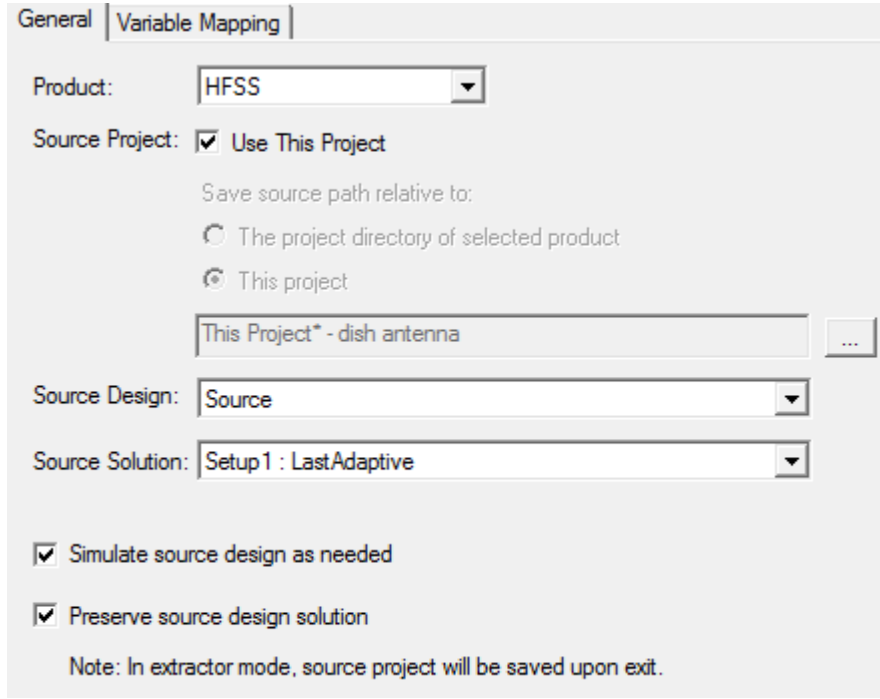
A perfect E-boundary is assigned on the reflector.



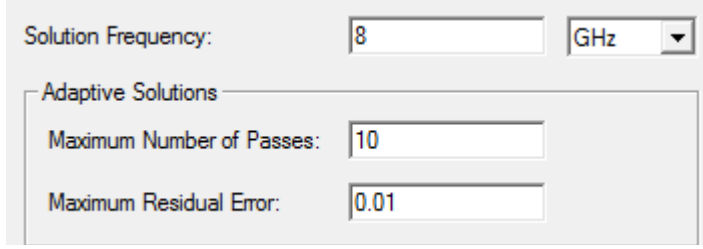
The **Far Field Wave Options** panel has the following settings.



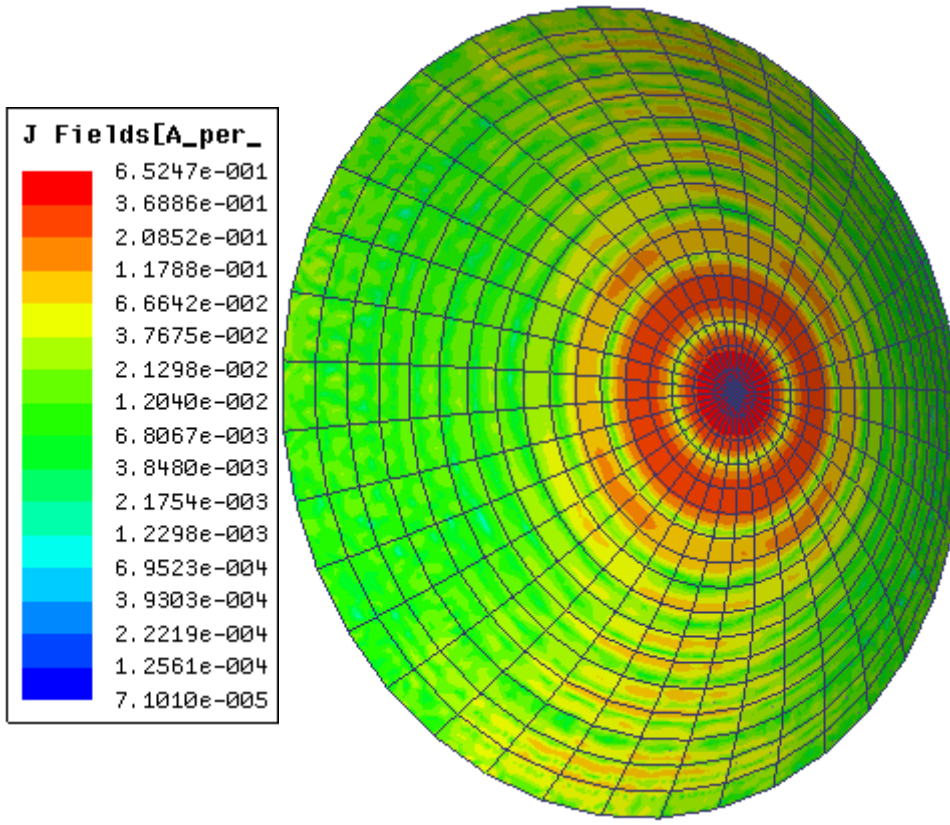
From the **Setup Link** option, the fields are set as shown in the figure to link the target to the source project (the horn antenna).



A sample solution setup for this project is shown below:



The resulting J Fields (magnetic fields) after solving the project are shown below.

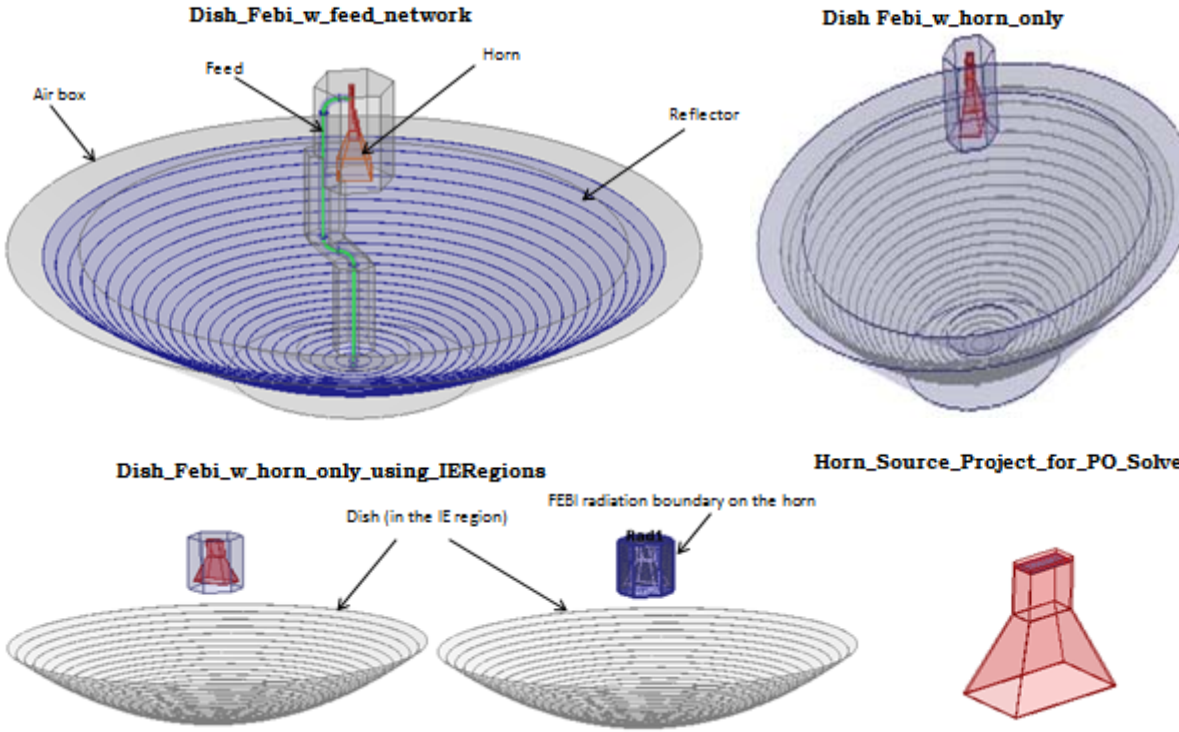


Near Field Wave

A Near Field wave source is close enough to the design for near field effects to occur, typically within one wave length. Near field waves tend to have both evanescent and propagating content.

Assign Near Field Wave

This section describes the **Near Field Wave Options** panel on the **Incident Wave Source** dialog box and how to assign near field waves. The following designs illustrate different ways of using near field sourcing to run a range of analyses on a dish antenna fed by a circular horn antenna.



First we will bring up the **Near Field Wave Options** panel and then, assign Near Field Wave.

1. Click **HFSS>Excitations>Assign>Incident Wave>Near Field Wave**.

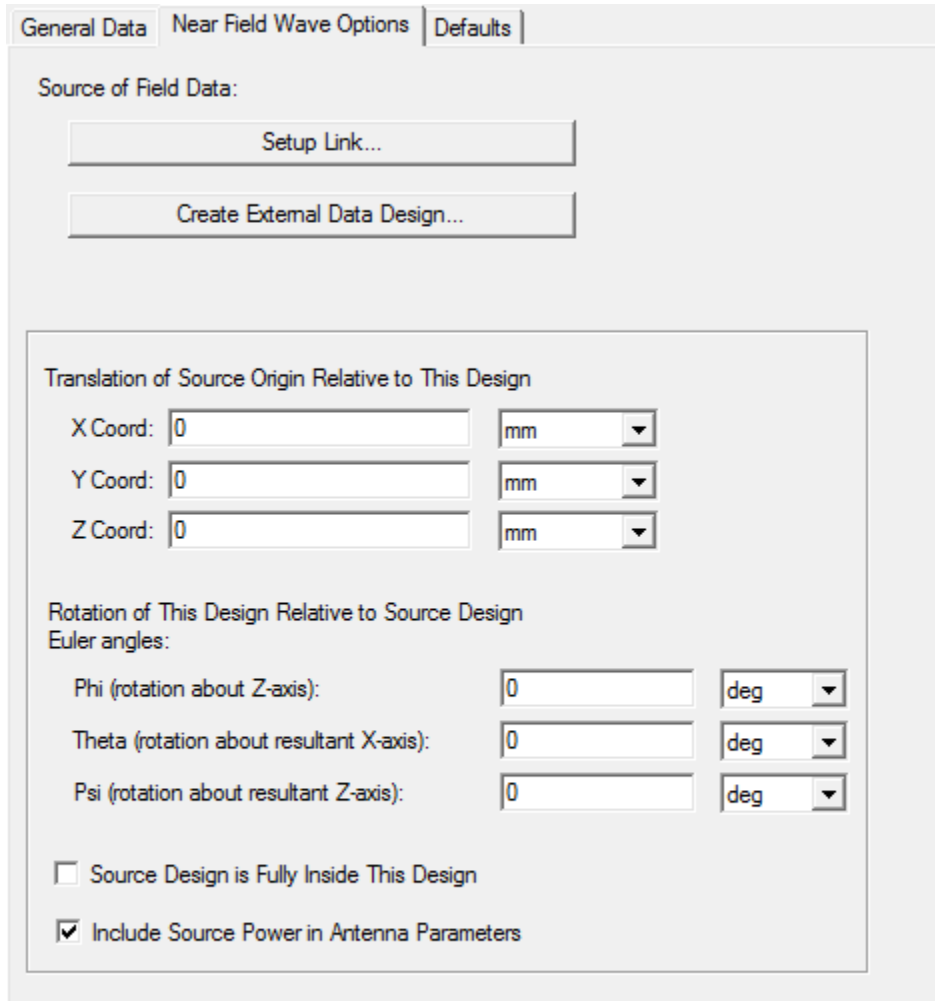
   Note: If you are using HFSS-IE go to **HFSS-IE>Excitations>Assign>Near Field Wave**

   The **Incident Wave Source: General Data** dialog box appears.

2. Fill in the **Excitation** name and click **Next** to access the **Near Field Wave Options** panel.

   Note: For array problems, the **Active** option includes the effects of the Near Field in the simulation. To disable the effects, uncheck the **Active** option.

3. Determine whether you want to use a linked design or an external data file. If you are using a linked design go to step 4. If you are using an external data file to use to **Create External Data Design...**, go to Access Near Field External Data File and follow the instructions.

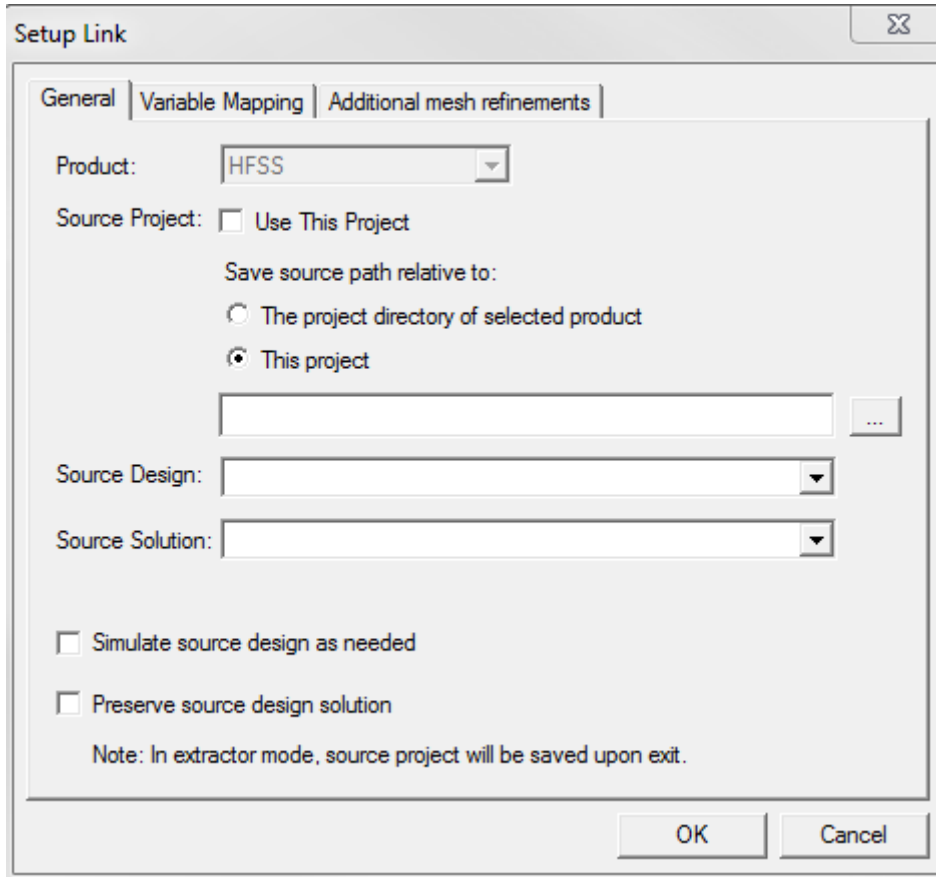


4. Click the **Setup Link** button on the **Incident Wave Source** dialog to specify the linked design.

   Note: It is through the **Setup Link** dialog box that you will link the source design to your target design.

5. On the **Setup Link** dialog box select the radio button **Edit Link**.

   The fields in the dialog box become active and editable as shown below.

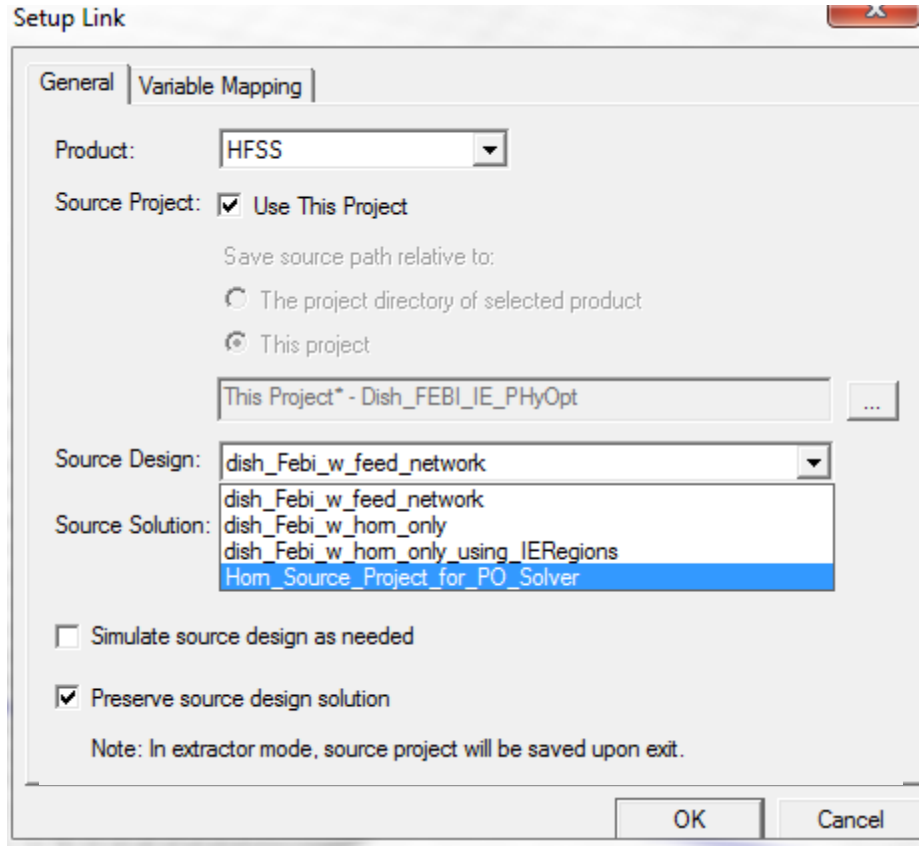


6. Select the product in which you created your source design from the **Product** drop down list.

   Note: Only HFSS, Maxwell, and SIWave appear in this list. An **HFSS-IE** project can be used as target or source.

7. Specify the project that has the source design in one of the following ways:
   a. Check **Use This Project** to pick a design from a current working project.
   b. Click the ellipsis and browse your computer to pick the design.

Note The option **The project directory of selected product** points to the default project directory location relative to the product that you selected from the drop-down menu. This is the same project directory that you set on the **General Options** dialog under **Tools**> **Options**. This option is useful especially, when you link across different products. The option **This project** points to the directory relative to where your current working project is located.

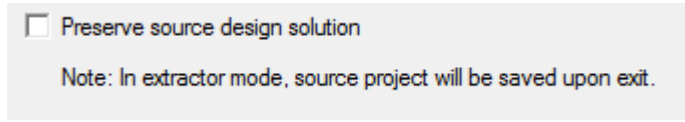


8. Pick the source design from the **Source Design** drop-down menu which lists all the designs in the specified project.
9. Pick the setup type from the **Source Solution** pull-down menu (e.g. Last Adaptive, or Sweep, etc.).
10. Select the last two check boxes as needed and click **OK**.

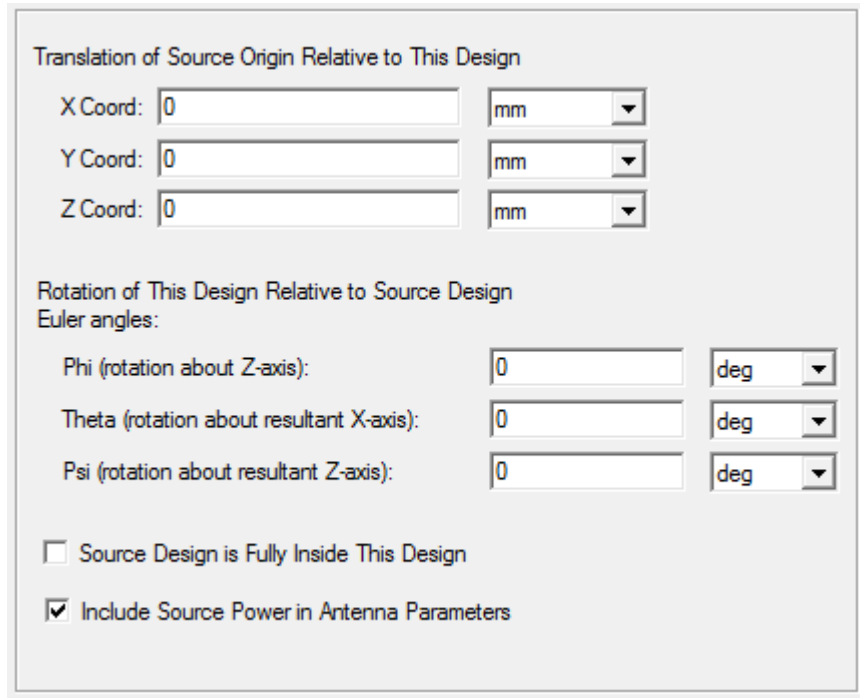
Note Go to Simulate source design as needed, for more information about that option.

Preserve source design solution

This is the last option on the Setup Link dialog box. When the source design is closed or it is not included in the project manager window containing the target design, the source design will not be saved. However, if you select this option, which is useful when you are using Maxwell or SIWave the results of the source design is saved.



11. Close the Setup Link dialog, and if necessary, specify the **Translation of Source Origin Relative to This Design** on the **Near Field Wave Options** panel of the **Incident Wave Source** dialog box.



12. If required check the option **Source Design is Fully Inside This Design**.

    When a source design is totally inside the target design, it should not be included into the near/far field calculation. If it is totally outside, the far field calculation should also integrate over the radiation surfaces of the source project.

13. If required check **Include Source Power in Antenna Parameters**.

    The antenna parameters include power-related entries such as incident power, accepted power, radiated power. If you check **Include Source Power in Antenna Parameters** the antenna parameters in the design that receives the fields from a source design will be based on a power set for the excitations in the source design under **HFSS>Fields >Edit Sources**. For instance,

this can be the power with which a port in a source antenna is excited. If the antenna in a source design is excited with 10 W, then the antenna parameters panel in the receiving design will show an incident power of 10 W.

Note For more information on how to obtain antenna parameters during post processing, see Computing Antenna Parameters.

14. Click **Finish** to close the dialog. The **Near Field** wave source point and direction is highlighted in the modeler window, and the wave appears in the **Excitations** list in the **Project**.

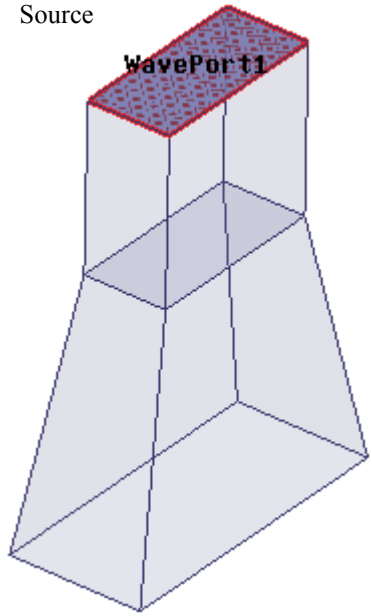
Simulate source design as needed

Select this option if you modify the source design and you want an updated source design solution to be linked to the target design. If you select this option, make sure that you invalidate the cached link data in the target. To do this, right click **Analysis** and select the option **Clear Linked Data** from the short-cut menu.

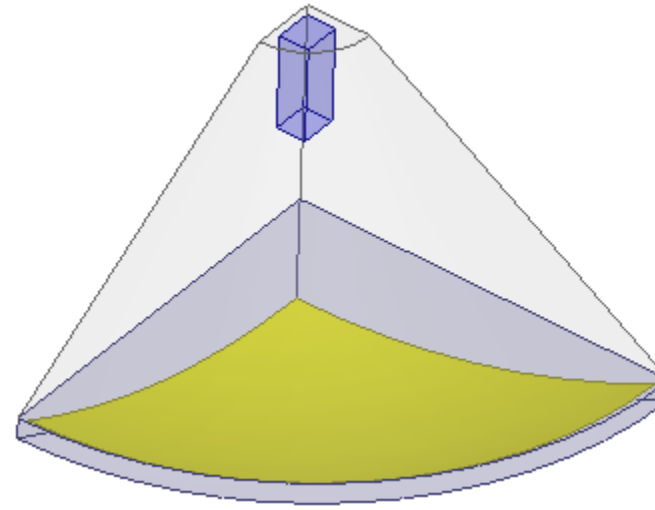
Example that Uses Near Field Links

The figure below shows a horn antenna that will be linked to a target disk where the source will be included inside of the target.

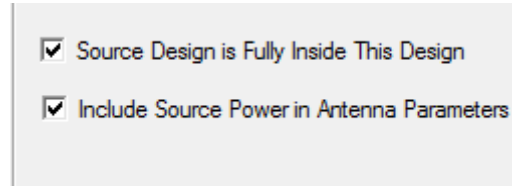
Source



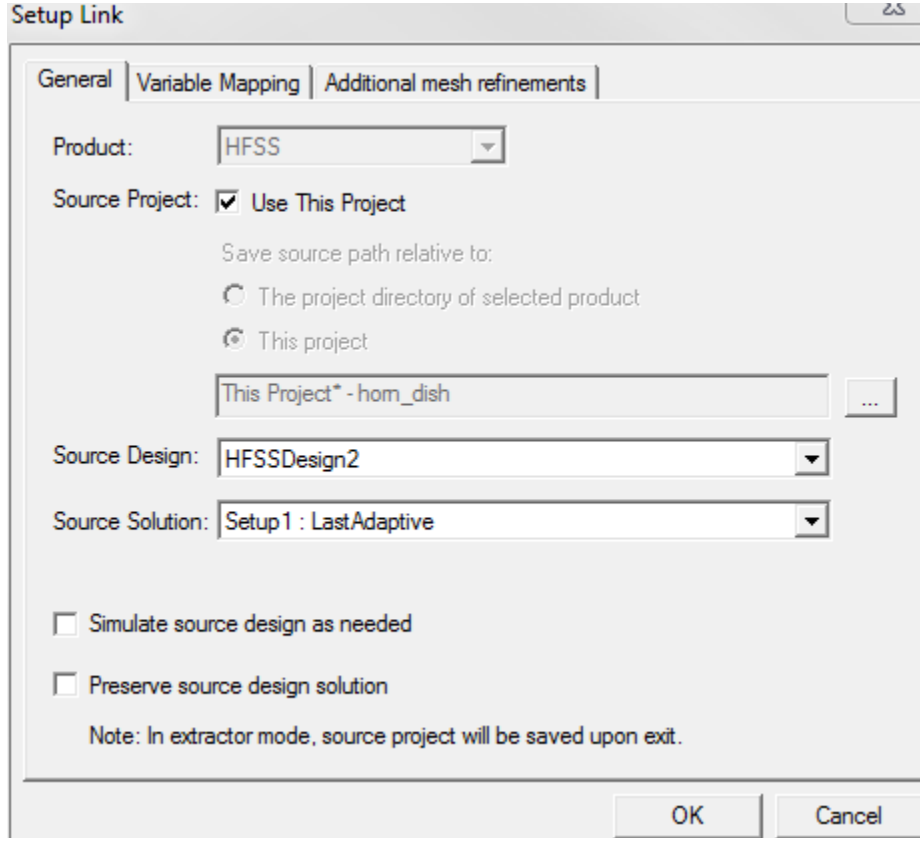
Target



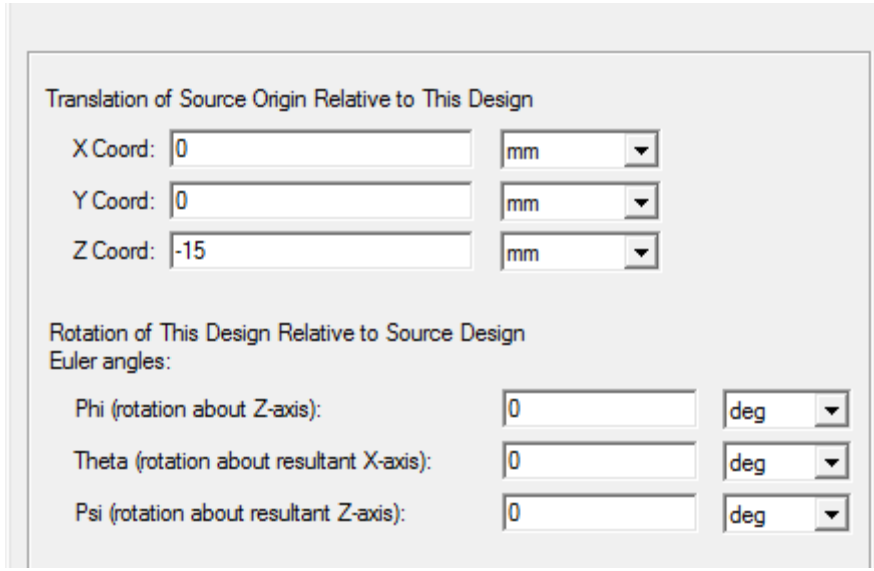
On the **Incident Wave Source: Near Field Wave Options** dialog box, the options shown in the figure below are checked owing to which the source is included in the target design.



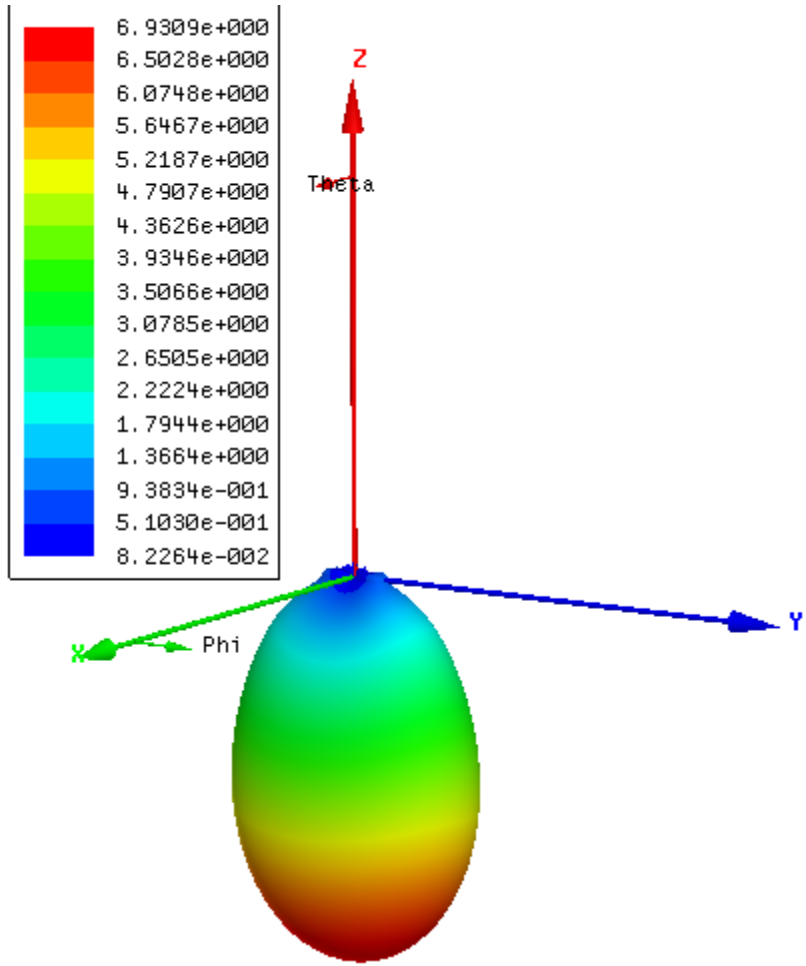
The Near Field Setup Link settings are as shown in the figure below.



In the example of the horn antenna linking to a target dish, the co-ordinate systems of the source and the target coincide. You can use the **X Coord**, **Y Coord**, and **Z Coord** fields to set the source co-ordinate system. For instance if you want to move the source and its co-ordinate system closer to the dish by 15 mm you can enter -15 mm in the **Z Coord** field as shown below.

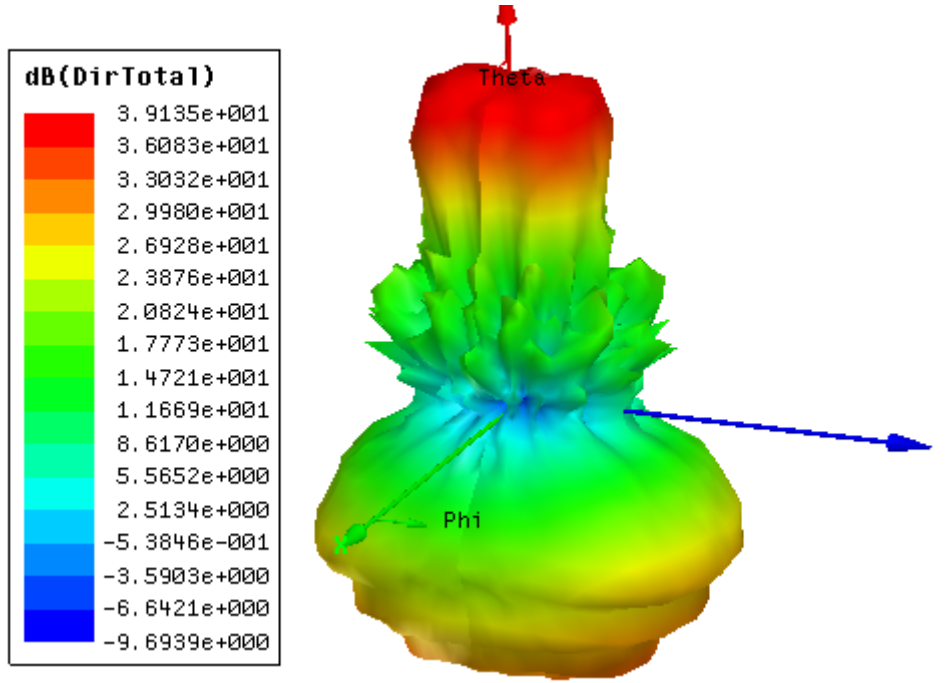


The 3D polar plot with the values of directivity for the source project (the standalone horn antenna without the dish) is shown below.



Then, when we apply the fields to the Target project with the dish, we get the values of the directivity shown in the figure below.

Note: The standalone horn antenna without the dish radiates in the negative Z direction while the target project which includes the dish antenna, radiates in the positive Z-direction. The 3D polar plot with the values of directivity after the application of the Near Field link is shown in the figure below.



Access Near Field External Data File

In this section we will show how to access a source of near field which is imposed on a target design through a data file. Instead of using a **Linked Design** as a source for a near field you can also use data files to impose the near field upon your target project.

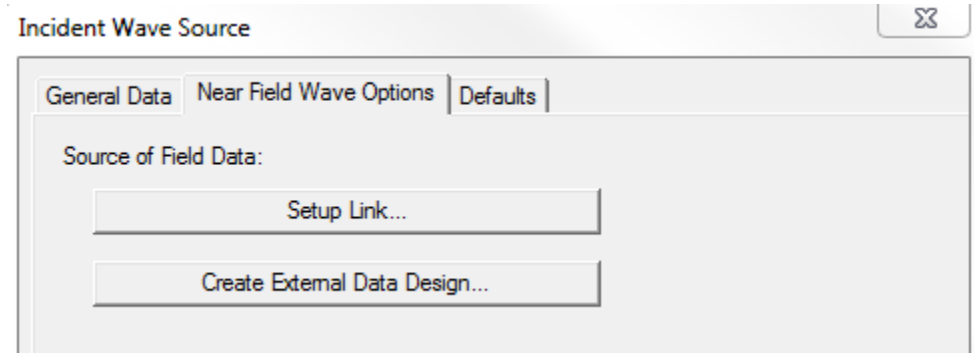
Note For more information about Ansys Near Field Data File (**.and*) format see the ANSYS Near Field Data File section.

Presupposing that you have prepared the *.and file and associated near field data files:

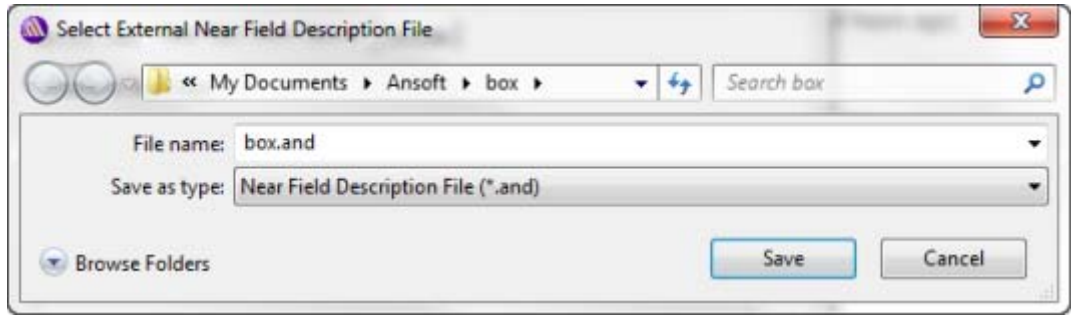
1. Go to **HFSS**>**Excitations**>**Assign**>**Incident Wave**>**Near Field Wave**.

   The **Incident Wave Source** : **General Data** dialog box appears.

2. Click **Next** to access the near field wave options.

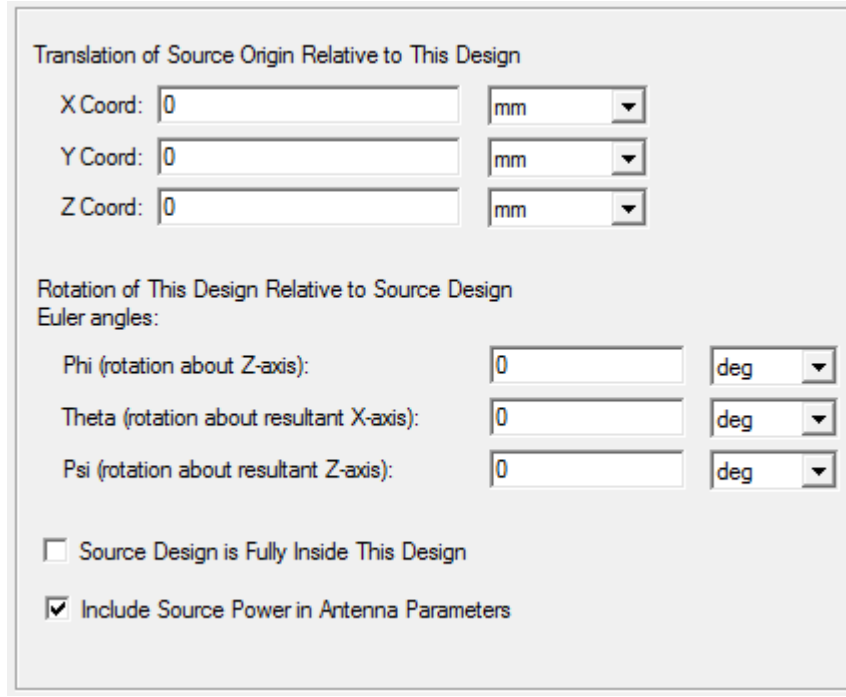


3. Select the **Create External Data Design** button and browse your computer to access a previously created **.and* file, which describes the name and location of one or more the near field data files (*.nfd).



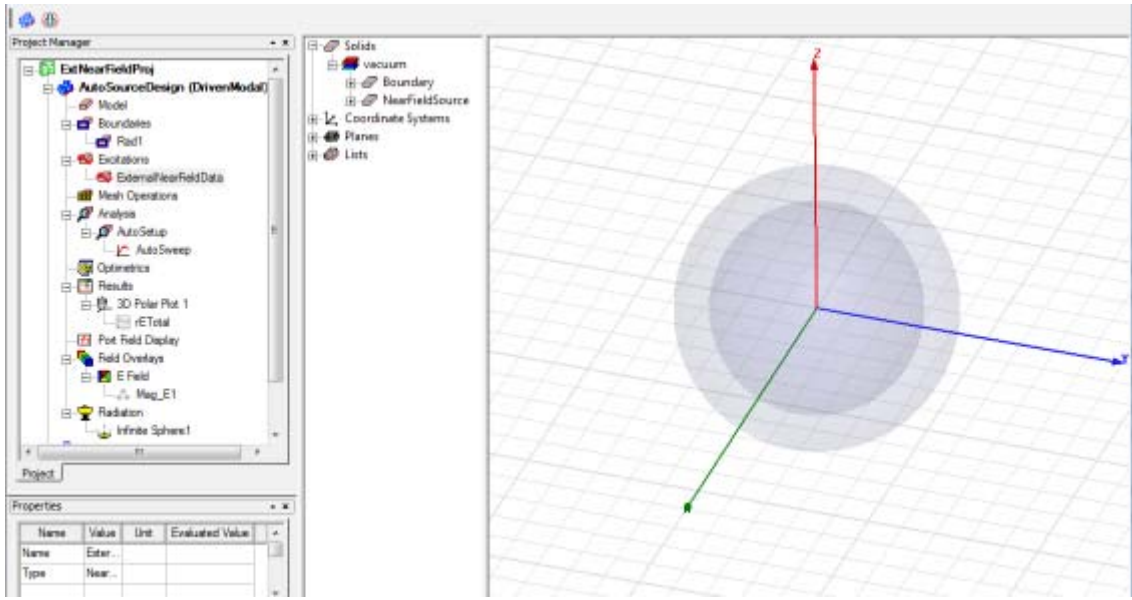
4. If necessary, specify the **Translation of Source Origin Relative to This Design** on the **Near**

Field Wave Options panel of the **Incident Wave Source** dialog box.



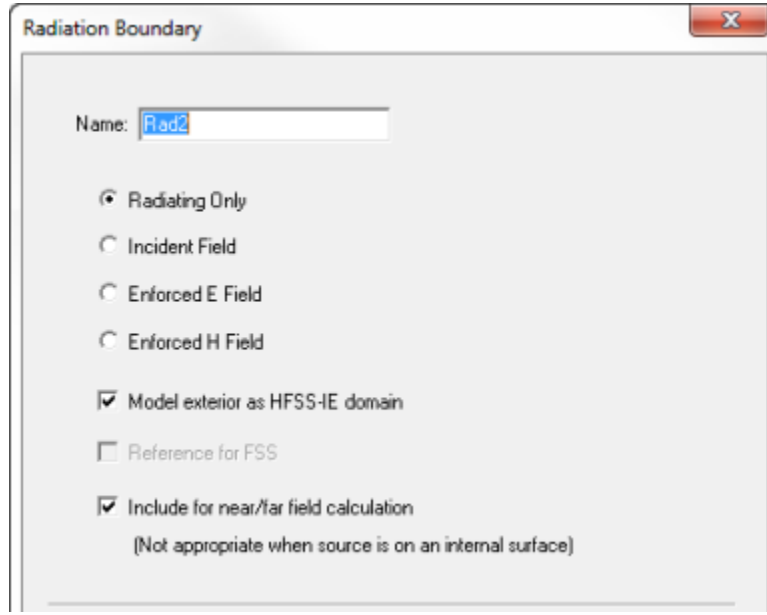
Upon a successful source design generation, the datalink setup for the near field incident wave

source will be automatically set to the generated source design.



This auto-design generation is neither undo-able nor scriptable. This design becomes the source design for external near field. This work flow allows you to view their external field data via HFSS.

It supports the choices of E field only and H field only external data by specifying the radiation boundary properties.



Automatically generated design components include:

- Geometry

  The external data geometry and a region. The padding of the region is 1/5 of lambda. Lambda is based on the highest frequency in the external data.

- Boundaries

  An "incident" wave radiation boundary on the external data geometry

  a. "Incident Field" selected for E & H external data

  b. "Enforce E Field" selected for E only data

  c. "Enforce H Field" selected for H only data

  d. J & M (see case a)

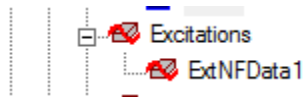
  e. J or M (see case b and c)

  By default, a radiating radiation boundary is assigned on the region. However, you can replace this boundary with a PML.

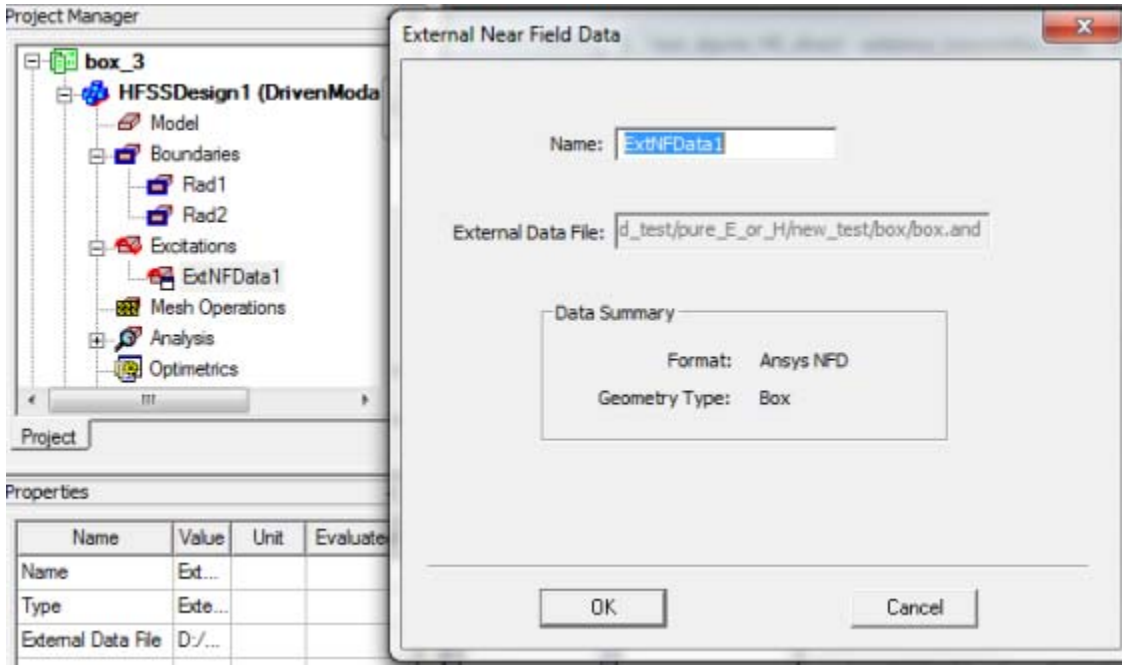
- Excitations

  An External Near Field Data Excitation. This is an excitation type for user defined external

near field.



Accessing the **External Near Field Data Excitation** dialog can only be done through the Auto Design under Excitations in Project tree. Other than the Name, no other properties this Excitation are editable through this dialog. It exists to provide you with information about the AND File that was used to create this Auto-Design.



- Analysis

  A default solve setup is created with the adaptive frequency set to the highest frequency in the external data

  A discrete frequency sweep is created if multiple frequencies are present in the External Data.

5. When you run the Analysis of your target design, it uses this newly created source design for the Near Fields.

ANSYS Near Field Data File

The data file that you select when you click the **Create External Design** button of the **Incident Wave Source** dialog box is in the **.and* format. The **.and* file describes the location and contents of one or more Ansys Near Field Data Files which are in the **.nfd* format. A example of a **.and* file is shown below.

Sample *.*and* File

```
$begin 'NearFieldHeader'
   type = 'nfd'
   geometry = 'sphere'
   center = '0mm, 0mm, 0mm'
   radius = '20mm'
   fsweep='1GHz, 10GHz'
$end 'NearFieldHeader'

$begin 'NearFieldData'
   FreqData("1GHz", "exportfields-1GHz.nfd")
   FreqData("2GHz", "exportfields-2GHz.nfd")
   FreqData("3GHz", "exportfields-3GHz.nfd")
   FreqData("4GHz", "exportfields-4GHz.nfd")
   FreqData("5GHz", "exportfields-5GHz.nfd")
   FreqData("6GHz", "exportfields-6GHz.nfd")
   FreqData("7GHz", "exportfields-7GHz.nfd")
   FreqData("8GHz", "exportfields-8GHz.nfd")
   FreqData("9GHz", "exportfields-9GHz.nfd")
   FreqData("10GHz", "exportfields-10GHz.nfd")
$end 'NearFieldData'
```

This example is a simple text file and saved with the *.and* extension. It is clear from the sample that the near field data was generated on a sphere centered at the origin and radius = 20mm for frequencies 1 GHz through 10 GHz. For the sample **.and* file 10 **.nfd* files are specified so 10 **.nfd* files were generated. Examples of such Ansys Near Field Data Files (*exportfields-1GHz.nfd*) are shown below.

Users can also use external data in the field format of J and M, J only and M only.

Related Topics

Technical Notes: ANSYS Near Field Description Format Specification

Sample NFD File in Cartesian Coordinate System

The sample below shows the near field data that the *.nfd file contains. The file uniquely specifies the position of several points (700 in this case) and the real and imaginary values of the E field and H field vectors at these points along the x, y, and z axes in the Cartesian Coordinate system.

```
#Index, X, Y, Z, Ex(real, imag), Ey(real, imag), Ez(real,
imag), Hx(real, imag), Hy(real, imag), Hz(real, imag)
Frequencies 5
```

```
Frequency 1.000000e+009
1, 0, 0, 0.02, 0.0938, -0.1140, -0.0136, -0.0132, -0.6038,
-0.4006, 0.0004, -0.0004, -0.0006, -0.0008, -0.0037, 0.0038
....... ....... ....... ....... ....... .......
....... ....... ....... ....... ....... .......
700, 2.1e-018, -1.2e-018, -0.02, 0.0839, -0.1080, -0.0125,
-0.012, -0.0213, 0.1423, -0.0002, 0.0002, 0.0003, 0.0009,
-0.003, 0.0032

Frequency 5.000000e+009
....... ....... ....... ....... ....... .......
....... ....... ....... ....... ....... .......
```

Sample NFD File in Spherical Coordinate System

The sample below shows the near field data that the *.nfd file contains. The file uniquely specifies the position of several points (700 in this case) and the real and imaginary values of the E field and H field vectors at these points along the r, theta, and phi directions in the Spherical Coordinate system.

```
#Index, Theta, Phi, Er(real, imag), Etheta(real, imag),
Ephi(real, imag), Hr(real, imag), Htheta(real, imag),
Hphi(real, imag)
Frequencies 5
Frequency 1.000000e+009

1, 0.000000e+000, 0.000000e+000, -1.835144e-001, -1.822055e-
001, -2.648018e-002, -2.096681e-001, -1.281892e-001, -
5.361222e-002, 1.279651e-004, -1.016122e-004, 6.289402e-004, -
6.598072e-004, -1.281892e-001, -5.361222e-002
....... ....... ....... ....... ....... .......
....... ....... ....... ....... ....... .......

700, 3.141593e+000, 5.759587e+000, -1.282983e-001, -1.141968e-
001, -2.207641e-001, -9.119582e-002, -2.669357e-001, -
1.063323e-001, 6.827586e-005, -9.142062e-005, 1.030250e-003, -
1.406250e-003, -2.669357e-001, -1.063323e-001

Frequency 5.000000e+009
....... ....... ....... ....... ....... .......
```

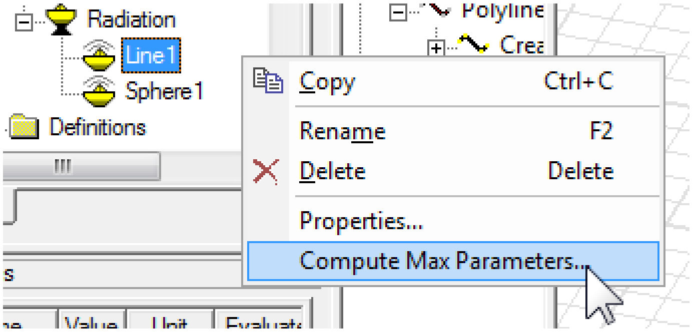
. . . .

. . . .

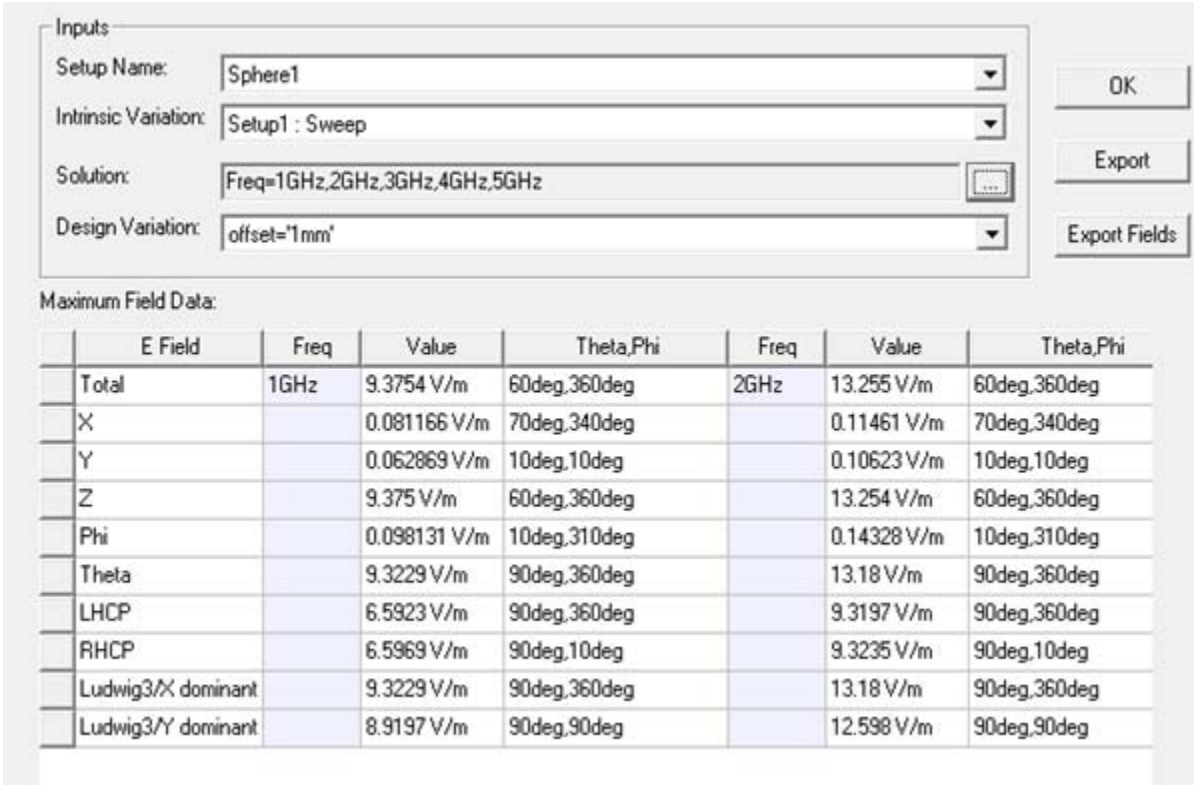
Generate Near Field Data File

Generate the ***.nfd** file from your existing data or from solved HFSS design with near fields using the following steps.

1. Right click the previously created near-field **Sphere** or **Line1** under **Radiation** as shown below.



2. Select **Compute Max Parameters**.
3. The **Max Field Data** dialog box appears for the selected Line or Sphere as shown below.



E Field	Freq	Value	Theta,Phi	Freq	Value	Theta,Phi
Total	1GHz	9.3754 V/m	60deg,360deg	2GHz	13.255 V/m	60deg,360deg
X		0.081166 V/m	70deg,340deg		0.11461 V/m	70deg,340deg
Y		0.062869 V/m	10deg,10deg		0.10623 V/m	10deg,10deg
Z		9.375 V/m	60deg,360deg		13.254 V/m	60deg,360deg
Phi		0.098131 V/m	10deg,310deg		0.14328 V/m	10deg,310deg
Theta		9.3229 V/m	90deg,360deg		13.18 V/m	90deg,360deg
LHCP		6.5923 V/m	90deg,360deg		9.3197 V/m	90deg,360deg
RHCP		6.5969 V/m	90deg,10deg		9.3235 V/m	90deg,10deg
Ludwig3/X dominant		9.3229 V/m	90deg,360deg		13.18 V/m	90deg,360deg
Ludwig3/Y dominant		8.9197 V/m	90deg,90deg		12.598 V/m	90deg,90deg

4. Click **Export Fields** and save the file in the *.nfd* format.

You can use an *.and file to refer to one or more *.nfd files to describe near fields.

Related Topics

Technical Notes: Incident Waves

Clear Linked Data

Using Field Solutions from Other Simulators

Access Near Field External Data File

Using Field Solutions from Other Simulations

HFSS and HFSS-IE can use field solutions from other simulations as source designs. The source designs can be done in HFSS, SIwave or Maxwell3D. Some examples are as follows

- A detailed and optimized design of a cell phone radiating in a larger environment (HFSS to HFSS).
- A complicated printed circuit board causing EMC/EMI problems in and around its housing (SIwave to HFSS).
- An electromechanical component causing EMC/EMI problems in a vehicle (Maxwell3D to HFSS).

In all cases, radiated fields from the source project are imposed as an incident wave in the target project.

These radiated fields can both be far fields and near fields, depending on your judgment of what fits a particular situation. In the target project, they are defined through Incident Wave / Far Field Wave and Incident Wave / Near-Field Wave. There, the link to the source project can be established.

Note: The environment variable SIWAVE_INSTALL_DIR should be set before executing the parent application like HFSS/Designer because SIwave is launched from HFSS/Designer and not separately.

Also, in the target project, radiation boundaries with Advanced Options must be defined in order to specify where the fields from the source project enter the target project.

HFSS-IE Link

You can link HFSS and HFSS-IE projects. The link is controlled with a "Near Field" or "Far-Field" Incident Wave source that is grouped with the Excitations.

- Far Field Wave
- Near Field Wave

Related Topics

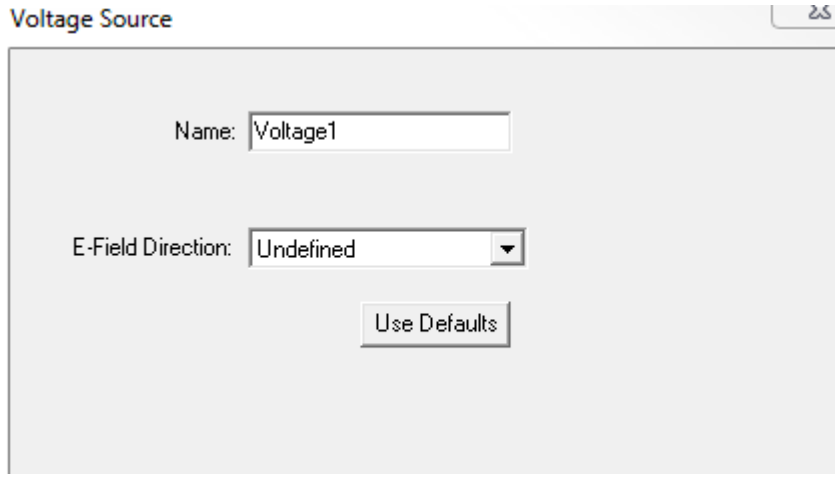
HFSS-IE Feature

Assigning Voltage Sources

A voltage source in HFSS can be defined on a surface located anywhere in the 3D problem space. Typically the source is placed on a surface between two conductors such that a user defined total voltage is maintained between the conductors. Voltage source is used when the feed structure is very small compared to the wavelength and a constant electric field may be assumed across the feed points; in this case, HFSS assigns a constant electric field across the gap on which you specified the voltage.

Note: For more information, see Voltage Source in **Technical Notes**.

1. Select the object face to which you want to assign the voltage source, and click **HFSS>Excitations>Assign>Voltage** to display the **Voltage Source** dialog box.



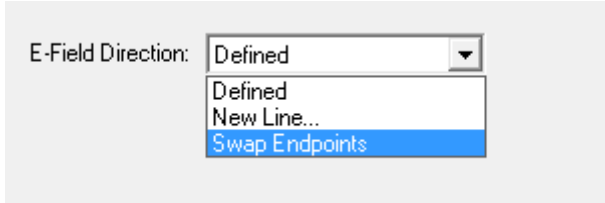
2. Specify the direction of the electric field by drawing a vector.

   When the source is selected, an arrow indicates the direction and a letter v indicates the type of source.

   Note: For Transient Solution types, you also designate sources as Active and Passive.

Port sources are created and solved with unit magnitude and 0 degree phase. They can be scaled within the Edit Sources dialog.

Note: The **E-field** direction can be reversed from the **Voltage Source** dialog box.



Related Topics

Scaling Magnitude and Phase Using the Edit Sources dialog.

Active and Passive Excitation in HFSS Transient

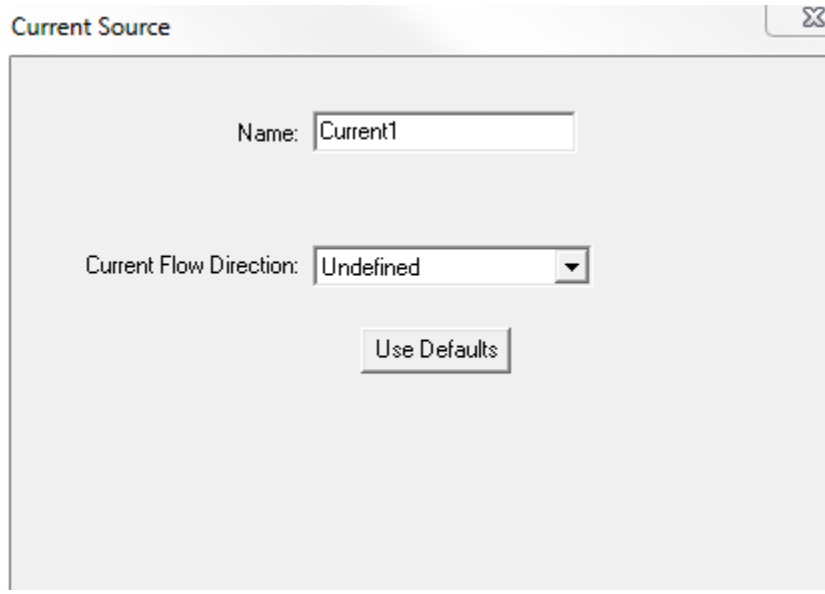
Modifying Voltage Sources

Assigning Current Sources

Assign a current source when you want to define the direction of the current flow through a surface. A current source is used when the feed structure is very small compared to the wavelength and the electric current on the surface is assumed to be constant across the feed points.

Note: For more information, see Current Sources in **Technical Notes**.

1. Select the object face to which you want to assign the current source and click **HFSS>Excitations>Assign>Current** to display the **Current Source** dialog box.



2. Specify the current flow direction by drawing a vector:

   When the source is selected, an arrow indicates the direction and a letter i indicates the type of source.

3. For Transient Solution types., you also designate sources as Active and Passive.

Port sources are created and solved with unit magnitude and 0 degree phase. They can be scaled within the Edit Sources dialog.

Related Topics

Scaling Magnitude and Phase Using the Edit Sources dialog.

Active and Passive Excitation in HFSS Transient

Modifying Current Sources

Modifying Voltage or Current Sources

To change the name, value, or electric field direction of an assigned voltage/current source:

1. Double-click the source's icon under **Excitations** in the project tree.

   The **Voltage Source/Current Source** dialog box appears.

2. Edit the name or value of the source.
3. Select **Swap Endpoints** from the **E-Field Direction**/Current Flow Direction pull-down list to reverse the direction of the e-field/current flow.

   The start and endpoints of the E-field/current flow line line are switched; the line's direction is reversed.

Assigning Magnetic Bias Sources

When you create a ferrite material, define the net internal field that biases the ferrite by assigning a magnetic bias source. The bias field aligns the magnetic dipoles in the ferrite, producing a non-zero magnetic moment.

1. Select the 3D ferrite object to which you want to assign the magnetic bias source.
2. Click **HFSS>Excitations>Assign>Magnetic Bias**.
3. Specify whether the applied bias field is **Uniform** or **Non-uniform**.

   If a design already contains a magnetic bias field, you cannot assign another of a different type. If a single bias field exists in a design, you can edit the type.
4. If you selected the **Uniform** radio button, click **Next** and do the following:
   a. In the **Internal Bias** field, type the value of the ferrite in amperes/meters. You can assign a variable as this value.
   b. Enter the rotation of the permeability tensor with respect to the xyz-coordinate system in the **X Angle**, **Y Angle**, and **Z Angle** boxes. You can assign variables to these values.

   If you selected **Non-uniform**, select the **Setup Link...** button to display the **Setup Link** dialog. Under the **General** tab, do the following:
   a. Select the radio button for **Extractor Mode** (the default) or **Interactive Mode.** Note that in **Extractor mode**, the source project will be saved upon exit. **Extractor mode** means that the software is opened during the link solely for the purpose of solving. Selecting **Interactive Mode** launches Maxwell.

      (If you open the **Setup** dialog for a pre-existing Magnetic Bias source, the **General** tab shows radio buttons for **View Only** and **Edit Link**. With **View Only** selected (the default), all of the link settings are grayed out. Selecting **Edit Link** enables the fields and changes the radio buttons to **Extractor Mode** and **Interactive Mode**.)
   b. Type the name of a Maxwell 3D Field Simulator project in the **Project File** box, or click the ellipsis [...] browse button display a file browser to select the project.

      HFSS uses the Maxwell 3D project as the source of the non-uniform magnetostatic field information during solution generation. Linking invokes a Maxwell 3D window to provide the solution for the targeted HFSS project.
   c. If there are multiple designs available for the project, you can select from the drop down menu.
   d. If there are multiple solutions available, you can select from the drop-down menu.

      The "Default" solution is the product dependent solution of the first Setup.

      That is the setup listed first in the source design's project tree (alphanumerical order). A product specific solution of this setup becomes the default solution. In most products, it is LastAdaptive. In a Transient solution type, it is "Transient."
   e. Use the radio button to specify whether to save the source path relative to **The project directory of the source project** or **This project**.

f. Use the checkbox specify whether to **Simulate source design as needed**.

g. Use the checkbox to specify whether to **preserve the source design solution**. Note that in **Extractor mode**, the source project will be saved upon exit. **Extractor mode** means that the software is opened during the link solely for the purpose of solving.

The **Setup Link** dialog also contains a **Variable Mapping** tab.

It lists variables available within the Maxwell 3D Field Simulator and the value can (and often will) be a variable in the HFSS Setup. You can edit the Value fields by typing, and the Units fields by selecting from a drop down list. You can choose to Map Variable By Name. In this case, same named variables have their values mapped automatically. Different named variables are unaffected.

To accept the settings and close the **Setup Link** dialog, click **OK**

5. Click **Finish** to close the **Magnetic Bias** wizard.

   The magnetic bias source is assigned to the selected object. If you have set up a link, HFSS invokes a Maxwell 3D window to provide the solution for the targeted HFSS project.

You can also access and edit the magnetic bias source information via the **Properties** dialog for the source. Magnetic bias sources always have the lowest priority compared to boundaries and other excitations in the solver view.

Related Topics

Reprioritizing Boundaries and Excitations.

For more information, see the following topics in Technical Notes.

Magnetic Bias Sources

Uniform Applied Bias Fields

Non-uniform Applied Bias Fields

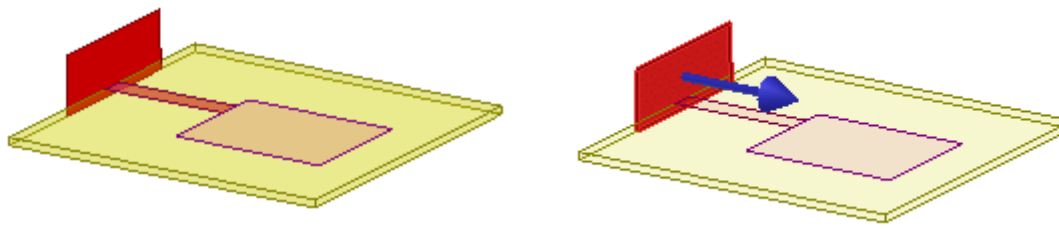
Magnetic Saturation

Applications for Deembedding

Deembedding helps to calculate the S-parameters near or on the plane of a discontinuity or when a long transmission line is attached to the port plane instead of explicitly modeling it in HFSS. For lossless ports, when you deembed them into or out of the model, there is change only in the phase of the S-parameters but not in the magnitude. For lossy ports, aside of the change in the phase of the S-parameters, there is only a slight change in the magnitude. Deembedding saves time and significantly reduces the simulation efforts.

Extract Input Impedance: Use Deembedding

This section describes how to extract the input impedance in the microstrip fed patch antenna model shown in the figure below. The port (highlighted in red ink) is placed at an appropriate distance away from the discontinuity. We cannot place the port near the intersection of the trace and the patch antenna i.e. at the discontinuity. However, if you want to measure the input impedance at the discontinuity, you can deembed the port as shown by the blue arrow in the figure.



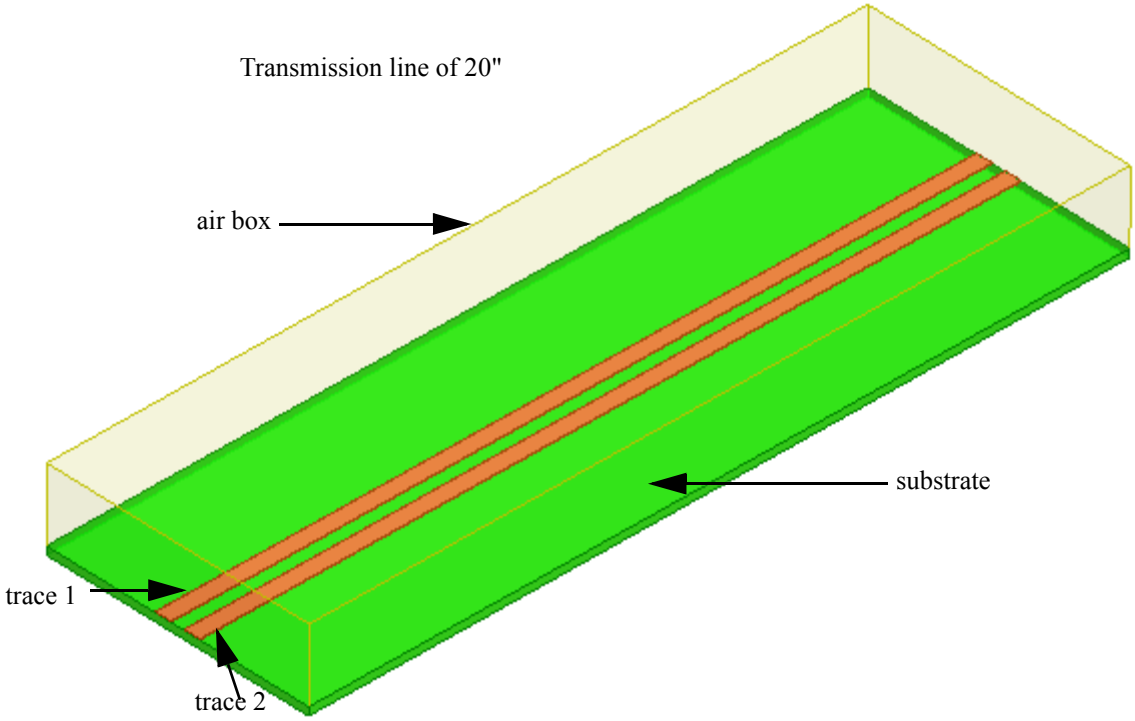
For lossless ports, deembedding will not change the magnitude of the S-parameters. It will only change the phase.

Related Topics

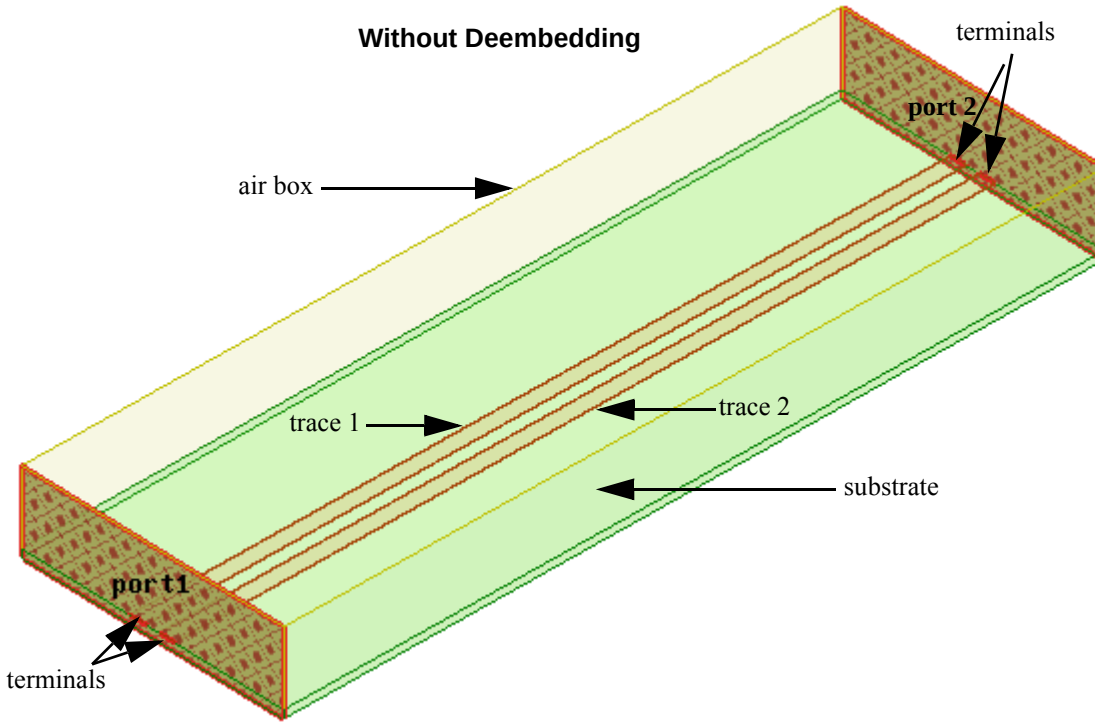
Waveport Placement.

Modeling Long Transmission Line: Use Deembedding and Port Solver

Suppose you want to model a 20 inch differential pair PCB microstrip transmission line as shown below.



Of course you can explicitly draw a 20 inch length of the microstrip model, define a port on either end and extract a 4-terminal S-parameter matrix from the simulation.



However, this model is 2D in the transverse plane since the distribution of the field occurs only in the transverse plane and does not vary in magnitude along the transmission line where all that changes is the relative phase of the fields. To extract S-parameters from such a long transmission line, you only need to model a minimal length of the transmission line and then, deembed the ports - thus leverage the information extracted from the 2D port solver and by deembedding generate all the relevant 2D aspect of the transmission line structure in its entirety.

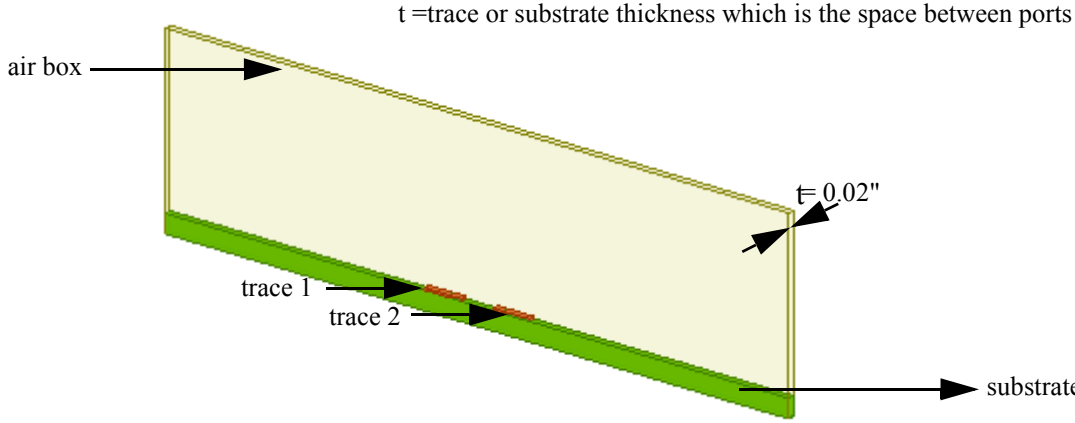
Although this is a 2D problem, HFSS being a 3D simulation tool requires creation of two ports each with two terminals separated by a minimal physical distance. For this separation distance as a rule of thumb use some dimension associated with the cross section of the transmission line such as the thickness of the trace or the substrate. Such a rule of thumb will ensure a physically small model needing fewer mesh elements than the explicitly long model as well as a mesh with high quality characteristics. Solve the model of this minimal length and then deembed outwards from the ports, using a negative sign in the deembed distance fields to effectively add the additional length to generate a model. This deembedding operation will add the effect of phase delay and additional dielectric and conduction losses to the resulting S-parameter from this model.

Note: We only use the propagation constant (referred to as gamma) to de-embed and characteristic impedance is only needed if renorm takes place.

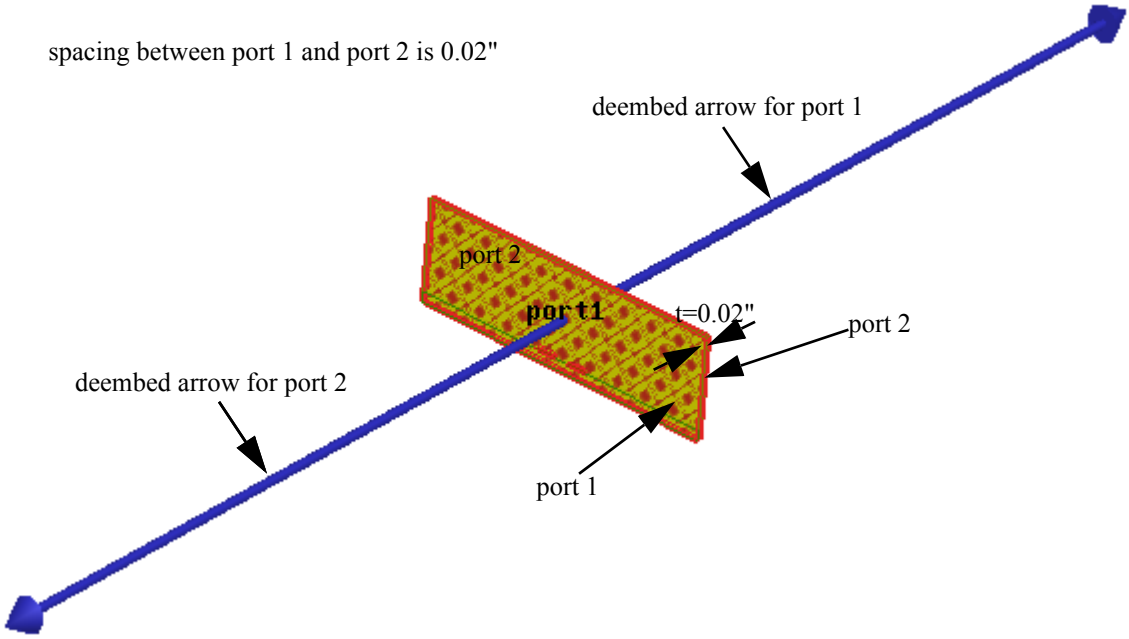
So, in the transmission line model shown in the figures below the explicit model length is 0.02" corresponding to the thickness of the microstrip trace. To extract a 20" length model from such an analysis the ports can be dembedded outwards with a length DL = (20 - 0.02)"/2 = 9.99". This model is 1000 times smaller and the simulation effort is reduced greatly.

The figures below shows the model of a 0.02" transmission line that can be used with deembedding to model a 20" long transmission line.

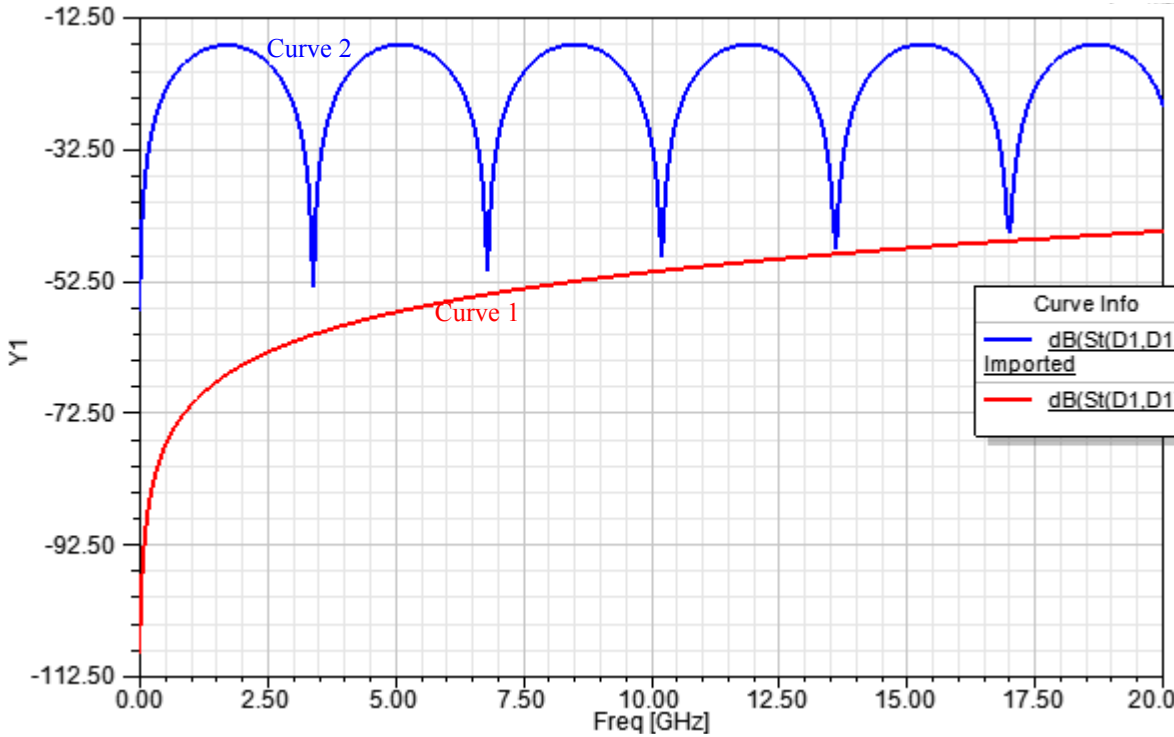
t =trace or substrate thickness which is the space between ports = 0.02"



With Deembedding



The plot for S-parameters before and after deembedding from the explicitly short model is shown below. The curve 1 (not deembedded) is a typical S(1,1) plot for a short transmission line. Curve 2 (with deembedding) represents the behavior of a long transmission line with many resonances in the frequency range.

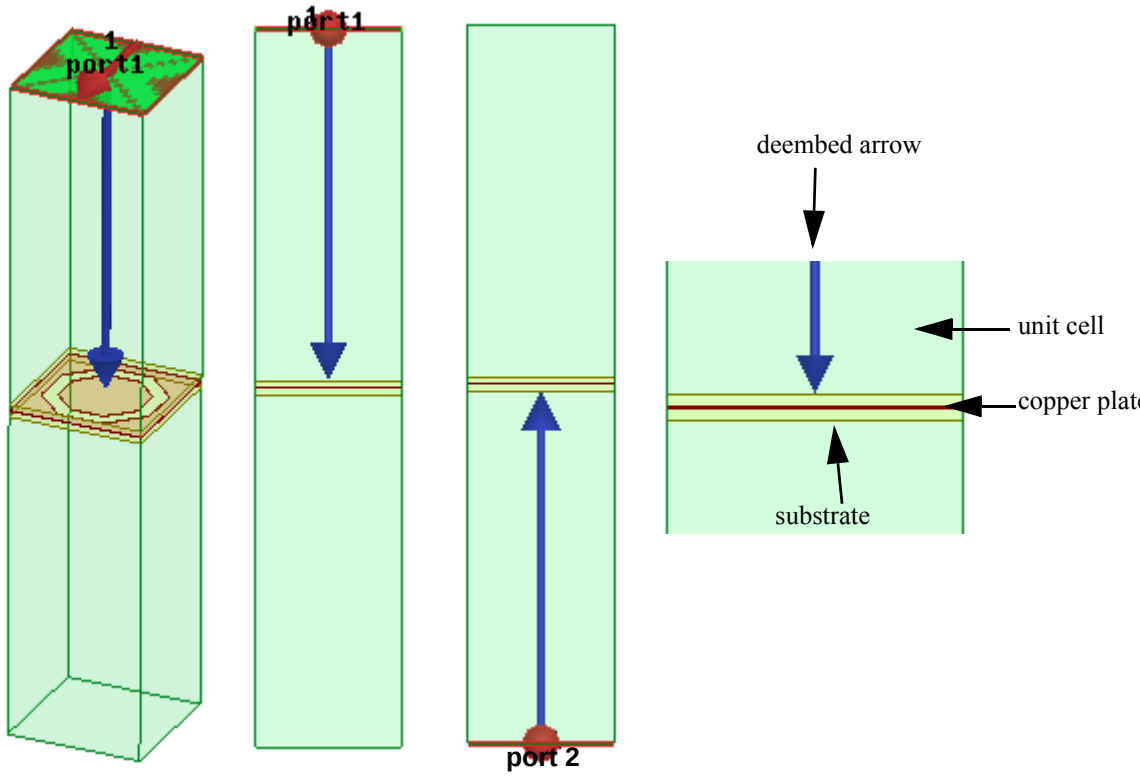


Extract Screening Impedance: Use Deembedding

This section shows how to extract the equivalent surface impedance of a screen by deembedding the distance of the ports. A unit cell of a periodic screen is modeled as shown in the figure below. The port is placed at a certain distance away from the scatterer. The screening impedance replaces a scattering planar periodic structure by homogeneous anisotropic boundary conditions.

Note: For more information, see the section on Waveport Placement.

A blue arrow depicts the dembedding distance while the port is selected, once you set the options under **Deembed Settings**. For a unit cell modeling equivalent screening impedance, the deembedding distances should point to the nearest surfaces of the substrate even if there is a thickness between these surfaces.



For port 1, the tip of the deembed arrow should touch the upper surface of the substrate. For port 2, it should touch the lower surface of the substrate.

You do not need to re-run a simulation in order to de-embed the S-matrix. Post-processing reports are automatically updated to reflect the deembedded S-matrix.

Related Topics

Exporting Matrix Data

Technical Notes: Renormalized S-Matrices

Technical Notes: De-embedded S-Matrices

Technical Notes: Deembedding

Setup Link Dialog

Linked data can be mesh, field or some other post-processing data that the source design generated. The **Setup link** dialog permits you to link the current project to another for:

- Magnetic Bias source
- Near Field Wave source
- Far Field Wave source
- Initial Mesh source
- Anisotropic Impedance Boundaries

You can link HFSS and HFSS-IE projects. This link is controlled with a "Near Field" or "Far-Field" Incident Wave source that is grouped with the Excitations.

Related Topics

Clear Linked Data

Modifying Excitations

To change the properties of an excitation, do one of the following:

- Double-click the excitation's icon under **Excitations** in the project tree.

  The excitation's properties window appears, in which you can modify its properties.
- Right-click the excitation in the project tree, and then click **Properties** on the shortcut menu.

  The excitation's dialog box appears, in which you can modify its properties.
- On the **HFSS** menu, click **List**.

  The **Design List** dialog box appears. Under the **Excitations** tab, you can modify the properties of one or more boundaries.

The following modifications to an excitation are possible:

- Change its properties.
- Delete it.
- Reassign it to another surface.
- Reprioritize it.
- Hide it from view.
- Show Nets for DC Continuity for 3D Conductors.
- Modify the impedance multiplier.
- Deembed the port.
- Set up Differential Pairs

For HFSS Transient solutions, render it Active or Passive.

Note The most common source of errors in simulations are related to defining ports. Examine your ports, their definitions, and inspect problem areas by using the **Zoom to** option. For more information, see Zoom to Selection.

To show or Hide Excitations

See Setting Boundary and Excitation Visualization Options.

See Show Nets for DC Continuity for 3D Conductors.

Related Topics

Modifying the Model View

Zoom to Selected Excitation

Active and Passive Excitation in HFSS Transient

Show Nets for DC Continuity for 3D Conductors.

Deleting Excitations

To delete *one excitation*:

1. Select the excitation you want to delete by clicking its icon in the project tree.
2. Click **Edit**>**Delete**  .

   The excitation is removed from the design and the project tree.

   For terminal solutions, if you delete a port with terminals associated with it, deleting the port also removes the associated terminals.

To delete *all excitations*:

- Click **HFSS**>**Excitations**>**Delete All**.

You can also *delete one or more excitations* in the **Design List** dialog box:

1. Click **HFSS**>**List**.

   The **Design List** dialog box appears.
2. Under the **Excitations** tab, click the row of the excitation you want to delete.
3. Click **Delete**.

Reassigning Excitations

You can reassign an excitation to another surface. This is useful when you have modified objects with assigned excitations, invalidating the excitations. For example, if you unite two objects with assigned excitations, the second object's excitation will become invalid because united objects maintain the characteristics of the first object selected. In this case, you would need to reassign the excitation or delete it.

1. Select the object or object face to which you want to assign an existing excitation.
2. Click **HFSS>Excitations>Reassign**.

   The **Reassign Excitation** window appears.
3. Select an existing excitation from the list, and then click **OK**.

   The excitation is reassigned to the object or object face.

Note: When reassigning an excitation that includes vectors in its definition, HFSS attempts to preserve the vectors with the new assignment, but this is not always possible.

Setting the Impedance Multiplier

For designs with ports.

If one or more symmetry planes have been defined or if only a wedge of a structure is modeled, you must adjust the impedance multiplier or the computed impedances will not be for the full structure.

Note: Changing the impedance multiplier invalidates solutions in projects where lumped ports are defined. In such projects, you need to re-solve the project after the change.

1. Click **HFSS>Excitations>Edit Impedance Multiplier**.

   The **Port Impedance Multiplier** dialog box appears.
2. Type a value in the **Impedance Multiplier** box.

   You can assign a variable as this value.
3. Click **OK**.

Related Topics

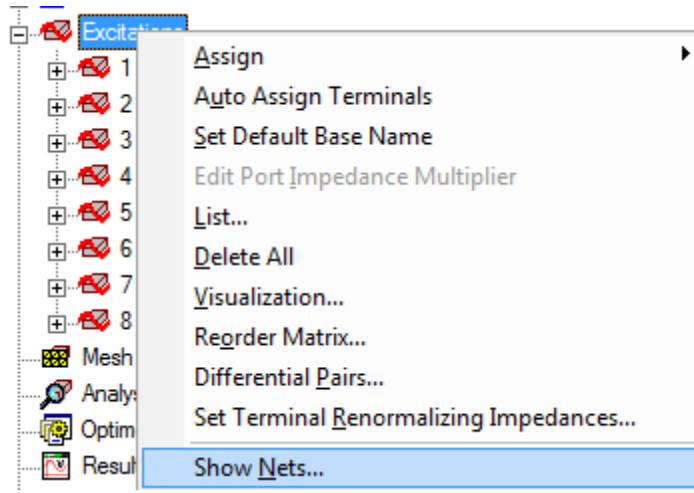
Technical Notes: Symmetry and Port Impedance

Technical Notes: Impedance Multipliers

Show Nets for DC Continuity for 3D Conductors

For Terminal Solution types, you can use the **Show Nets** command to visualize DC continuity and terminal associations for 3D conductors and terminals assigned to the edge/face of 2D objects when the 2D object or its face is assigned as a "Port" (terminal is always assigned on a port) or the port is touching the 3D conducting object.

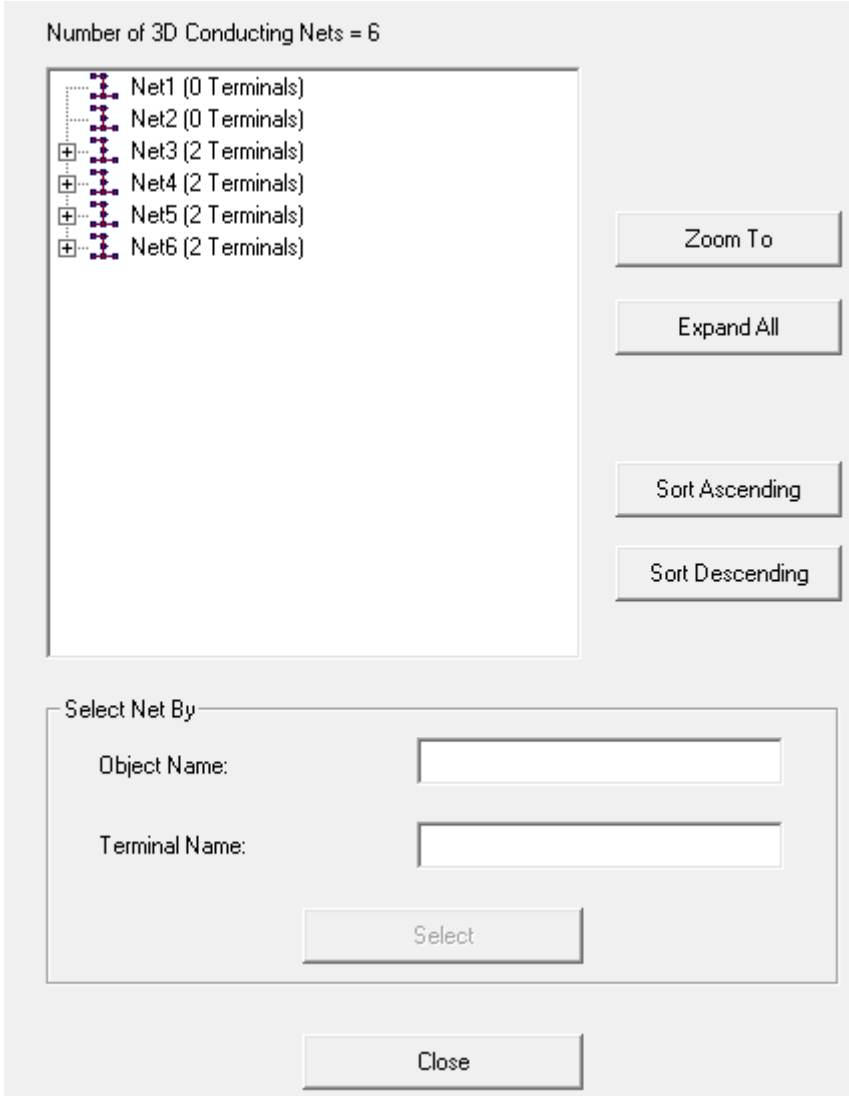
The **Show Nets** command appears in the **HFSS>Excitations** menu and by right-clicking on **Excitations** in the Project tree.



Selecting **Show Nets...** displays the **Net Visualization** dialog that lists the nets and associated terminals. You can use the dialog to select any net or terminal. You can continue to work in the Modeler window with the dialog open.

In the **Net Visualization** dialog, you can sort the nets in ascending or descending order, relative to the number of terminals. This helps you locate GND nets, which have no terminals.

Text at the top of list box provides the total number 3D Conducting nets.



You can expand or collapse the net tree. Clicking the **Expand All** button expands the tree, and the button changes to **Collapse All**. For longer lists, a slider bar lets you navigate.

Naming of nets is automatic and follows the convention: "Net1 (0 Terminals)", "Net2 (1 Terminals)"…"Net<N> (<T> Terminals)",

Net Names are not editable.

To highlight a net or terminal in the Modeler window, select the net in the tree and click the **Zoom to** button in the **Net Visualization** dialog.

The **Select Net By** fields let you select nets according to **Terminal Name** or **Object Name**. You can enter wild card expressions to select a net. For example: in the Object Name field if you specify "Box*" object name, then click the **Select** button, HFSS selects all the nets that have the object name "Box".

Related Topics

Assigning Wave Ports for Terminal Solutions

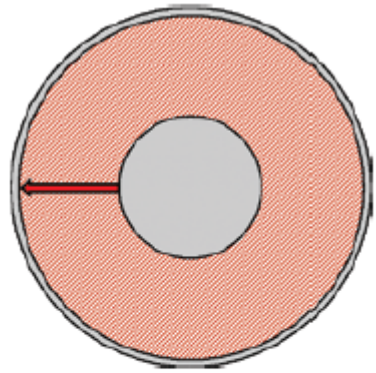
Modifying the Model View

Define an Integration Line

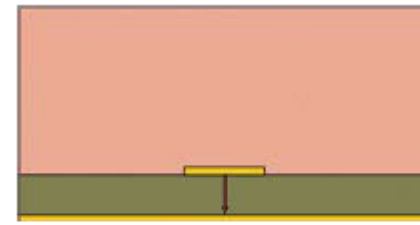
An integration line is a vector that can represent the following:

- A calibration line that specifies the direction of the excitation field pattern at a port. If you are analyzing more than one mode at a port, define a separate integration line for each mode; the orientation of the electric field differs from mode to mode.
- An line along which to integrate **E**.dl to compute a voltage for Zpv or Zvi impedance of a port. In this case, select two points at which the voltage differential is expected to be at a maximum. For example, on a microstrip port, place one point in the center of the microstrip, and the other directly underneath it on the ground plane. In a rectangular waveguide, place the two points in the center of the longer sides.

Note: For more information, see Wave Port Dialog For Modal Solutions. For definitions of how HFSS defines these Zpv and Zvi values, see Calculating the PV Impedance, and Calculating the VI Impedance.



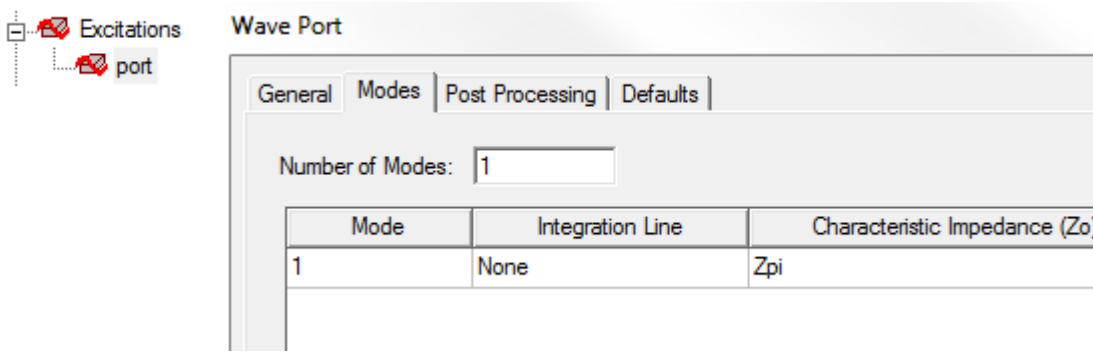
Coaxial cable wave port showing integration line (in red). Line is drawn between points of maximum potential difference. In this case, it is drawn in the space between the center conductor and the shield.



Microstrip wave port showing integration line (in red). Line is drawn between points of maximum potential difference. In this case, it is drawn in the space between the signal line conductor and the "ground" plane.

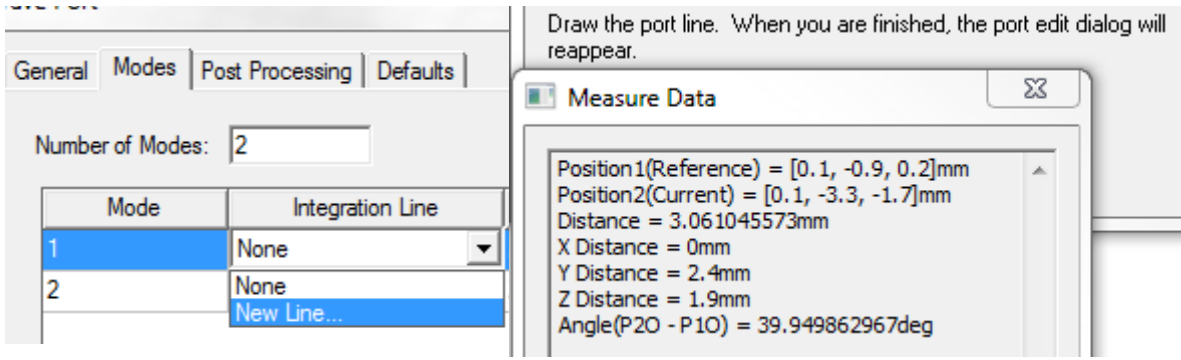
To define an integration line:

1. Double-click the port excitation from the project tree to bring up the **Wave Port** dialog box and click the **Modes** tab.



2. From the **Integration Line** column, select **New Line**.

   The port dialog box disappears. If the **Show Measure** dialog option on the Modeler Options: Drawing tab is selected, the **Measure Data** dialog appears when you draw the vector.



3. Use the **Measure Data** dialog to locate the start and end points and draw the integration line.

   Note: The **Measure Data** dialog displays data for the face area, and the positions for the reference point (start point) and end point (end point) as you define them.

Related Topics

Wave Port Dialog For Modal Solutions

Duplicating Integration Lines

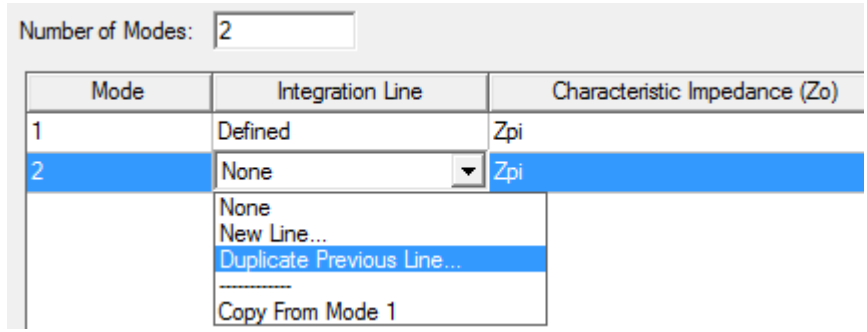
Modifying an Integration Line

Duplicating Integration Lines

You can duplicate an integration line along a vector multiple times and assign to additional modes at the port. This section shows how to duplicate the integration line in the figure below.

1. In the **Wave Port** dialog box, click the **Modes** tab and draw the integration line 1 on the port.

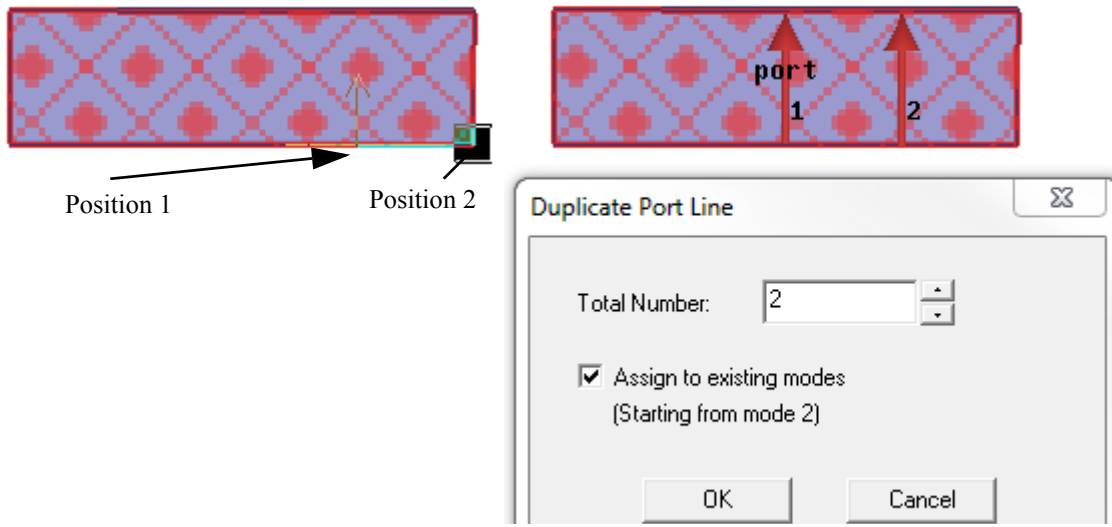
2. Enter the number of modes and select **Duplicate Previous Line** from the Integration Line column.



3. Click an arbitrary anchor point on the edge of the port face for position 1 and drag the cursor along the edge to position 2 as shown in the figures below.

   The **Integration Line 2** shifts from **Integration Line 1** by a distance equal to that between Position 1 and Position 2.

   Note: Use the **Measure Data** dialog to set this distance.



4. Enter the total number of lines, including the original and duplicates, to make in the **Duplicate Port Line**.

   If you type a value that is greater than the number of assigned modes, the extra duplicates will appear as gray integration lines until they are assigned to a mode.

5. Optionally, select **Assign to existing modes**. The duplicates will be assigned to the modes defined for the port, beginning with the mode after the one with the line that was duplicated.

Modifying Integration Lines

Modify an existing integration line under the **Modes** tab in the **Wave Port** or **Lumped Port** dialog boxes.

To *swap the coordinates* of an integration line's start point and endpoints:

- Select **Swap Endpoints** from the mode's **Integration Line** list.

  The line's direction will be reversed.

To *copy* a previously defined Wave Port integration line's points:

- Select **Copy from Mode*n*** from the mode's **Integration Line** list.

  The new integration line will have the same start and endpoints as the selected mode's integration line.

To *delete* a defined integration line for a mode:

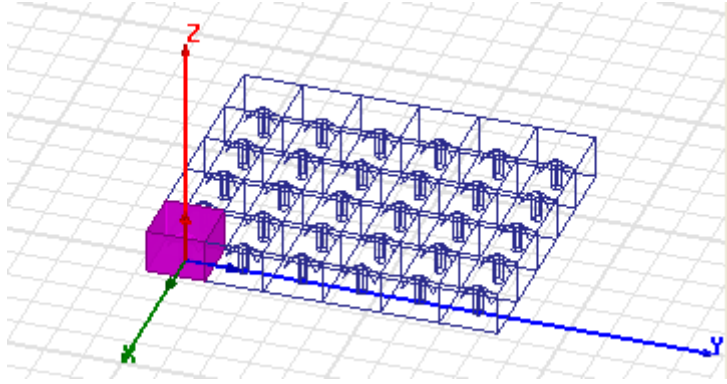
Select **None** from the mode's **Integration Line** list.

10

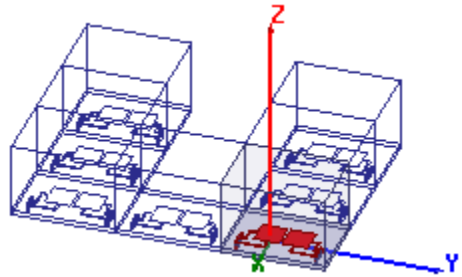
Arrays

HFSS permits you to assign a virtual array based on a unit cell object. For driven Modal and driven Terminal designs, you can then simulate the array model using distributed processing, treating the instances as parent and child objects. This permits faster definition, display, and simulation of array based designs, such as antenna arrays. You can plot and animate array fields on cutplanes, lines or points. Post processing lets you view fields on any virtual instance.

The unit cells for an array can be rectangular, parallelogram, or hexagonal. You can define the required master and slave boundaries so as to create offset arrays. You can only edit the settings in the physical cell and these settings will be applied to the corresponding instances in the virtual cells.



Once you have defined an array, you can designate any cell in the array as active or passive, or as padding. You can use the padding cell designation to define arbitrarily irregular arrays. Cells designated as padding are treated as background material for fields calculations.



Most boundaries and excitations defined in the physical unit cell will have their corresponding instances in each virtual cell. The exception is incident wave, which is applied across the whole model and should include the 'expanded' model based on the array setup.

The basic process flow for using **Create Array** is:

1. Draw the unit cell, containing all appropriate boundaries and source definitions.
2. Create the antenna array, including name, dimensions, master and slave boundaries where needed for conformal meshing, and selection of row and column master/slave pairs for implicit definition of lattice propagation vectors. Designate which cells are active, passive, and padding.
3. Setup the distributed processor pool. Designs with arrays require HPC licenses.
4. Provide a memory statistic for the amount of RAM guaranteed on each DSO processor.

In the Setup, Enable Solver Domains is disabled because an array solve uses UI defined domains, not solver defined domains. Given a valid configuration, an Array solve can use a distributed memory solution.

The UI will provide the antenna array definition to the domain manager. This will cause the following to occur:

1. Instantiation of domains to represent the cells of the antenna array plus surrounding air padding cells.
2. Creation of internal domain manager data structures that are needed to support the solve and post processing. This includes appropriate domain parent/child relationships, transformations from the physical domain, interface information per pair of domains, and support for locating a domain by row/column coordinates within the antenna array.

For linking to Designer, the network data from HFSS will include both physical and virtual cells. This applies to both port locations and push excitations.

For Optimetrics solution quantities of both virtual and physical cells can be used for calculation.

For 2D Reports for models with Arrays, matrix solution quantities of virtual cells will be expanded into a vector in the same fashion as without the array. The entries are listed according to their [row, column] order in the corresponding "expanded" matrix.

For Port Field Display there is no GUI change. Only physical ports/terminals will be listed. There is no need to support visualization of user-selected cell (like field overlay plot) because the field patterns of the virtual modes are the same as those in the physical cells.

For designs with an Array, the Edit Sources dialog listing order will be as follows: Sources will be listed according to their cell [row, column] order in the array. For each cell, port/terminals are listed in creation/assignment order with mode in each port listed sequentially. Other type of sources, such as incident waves, will be listed after ports/terminals.

There will be no change in the far/near field pattern setup and far/near fields will be computed from radiation surfaces on all cells (both physical and virtual).

Copy/Paste design will copy an array. Copy/Paste geometry will NOT copy an array.

Related Topics

Creating a Unit Cell for an Array

Create Array Command

Array Visualization

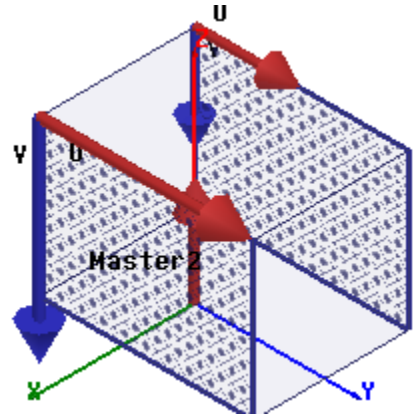
Setup and Run an Array Simulation

Post Processing Array Models

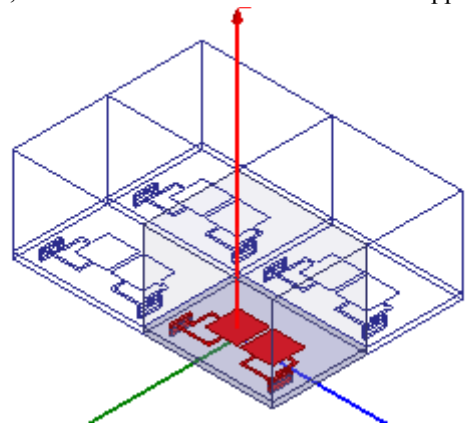
Creating a Unit Cell for an Array

You use the Modeler to create a unit cell for an array.

- The unit cell can be rectangular, parallelogram or hexagonal.
- You must define appropriate master and slave boundaries, as well as other boundaries required for the model. Deleting a master or slave boundary that is referenced by an existing array also deletes the array. Typically, you define master and slave boundaries as opposite sides of a cell.

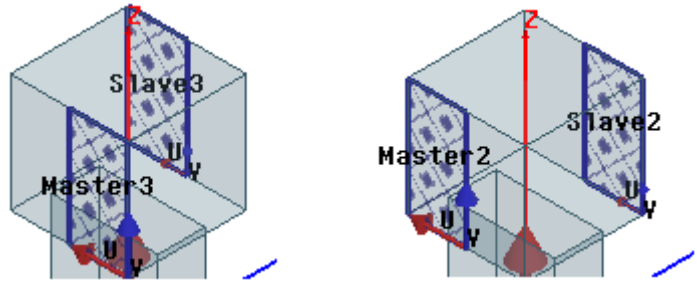


In such cases, with master and slave boundaries on opposite sides, the array sides align.

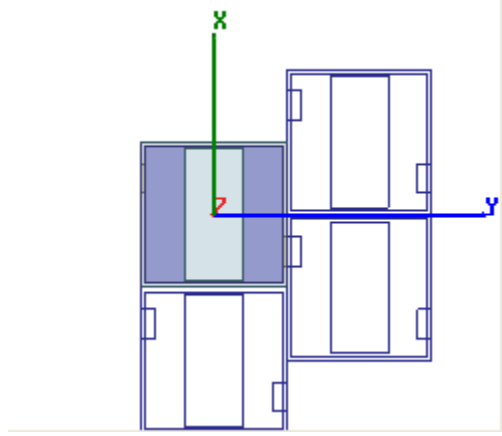


However, you can arrange master and slave boundaries to create an offset array. In such cases, you create both a master and a slave boundary for corresponding sections of a cell side, and

assign master and slave to create offset alignment.



With all boundaries defined in this manner, the array can be offset.



- In making source definitions for unit cell for the unit cell, you cannot assign a Floquet port.
- Each driven Modal or driven Terminal design can contain a single array.

Editing the model object for an existing array invalidates any existing solutions.

Related Topics

Drawing a Model

Create Array Command

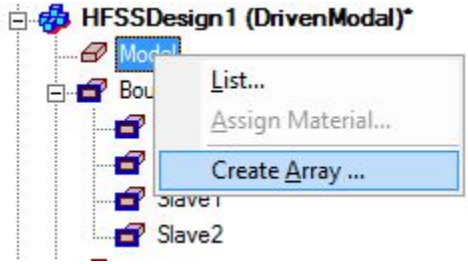
Array Visualization

Setup and Run an Array Simulation

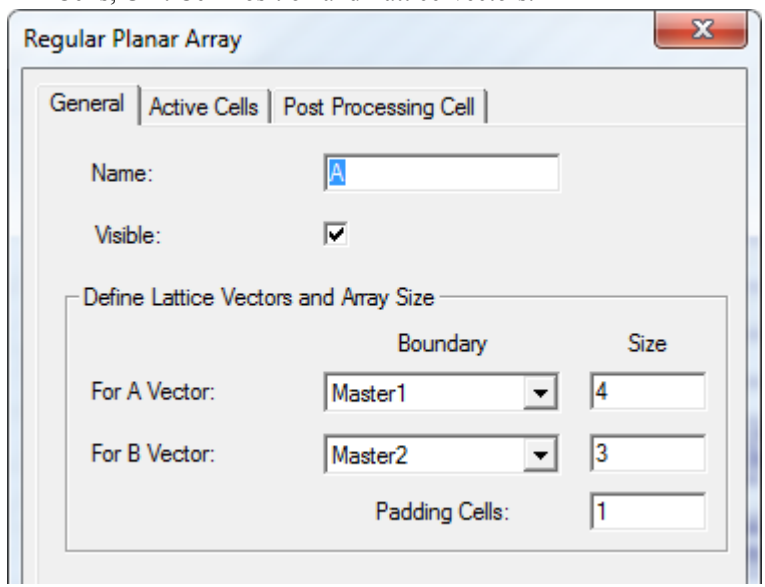
Post Processing Array Models

Create Array Command

The **Create Array** command is enabled for Modal and Driven Terminal problems after you have assigned master and slave boundaries to your unit cell model. You can access the command in three ways: click **HFSS>Model>Create Array**, right-click on the Model icon in the Project tree and select **Create Array** from the shortcut menu, or select the unit cell in the modeler window, and select **Create Array** from the short-cut menu.

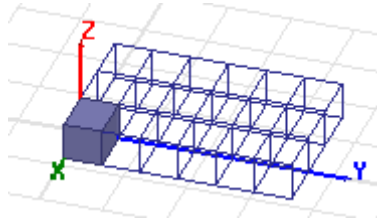


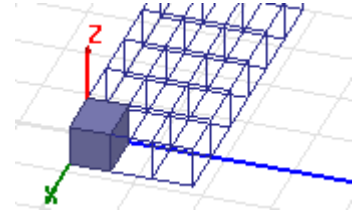
This displays the **Regular Planar Array** dialog where you specify parameters for Number of Cells, Unit Cell Position and Lattice Vectors.



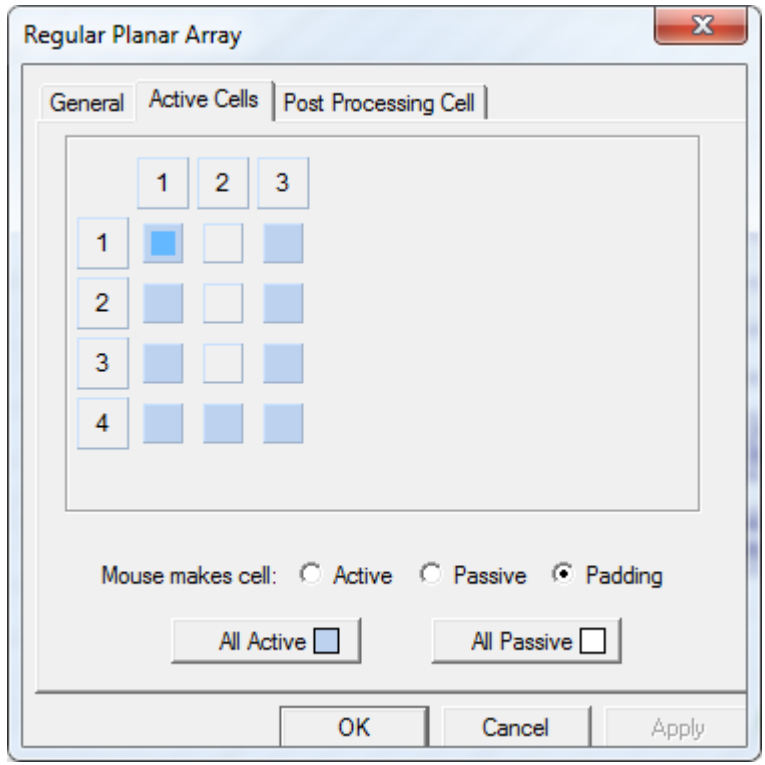
- If you have **Visible** enabled, you can see any changes.
- To Define Lattice Vector directions for A and B Vectors, specify the master boundary.
- To define the array size, for the A and B vectors, specify the number of cells for each row and column, respectively.
- For example, the following figure shows the results when the initial Lattice Vectors are

swapped.t

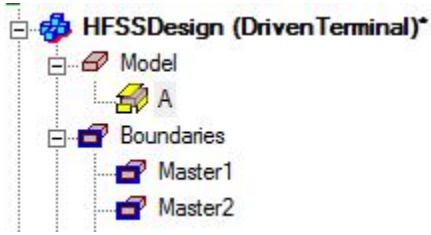




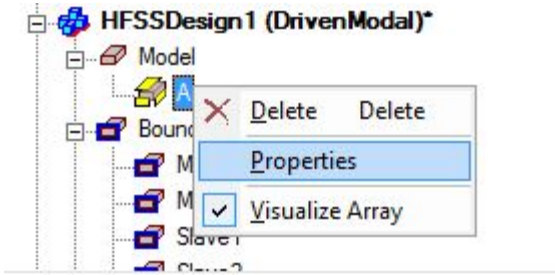
To create an irregular array, select the Active Cells tab, select the radio button for Mouse makes cell as Padding



Once you have specified the parameters, the Array object appears in the Project tree under the Model. Only one Array is permitted for a model. The **Create Array** command is disabled if an array is defined.

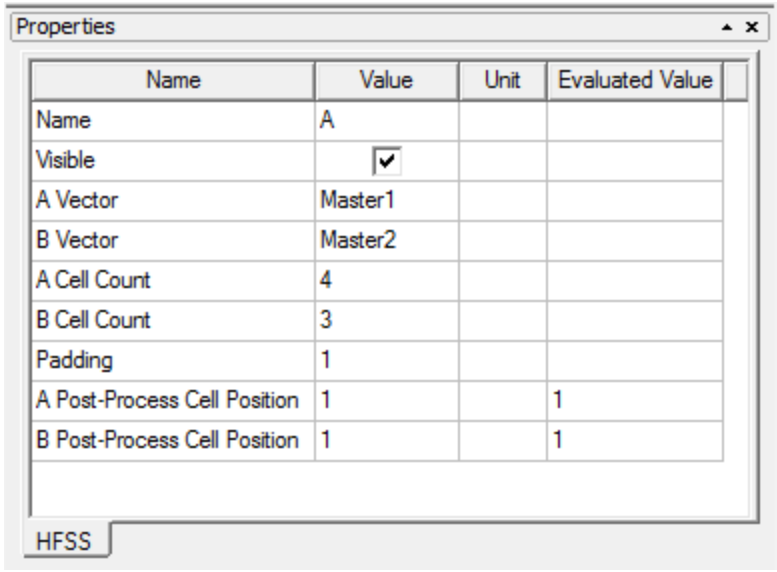


By selecting the Array icon in the Project tree, you right-click for the short cut menu.



- Selecting **Delete** removes the array from the Model.
- Selecting **Properties** displays the **Regular Planar Array** dialog.
- Selecting Visualize Array lets you toggle the array display. A check mark indicates that the array is being displayed.

With the Array icon in the Project tree selected, if you have a docked Properties window displayed, you can see and edit Array Properties. All edits are undoable and informational messages will be posted to the Message window when design data is deleted.



Related Topics

Creating a Unit Cell for an Array

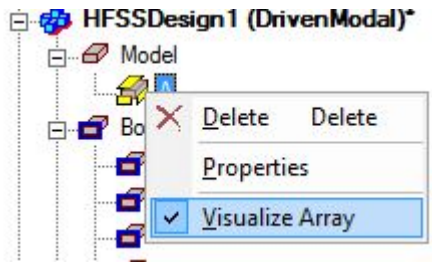
Array Visualization

Setup and Run an Array Simulation

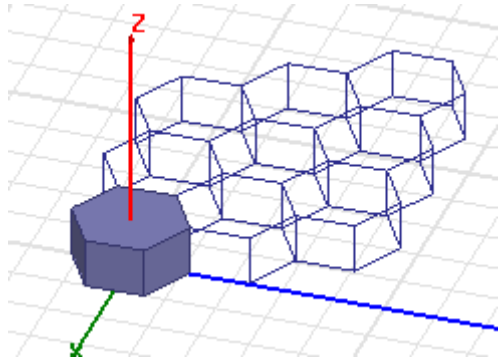
Post Processing Array Models

Array Visualization

You control the visualization of an array by selecting an existing Array in the Project tree and toggling the **Visualize Array** command on the short-cut menu.



Virtual objects in an array display as wire frames. If the all virtual objects for an array do not appear in the view modeler window, perform a **View>Fit All** operation. If you change the view, you can also use **Ctrl-D** to fit the array into the current view.



You can disable the array visualization by clicking on the **Visualize Array** option again. A check mark appears in front of Visualize Array if the option is already ON. This option will also be present in general active view visibility options, **View>Active View Visibility**.

The visualize array option will be applied to the current active view. You can have multiple windows in the modeler, each window will have its own **Visualize Array** setting. This allows you to use different windows to look at unit cell display and whole array display at the same time.

If the **Visualize Array** option is set to true, the array will be visible when you open the project.

Related Topics

Creating a Unit Cell for an Array

Create Array Command

Setup and Run an Array Simulation

Post Processing Array Models

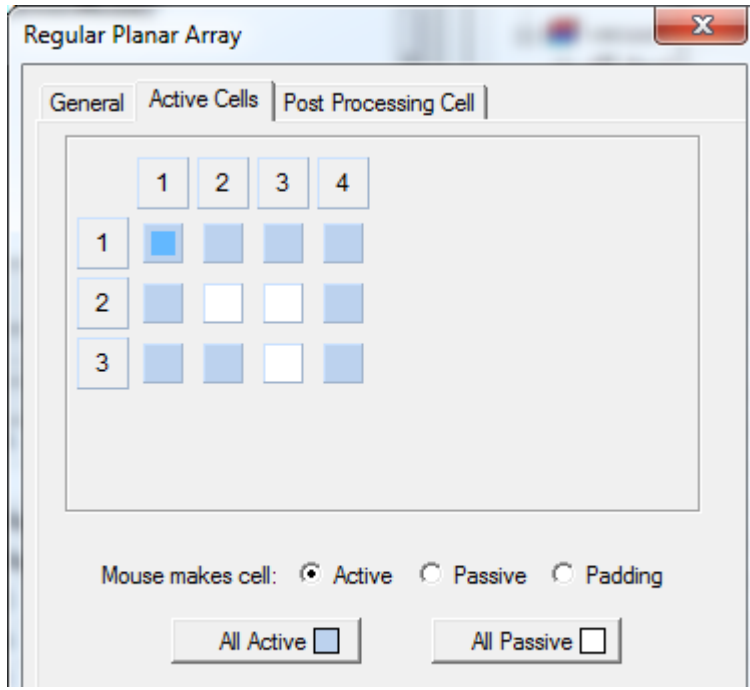
Setup and Run an Array Simulation

There is no change in the GUI for setting up the adaptive simulation for designs with an array. However, the solution quantities of the virtual cells will be available for convergence setup (In both adaptive and interpolating sweep). There are no changes in the way convergence information is presented on the Convergence tab of the Solution Display panel.

If your design contains a virtual array, the setup can have some differences.

- **Enable Solver Domains** should not be checked (these are UI defined domains, not solver defined domains).
- Setup the distributed processor pool. Designs with arrays require HPC licenses.
- General Setup for Virtual Array Simulation for Matrix Convergence, if you choose Selected Entries.
- Interpolating Sweep Advanced Options for Array Simulation
- Fast sweep is not supported.
- You can also setup the expression cache at solve setup. The expression cache interface for accessing array elements is the same as those used in report setup.
- Use the **Active Cells** tab on the **Regular Planar Array** dialog to designate which cells are active or passive for a simulation. You can make All Active, All Passive or select which cells are active or passive. The more active cells there are for a simulation, the more processing required. By default, clicking the corresponding array elements toggles the current selection, You can also choose the **Mouse makes cell** setting to click for Active or Passive, whichever is most convenient. Clicking on a row or column number applies the mouse click command to all cells in that row or column. Dragging the cursor over cells performs the current operation on

them.



It is important to understand the impact of passive ports on antenna parameters. For accepted power calculations, passive ports are not included when computing the total power passing through the union of all port surfaces. This means that the passive ports can be viewed as a loss mechanism for the device and it is not equivalent to viewing the passive ports as active ports with zero excitations.

- Report setup for Arrays.

  The solution/matrix quantities are grouped by category. The entries in each category are listed according to their [row, column] order in the corresponding matrix.

  The entry in [row1, column1] will be listed first, followed by

  [row1, column2], … [row1, columnN], [row2, column1], …

  [row2, columnN], … [rowN, columnN]. Note that the [row, column] order of each entry in the matrix is controlled by the 'Matrix' order as specified by user.

  The existing "Filter" capability can help locate the desired quantity from the potentially very long list.

Related Topics

Creating a Unit Cell for an Array

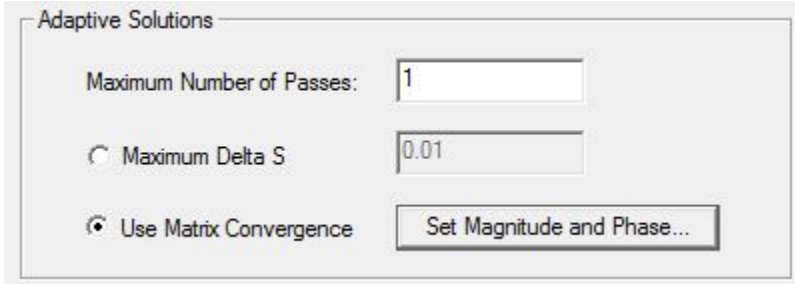
Create Array Command

Array Visualization

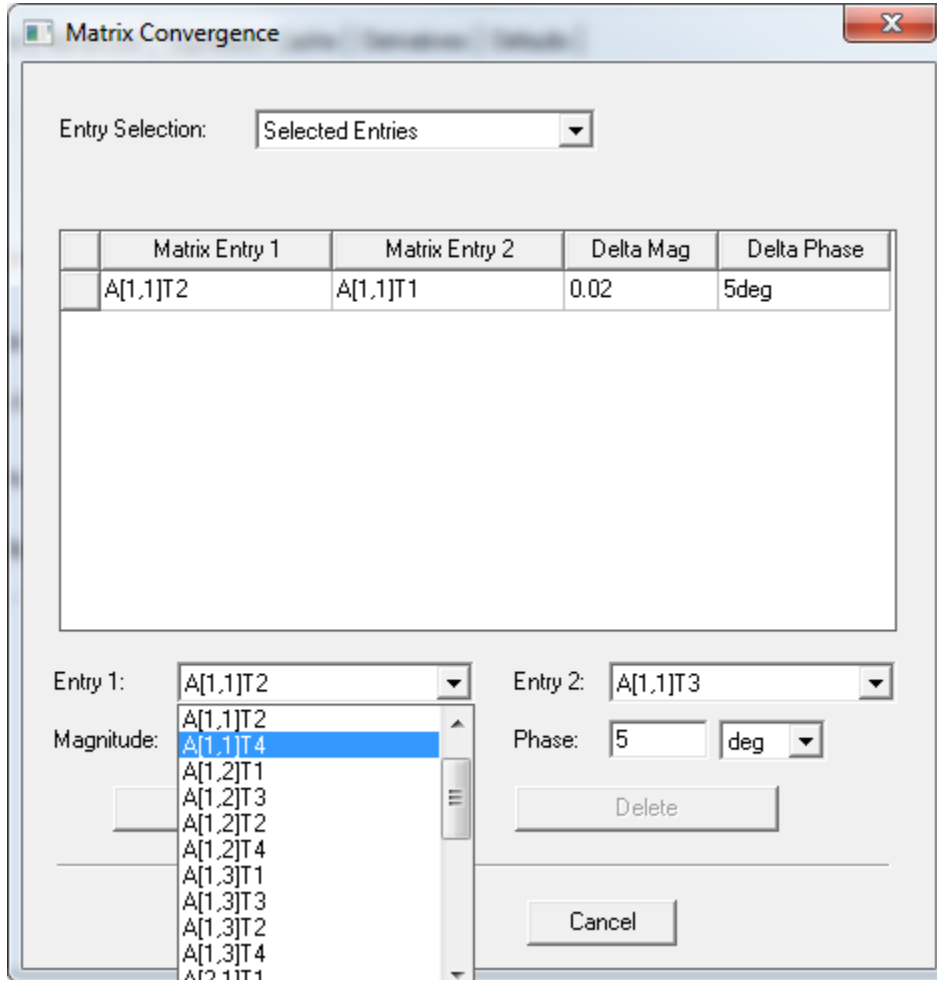
Post Processing Array Models

General Setup for Virtual Array Simulation

For a project with an array the General Setup some differences appear in how you can specify Matrix convergence.



If you select Matrix Convergence, and click **Set Magnitude and Phase**, you will see the Matrix convergence dialog. In the Matrix Convergence dialogue, if for **Entry Selection** you choose Selected Entries, (rather than All or Diagonal/Off Diagonal), you will see scrollable drop down menus that let you select from all Array elements to define pairs of Matrix entries.



Related Topics

Creating a Unit Cell for an Array

Create Array Command

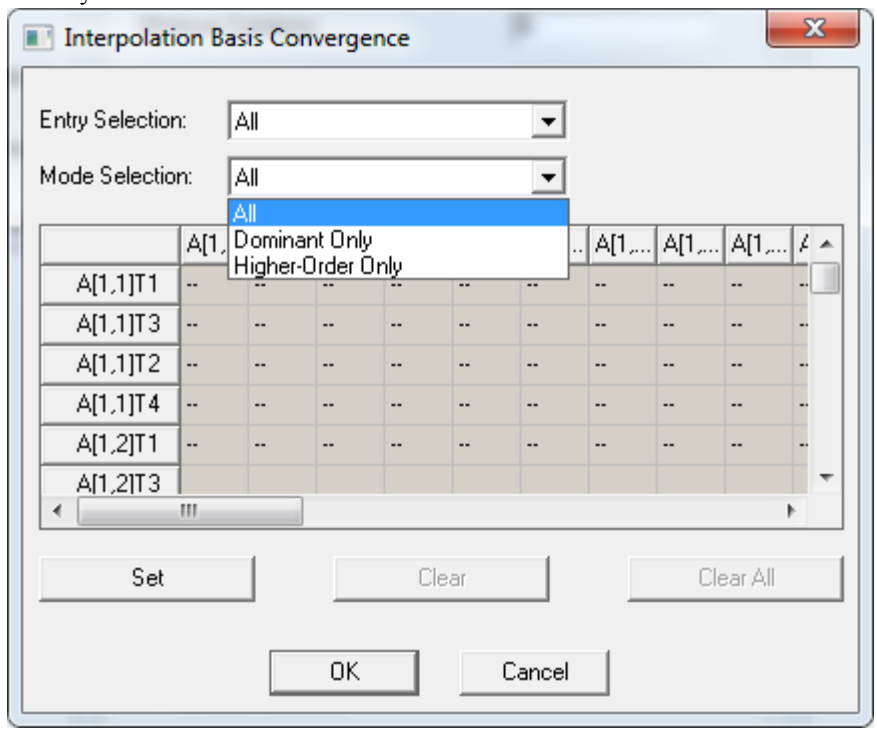
Array Visualization

Post Processing Array Models

Interpolating Sweep Advanced Options for Arrays

For an Interpolating Sweep Advanced Options, if you select **Use Selected Entries**,

clicking the **Select Entries** button displays an Interpolation Basis Convergence dialog that lists the Array elements.



Note that the matrix entries are listed according to the Matrix Sort Order as specified by user.

Entry Selection can be All, Diagonal, or Off-Diagonal.

Mode Selection can be All, Dominant Only, or Higher-Order Only.

Related Topics

Creating a Unit Cell for an Array

Create Array Command

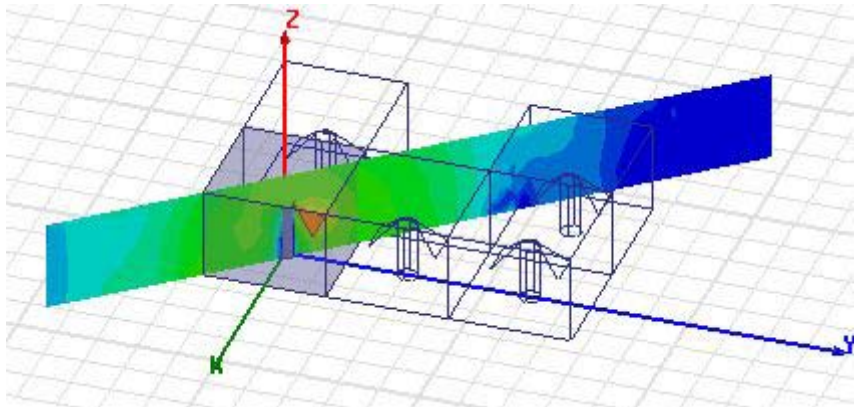
Array Visualization

Post Processing Array Models

Post Processing for Array Models

Solution quantities of both physical and virtual cells are available for post-processing.

Radiation fields (near and far) post processing is over the whole array. You can plot and animate fields on non-model cut planes, vectors, and points, as well as on selected model object faces. Any padding cells in or around the array are treated as background material.You can plot and animate fields on cut planes, vectors, and points.



You can use the Fields calculator to define calculated expressions.

HFSS can also post process individual cells in an array one at a time. You can select an arbitrary cell and do fields post processing on that cell. HFSS post processes on a single user selected cell in the array.

For Port Field Display there is no GUI change. Only physical ports/terminals will be listed. There is no need to support visualization of user-selected cell (like field overlay plot) because the field patterns of the virtual modes are the same as those in the physical cells.

- Reports for Arrays
- Plotting Fields for Array Models
- Fields Calculator Applications for Geometries in Arrays
- Fields Post Processing for Selected Cells in Arrays

Related Topics

Creating a Unit Cell for an Array

Create Array Command

Array Visualization

Setup and Run an Array Simulation

Reports for Arrays

There are some differences in the Report setup for Arrays.

1. The solution/matrix quantities are grouped by category. The entries in each category are listed according to their [row, column] order in the corresponding matrix.

   The entry in [row1, column1] will be listed first, followed by

   [row1, column2], … [row1, columnN], [row2, column1], …

   [row2, columnN], … [rowN, columnN]. Note that the [row, column] order of each entry in the matrix is controlled by the 'Matrix' order as specified by user.

2. The existing "Filter" capability can help locate the desired quantity from the potentially very long list.

Related Topics

Creating a Unit Cell for an Array

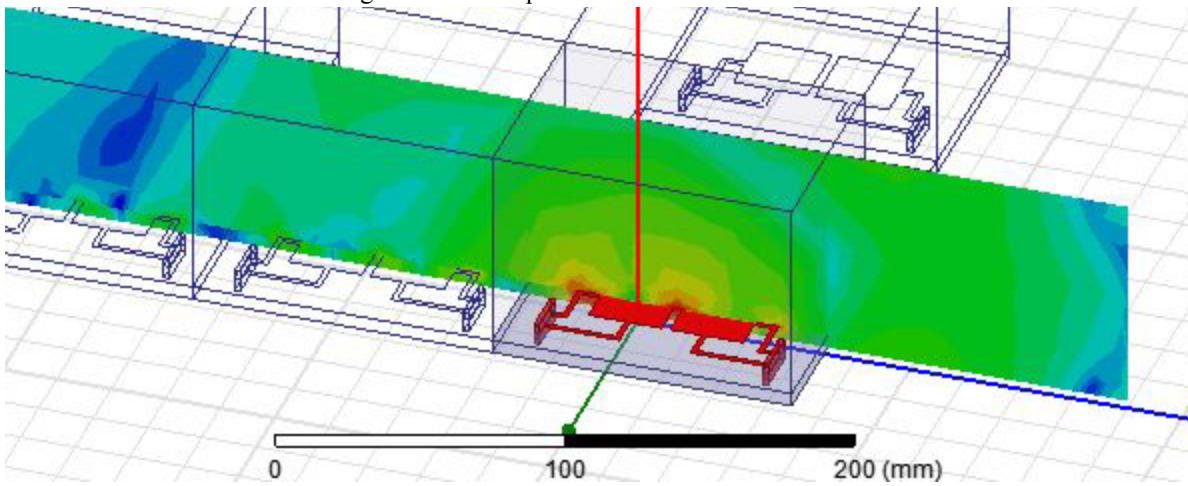
Create Array Command

Array Visualization

Setup and Run an Array Simulation

Field Plots for Arrays

You can generate field plots on object faces, as well as on non-model planes, lines, and points. The plots display calculated fields where ever the selected geometry intersects the array cells based on whether the cells are active, passive, or padding in the array properties. Locations designated as padding are treated as background material in field calculations. The virtual cells do not need to be visible to affect the generated field plots.



You can also create animations of field plots. If the plotted geometry is controlled by a swept variable (such as plane angle or a point location) the animation displays the correct values where ever they intersect the array based on the designations as active cells, passive cells, or padding cells.

Related Topics

Creating Animations

Fields Calculator Applications for Arrays

Geometries selected for Fields Calculator expressions can intersect any active, passive, or padding cells defined for the array. You can generate animated field output in which each frame is a snapshot of the fields on a different plane of the modeled volume. The Fields Calculator cookbook describes an example of the technique. Any derived field quantity can be plotted in this manner.

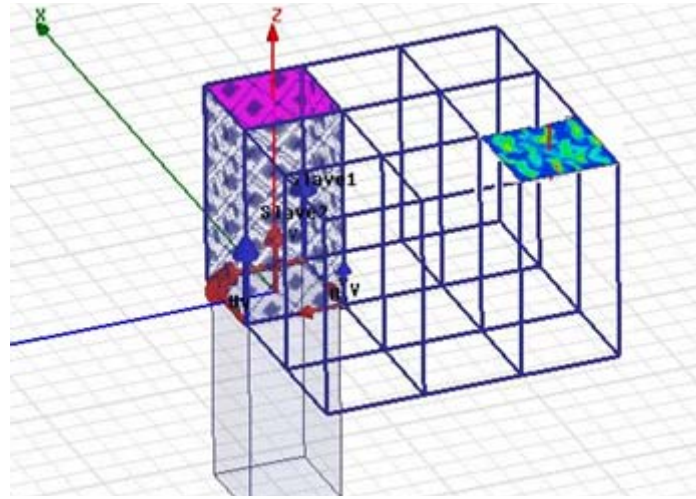
Related Topics

Using the Fields Calculator

Fields Post Processing on a Designated Array Cell

For post processing fields information for arrays, you can select an arbitrary cell and do fields post processing on that cell.

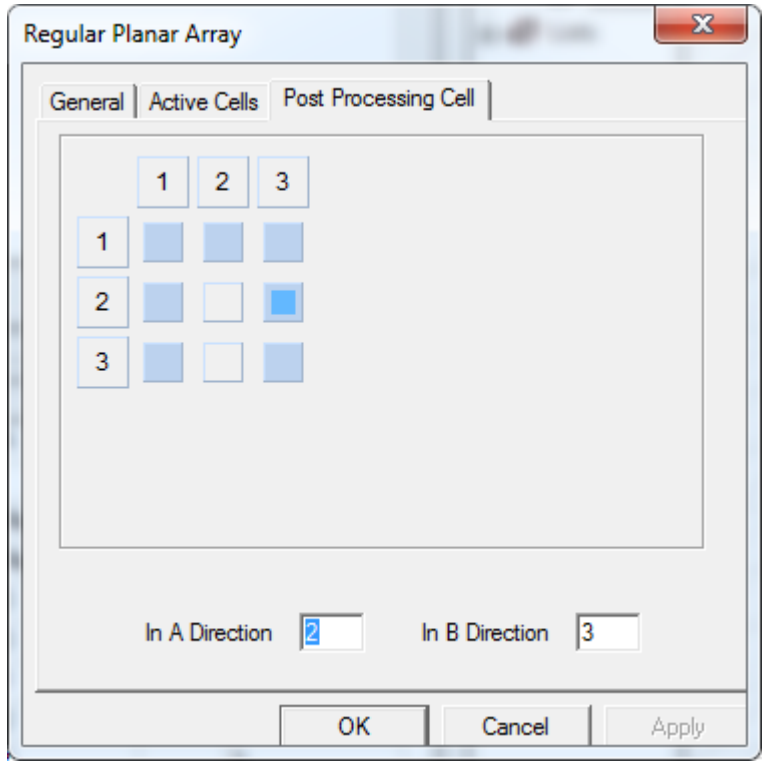
The unit cell mesh from position (1,1) will be translated to the user selected cell and fields will be plotted in the user selected cell.



The procedure is:

1. Select objects in the unit cell
2. Use the Array dialog to specify the cell to post process for fields, either by clicking on a the corresponding array position indicator in the Post Processing Cell tab in or by giving the A and

B direction indices (row and column respectively).



The Field plot will be visible in the user selected cell.

Radiation fields post processing will be over the whole array.

Jsurf is calculated inside the geometry of the cell but it will be equal to 0 along cell borders.

Related Topics

Creating a Unit Cell for an Array

Create Array Command

Array Visualization

Setup and Run an Array Simulation

11

Assigning Materials

You can add, remove, and edit materials in two main ways:

- Using the **Tools>Edit Configured Libraries>Materials** menu command.
- Right-clicking **Materials** in the project tree and selecting **Edit All Libraries**.

Editing definitions from the project window does not modify the configured libraries for any particular design. To consider the current design, use the **Tools>Edit Configured Libraries** option. Doing so ensures that new libraries are added to the configured list for the current design. If you edit materials from this command for the current and then export them, they will also be available to assign to objects in other designs.

To assign a material to an object, follow this general procedure:

1. Select the object to which you want to assign a material.
2. Click **Modeler>Assign Material**  .

   The **Select Definition** window appears. When the **Show all libraries** checkbox is selected, the window lists all of the materials in the global material library as well as the project's local material library.

   You can also open the **Select Definition** window in one of the following ways:

   - In the **Properties** dialog box for the object, click the material name under the **Attributes** tab. A drop-down menu shows an **Edit...** button that opens the **Select Definition** window.

     The menu also lists materials included in the current project. Selecting one of these materials provides another way to assign materials to an object.
   - Right-click **Model** in the project tree, and then click **Assign Material** on the shortcut menu.
   - Right-click the object in the history tree, and then click **Assign Material** on the shortcut menu.

3. Select a material from the list.

Note You can search the listed materials by name or property value.

If the material you want to assign is not listed, add a new material to the global or local material library, and then select it.

4. Click **OK**.

The material you chose is assigned to the object.

Note For HFSS-IE, problems with dielectrics will generally run more slowly, so HFSS-IE issues a warning for this case. In HFSS-IE:

- Only isotropic materials are allowed
- Frequency dependent materials are allowed.
- No spatial dependent materials are allowed.
- You cannot set background material in the solver. Solver assumes vacuum as background material.

Note In the history tree, by default, HFSS groups objects by material. To change the default, select the object icon and right-click to display the **Group Objects by Material** checkbox.

Related Topics

Solve Inside or On a Surface

Assigning DC Thickness

Searching for Materials

Adding New Materials

Assigning Material Property Types

Defining Variable Material Properties

Defining Frequency Dependent Material Properties

Defining Material Properties as Expressions

Defining Functional Material Properties

Viewing and Modifying Material Attributes

Validating Materials

Copying Materials

Removing Materials

Export Materials to a Library

Solving Inside or on the Surface

For HFSS designs, when you assign a material to an object, you can specify whether to generate a field solution inside the object or on the surface of the object. If you elect to generate a solution inside the object, HFSS will create a mesh inside the object and generate a solution from the mesh. If you elect to generate a solution on the surface of the object, HFSS will create only a surface mesh for the object.

If you want a solution to be generated inside an object, select **Solve Inside** in the **Properties** window. Conversely, if you want a solution to only be generated on the surface of an object, clear the **Solve Inside** option in the **Properties** window.

By default, **Solve Inside** is selected for all objects with a bulk conductivity less than 10^5 siemens/meter and for perfect insulators. By default, the **Solve Inside** option in the **Properties** window is clear for perfect conductors.

To change the threshold for solving inside objects, do the following:

1. Click **Tools**>**Options**> **HFSS Options**.
2. Under the **General** tab, enter a new value in the **Solve Inside threshold** text box.

HFSS implicitly places a finite conductivity boundary condition on the surfaces of an object that should not be solved inside based on the material properties of that object. You can also incorporate a DC Thickness for the implicit boundary condition by setting an appropriate thickness value as described in Assigning DC Thickness.

Related Topics

Assigning DC Thickness

Assigning DC Thickness

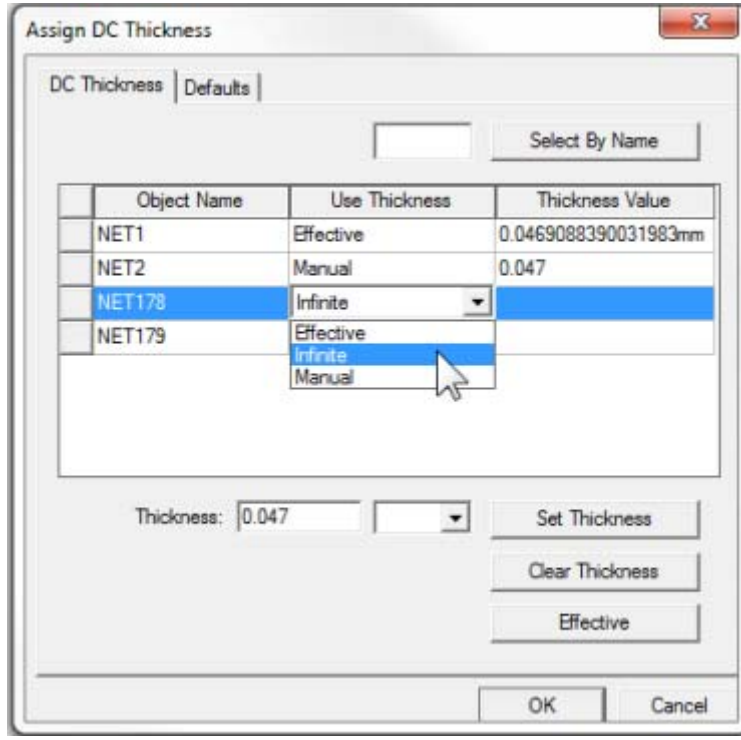
You can select the **Assign DC Thickness** option to more accurately compute DC resistance of a thin conducting object for which Solve Inside is not selected. Skin impedance of the object will be calculated using the defined finite thickness, which can be Effective (the default), Manually assigned per object, or Infinite per object. This option also exists for finite conductivity boundaries.

The **Assign DC Thickness** option on the **HFSS>Boundaries** menu is enabled if at least one object contains a good conducting isotropic material (such as copper), and the **Solve Inside property** is not selected. If the object meets these conditions, you can assign a DC thickness, either by enabling the Effective default, or by specifying a value for a selected object.

To see the **Assign DC Thickness** dialog:

1. Select **HFSS>Boundaries>Assign DC Thickness**.

   This displays the **Thickness of Objects for DC Resistance** dialog with the DC Thickness tab selected. Objects to which the thickness can be applied are listed in the **Object Name** column.



2. Select the objects to assign a value. You can select objects either by:
   - Clicking on the **Object Name** to highlight it.
   - Use the **Select By Name** field to type the object name, and click the **Search** button.

     The first object to match the name is highlighted.

Selecting an object highlights the **Thickness** field and the **Set Thickness** button.

3. In the Use Thickness column, you can specify that the value the object uses is Effective, Infinite, or Manual.

   You can disable automatic assignment of the Effective calculation on the **Defaults** tab of the **Assign DC Thickness** dialog.

   The Effective value is calculated as Thickness ~ 2* Volume / Surface Area.

   It should be noted that this is a calculation for an "effective" DC thickness to be used by the correction calculation. For a "thin" object this will work well. For example, a rectangular microstrip trace described by a box with dimensions 100 by 10 by 1 the volume is 1000 and the surface area is 2022 resulting in an apparent thickness of .99891, close to the geometric thickness of "1". For arbitrary shapes of "thicker" objects, this calculation will not work as well. For example, a cube with sides of 1 will have volume of 1 and surface area of 6, and a resulting apparent thickness of .3333. Another example is the case of a cylindrical wire (e.g. bond wire). In this case, the automatic effective DC thickness will be about R_o, which gives us the best approximation of the DC resistance of a cylindrical wire.

   The intention is that the Effective thickness will provide an accurate representation the majority of the time and is superior to not using any DC thickness setting. When the Effective value is not appropriate, you can override it using the manual technique.

   DC thickness impedance is an approximation. It is accurate just for TE/TM waves when the widths are infinite which clearly never occurs in a "real" design. The fact that the object is finite causes an increase of the effective impedance due to current crowding/edge effects. Thus entering the exact geometrical thickness would actually underestimates the impedance. By returning a DC thickness smaller than the geometric thickness the Effective DC thickness compensates for this underestimation resulting in a slightly higher impedance as desired.

4. To manually apply a value, enter a Thickness value, select the units and click the **Set Thickness** button.

   This applies the value to the selected object and changes the Use Thickness selection for that object to Manual.

Note If you enter a "0" for the thickness, HFSS gives a warning that this will cause infinite impedance that causes isolation.

5. To change the value, select the **Clear Thickness** button and then enter a different value. You can also manually select or deselect the box and manually enter or delete a thickness value in the table.
6. When you have assigned the values you need, click **OK** to close the dialog.

Related Topics

Technical Notes: Calculating Finite Thickness Impedance

Searching for Materials

If there is a specific material or material property value that you want to assign to an object, you can search the materials in the **Select Definition** window by name or by material property.

Searching by Material Name

1. In the **Search Criteria** area of the **Select Definition** window, select **by Name**.
2. In the **Search Parameters** area, type a material name in the **Search by Name** text box.

   The row containing the material name most similar to the one you typed will be selected. If the letter typed provides an exact match, the cursor moves to the right. If there is no match, the cursor moves to the beginning of the search field. If you type a space after word, the cursor moves to the beginning of the search field. If the name you want to search for contains a space, you can type the next letter after the space, backspace, and insert a space.

If the selected material is not the one you are searching for, do one of the following:

- Use the keyboard's arrow keys to scroll up or down the list of materials.
- Type a new material name in the **Search by Name** text box.

Searching by Material Property

1. In the **Search Criteria** area of the **Select Definition** window, select **by Property**.
2. Select a material property from the pull-down list:

Note By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

3. In the **Search Parameters** area, type a value for the property in the **Search by Property** text box, and then click **Search**.

   The materials are sorted according to the value you entered. The material with the property value closest to the one you typed will be selected.

If the selected material is not the one you are searching for, do one of the following:

- Use the keyboard's arrow keys to scroll up or down the list of materials.
- Type a new value in the **Search by Property** text box.

Adding New Materials

You can add a new material to a project or global user-defined material library. To make the new project material available to all projects, you must export the material to a global user-defined material library.

Materials are added using the **View/Edit Material** dialog box, which can be opened from either the **Select Definition** dialog box or the **Edit Libraries** dialog box.

To open the **Select Definition** dialog box:

- Click **Modeler>Assign Material**.
  The **Select Definition** dialog box appears.

To open the **Edit Libraries** dialog box:

- Click **Tools>Edit Configured Libraries>Materials**. (In the project tree, you can also right-click **Materials**, and select **Edit All Libraries**.)
  The **Edit Libraries** dialog box appears.

To add a new material:

1. From either the **Select Definition** dialog box or the **Edit Libraries** dialog box, click **Add Material**.

   The **View/Edit Material** dialog box appears.
2. Type a name for the material in the **Material Name** text box, or accept the default.
3. Select one of the following from the **Material Coordinate System Type** pull-down list:
   - **Cartesian** (default)
   - **Cylindrical**
   - **Spherical**
4. Use the radio buttons in the **View/Edit** section to specify whether the new materials apply to Active Design, This Product, or All Products.
   - **Active Design** to display properties used in the active design.
   - **This Product** to display properties commonly used by this product.
   - **All Products** to display all properties available. This enlarges the table of properties to show all properties possible. You can use the scroll bars or size the dialog to see all properties.

   You can also enable the View/ Edit Modifier checkbox for Thermal Modifier. Checking this box causes the Thermal Column to display at the right side of the Properties of the Material table. Selecting **Edit** rather than None causes display of the Edit Thermal Modifier dialog.
5. Type a new name for the material in the **Material Name** text box or accept the default.
6. Select a material property type - **Simple** or **Anisotropic** - for each property from the **Type**

pull-down list.

Note By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

7. If the material is *linear*, enter values for the following material properties in the **Value** boxes:
   - Relative Permeability
   - Relative Permittivity
   - Bulk Conductivity
   - Dielectric Loss Tangent,
   - Magnetic Loss Tangent

   If the material is a *ferrite*, enter a value greater than 0 in the Magnetic Saturation **Value** box. You may also choose to enter values in the Lande G Factor and Delta H **Value** boxes. Because Delta H values are measured at specific frequencies, you should also enter a - Measured Frequency value (default 9.4 GHz).

Note You may enter a variable name or mathematical expression in the **Value** box.

8. If one or more of the material properties are dependent on frequency, click **Set Frequency Dependency**, and then follow the directions for defining frequency dependent materials.
9. To modify the units for a material property, double-click the **Units** box, and then select a new unit system.
10. Click **OK**.

    The new material is added to the material library.

Related Topics

Defining Variable Material Properties

Assigning Material Property Types

Defining Frequency-Dependent Material Properties

Spatially Dependent Materials in HFSS

Specifying Thermal Modifiers

Assigning Material Property Types

Each material property can be assigned one of the following material property types:

Simple	The material is homogeneous and linear.
Anisotropic	The material's characteristics vary with direction.

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for anisotropic permittivity, electric loss tangent, conductivity, permeability, and magnetic loss tangent. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

Related Topics

Setting Coordinate Systems

Creating a Relative Coordinate System

Change the Orientation of an object

Defining Anisotropic Relative Permeability Tensors

Defining Anisotropic Relative Permittivity Tensors

Defining Anisotropic Conductivity Tensors

Defining Anisotropic Dielectric Loss Tangent Tensors

Defining Magnetic Loss Tangent Tensors

Defining Anisotropic Relative Permeability Tensors

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for anisotropic permeability. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

1. In the **Relative Permeability** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.

   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Relative Permeability** row.
2. Enter the relative permeability along one axis of the material's permeability tensor in the **Value** box of the **T(1,1)** row.
3. Enter the relative permeability along the second axis in the **Value** box of the **T(2,2)** row.
4. Enter the relative permeability along the third axis in the **Value** box of the **T(3,3)** row.

If the relative permeability is the same in all directions, use the same values for each axis.

These values can also be defined as variables.

Related Topics

Technical Notes: Anisotropic Relative Permeability Tensors

Setting Coordinate Systems

Creating a Relative Coordinate System

Change the Orientation of an object

Defining Anisotropic Relative Permittivity Tensors

Defining Anisotropic Conductivity Tensors

Defining Anisotropic Dielectric Loss Tangent Tensors

Defining Magnetic Loss Tangent Tensors

Defining Anisotropic Relative Permittivity Tensors

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for anisotropic permittivity. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

1. In the **Relative Permittivity** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.

   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Relative Permittivity** row.
2. Enter the material's relative permittivity along one tensor axis in the **Value** box of the **T(1,1)** row.
3. Enter the relative permittivity along the second axis in the **Value** box of the **T(2,2)** row.
4. Enter the relative permittivity along the third axis in the **Value** box of the **T(3,3)** row.

If the relative permittivity is the same in all directions, use the same values for each axis.

These values can also be defined as variables.

Related Topics

Technical Notes: Anisotropic Relative Permittivity Tensors

Setting Coordinate Systems

Creating a Relative Coordinate System

Change the Orientation of an object

Defining Anisotropic Relative Permeability Tensors

Defining Anisotropic Conductivity Tensors

Defining Anisotropic Dielectric Loss Tangent Tensors

Defining Magnetic Loss Tangent Tensors

Defining Anisotropic Conductivity Tensors

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for anisotropic conductivity. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

1. In the **Bulk Conductivity** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.

   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Bulk Conductivity** row.

2. Enter the conductivity along one axis of the material's conductivity tensor in the **Value** box of the **T(1,1)** row.
3. Enter the conductivity along the second axis in the **Value** box of the **T(2,2)** row.
4. Enter the conductivity along the third axis in the **Value** box of the **T(3,3)** row.

The values of the conductivity along the first and second axis apply to all axes that lie in the xy cross-section being modeled. The values of the conductivity along the third axis applies to the z-component. These values affect current flowing in dielectrics between the conductors.

These values can also be defined as variables.

Related Topics

Technical Notes: Anisotropic Conductivity Tensors

Setting Coordinate Systems

Creating a Relative Coordinate System

Change the Orientation of an object

Defining Anisotropic Relative Permeability Tensors

Defining Anisotropic Relative Permittivity Tensors

Defining Anisotropic Dielectric Loss Tangent Tensors

Defining Magnetic Loss Tangent Tensors

Defining Anisotropic Dielectric Loss Tangent Tensors

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for electric loss tangent. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

If electric loss tangent is anisotropic, do the following:

1. In the **Dielectric Loss Tangent** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.

   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Dielectric Loss Tangent** row.
2. Enter the ratio of the imaginary relative permittivity to the real relative permittivity in one direction in the **Value** box of the **T(1,1)** row.
3. Enter the ratio of the imaginary relative permittivity to the real relative permittivity in the second direction in the **Value** box of the **T(2,2)** row.
4. Enter the ratio of the imaginary relative permittivity to the real relative permittivity in the third orthogonal direction in the **Value** box of the **T(3,3)** row.

If the electric loss tangent is the same in all directions, use the same values for each direction.

These values can also be defined as variables.

Related Topics

Technical Notes: Anisotropic Dielectric Loss Tangent Tensors

Setting Coordinate Systems

Creating a Relative Coordinate System

Change the Orientation of an object

Defining Anisotropic Relative Permeability Tensors

Defining Anisotropic Relative Permittivity Tensors

Defining Anisotropic Conductivity Tensors

Defining Magnetic Loss Tangent Tensors

Defining Magnetic Loss Tangent Tensors

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for anisotropic magnetic loss tangent. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

1. In the **Magnetic Loss Tangent** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.

   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Magnetic Loss Tangent** row.
2. Enter the ratio of the imaginary relative permeability to the real relative permeability in one direction in the **Value** box of the **T(1,1)** row.
3. Enter the ratio of the imaginary relative permeability to the real relative permeability in the second direction in the **Value** box of the **T(2,2)** row.
4. Enter the ratio of the imaginary relative permeability to the real relative permeability in the third direction in the **Value** box of the **T(3,3)** row.

If the magnetic loss tangent is the same in all directions, use the same values for each direction. These values can also be defined as variables.

Related Topics

Technical Notes: Anisotropic Magnetic Loss Tangent Tensors

Setting Coordinate Systems

Creating a Relative Coordinate System

Change the Orientation of an object

Defining Anisotropic Relative Permeability Tensors

Defining Anisotropic Relative Permittivity Tensors

Defining Anisotropic Conductivity Tensors

Defining Anisotropic Dielectric Loss Tangent Tensors

Defining Variable Material Properties

When defining or modifying a material's properties, each material property value in the **View/Edit Material** window can be assigned a project variable. Simply type the project variable's name in the appropriate **Value** box. Project variables are used for material properties because materials are stored at the project level.

For example, define a project variable with the name **MyPermittivity** and define its value as **4**. To assign this property value to a material, type **$MyPermittivity** in the **Relative Permittivity Value** box for the material. Be sure to include the prefix **$** before the project variable name, which notifies HFSS that the variable is a project variable.

Note By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

Related Topics

Working with Variables

Assigning Material Property Types

Defining Frequency-Dependent Material Properties

Specifying Thermal Modifiers

Defining Frequency-Dependent Material Properties

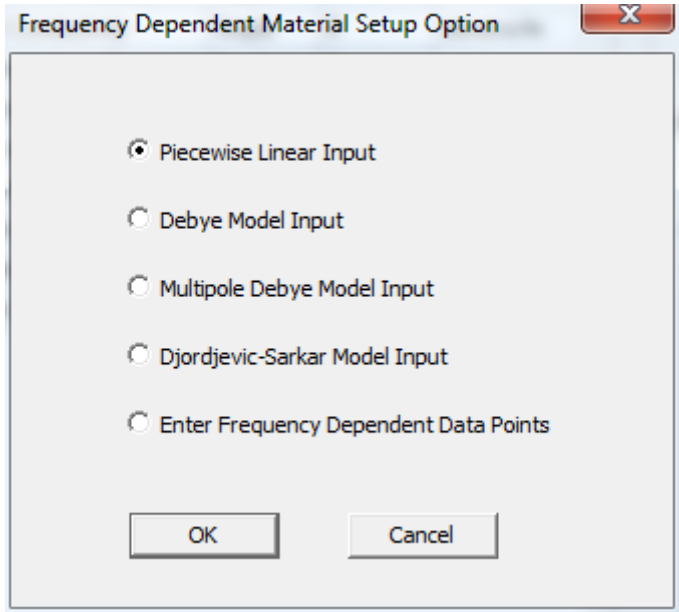
HFSS provides several frequency-dependent material models. The Piecewise Linear and Frequency Dependent Data Points models apply to both the electric and magnetic properties of the material. However, they do not guarantee that the material satisfies causality conditions, and so they should only be used for frequency-domain applications.

The Debye and Djordjevic-Sarkar models apply only to the electrical properties of dielectric materials. These models satisfy the Kramers-Kronig conditions for causality, and so are preferred for applications (such as TDR or Full-Wave Spice) where time-domain results are needed. The HFSS Design Settings also include an automatic Djordjevic-Sarkar model to ensure causal solutions when solving frequency sweeps for simple constant material properties.

In HFSS, you can assign conductivity either directly as bulk conductivity, or as a loss tangent. This provides flexibility, but you should only provide the loss once. The solver uses the loss values just as they are entered.

1. With respect to a material selected in the **Select Definition** window, in the **View/Edit Mate-**

rial window, click **Set Frequency Dependency**.



2. In the **Frequency Dependent Material Setup Option** window, do one of the following:
   - Select **Piecewise Linear Input**. This defines the material property values as a restricted form of piecewise linear model with exactly 3 segments (flat, linear, flat). You will specify the property's values at an upper and lower corner frequency. Between these corner frequencies, HFSS linearly interpolates the material properties; above and below the corner frequencies, HFSS extrapolates the property values as constants. This dataset can be modified with additional points if desired.
   - Select **Debye Model Input**. This is a single-pole model for the frequency response of a lossy dielectric material. In some materials, up to about a 10-GHz limit, ion and dipole polarization dominate and a single pole Debye model is adequate. HFSS allows you to specify an upper and lower measurement frequency, and the loss tangent and relative permittivity values at these frequencies. You may optionally enter the permittivity at optical frequency, the DC conductivity, and a constant relative permeability.
   - Select **Multipole Debye Model Input**. This lets you provide the data of relative permittivity and loss tangent versus frequency. Based on this data the software dynamically generates frequency dependent expressions for relative permittivity and loss tangent through the Multipole Debye Model. The input dialog plots these expressions together with your input data through the linear interpolations.
     - The generated expressions provide the new value for the material properties of relative permittivity and loss tangent.
     - Both the expressions and data triples can be saved and reloaded.
   - Select **Djordjevic-Sarkar Model Input**. This model was developed for low-loss dielec-

tric materials (particularly FR-4) commonly used in printed circuit boards and packages. In effect, it uses an infinite distribution of poles to model the frequency response, and in particular the nearly constant loss tangent, of these materials. HFSS allows you to enter the relative permittivity and loss tangent at a single measurement frequency. You may optionally enter the relative permittivity and conductivity at DC.

If you try to enter invalid values for the Djordjevic-Sarkar model, you receive error messages.

- Select **Enter Frequency Dependent Data Points**. This allows you to enter, import or edit frequency dependent data sets for each material property. Any number of data points may be entered. This is an arbitrary piecewise linear model. A simple data set will not provide a causal material.

3. Click **OK**.

   A dialog appears, based on your selection.

   Piecewise Linear Input

   Debye Model Input

   Multipole Debye Model Input

   Djordjevic-Sarkar

   Enter Frequency Dependent Data Points

   After you have entered the data for your selection, you return to the **View/Edit Material** window. New default function names appear in the material property text boxes. HFSS automatically created a dataset for each material property. Based on a varying property's dataset, HFSS can interpolate the property's values at the desired frequencies during solution generation.

   To modify the dataset with additional points, see Modifying Datasets.

Note Neither the piecewise or the loss models ask for frequency dependent conductivity because there the constant sigma represents the DC loss and the frequency dependent loss tangent represents the polarization losses.

Related Topics

Piecewise Linear Input

Debye Model Input

Multipole Debye Model Input

Djordjevic-Sarkar

Enter Frequency Dependent Data Points

Technical Notes: Multipole Debye Model

Technical Notes: Frequency-Dependent Material Properties

Technical Notes: Djordjevic-Sarkar Causal Dielectric Model

Technical Notes: Djordjevic-Sarkar Model Parameter Calculation

Technical Notes: Expressions for Permittivity and Conductivity

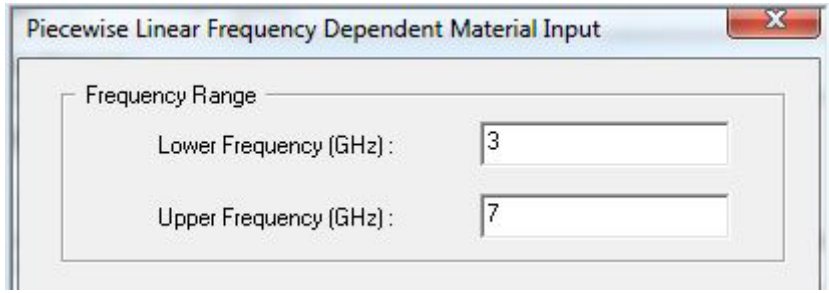
Technical Notes: Conductivity and Permittivity Limits

Modifying Datasets

Assigning Frequency Dependent Material: Piecewise Linear Input

Having selected Piecewise Linear Input as the model for the frequency dependent material property, you enter the required values in the dialog displayed:

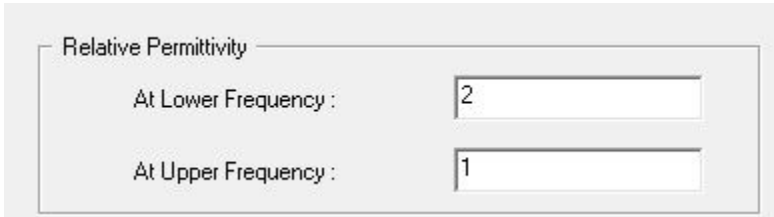
1. Enter Upper and Lower Frequency Range.



   HFSS assumes that the material's property values remain constant for frequencies below the Lower Frequency and above the Upper Frequency.

2. Enter the Upper and Lower Relative Permittivity of the material.

   If the permittivity of the material does not vary with frequency, enter the same value you entered for the permittivity's lower frequency.



3. Follow the same procedure for entering values for permeability, dielectric loss tangent, and magnetic loss tangent.

   The example of a high value for dk (low value for loss tangent) at low frequency and a lower value for dk (higher value for loss tangent) at some higher frequency is intended to roughly simulate how material should behave with reference to frequency for a causal model.

   After you have entered the data for your selection, you return to the **View/Edit Material** window. New default function names appear in the material property text boxes. HFSS automatically created a dataset for each material property. Based on a varying property's dataset, HFSS can interpolate the property's values at the desired frequencies during solution generation.

To modify the dataset with additional points, see Modifying Datasets.

Note Neither the piecewise or the loss models ask for frequency dependent conductivity because there the constant sigma represents the DC loss and the frequency dependent loss tangent represents the polarization losses.

Related Topics

Piecewise Linear Input

Debye Model Input

Multipole Debye Model Input

Djordjevic-Sarkar

Enter Frequency Dependent Data Points

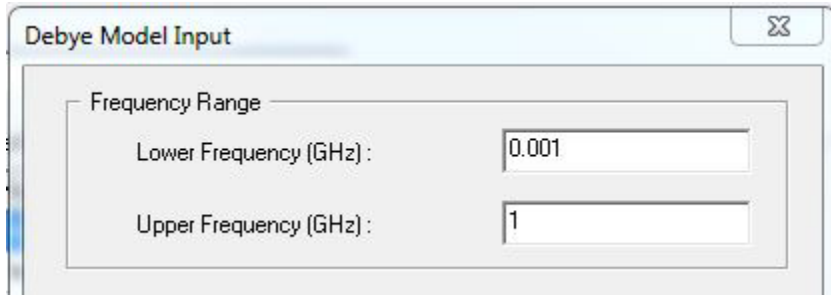
Technical Notes: Frequency-Dependent Material Properties

Modifying Datasets

Assigning Frequency Dependent Material: Debye Model Input

Having selected Debye as the model for the frequency dependent material property, you enter the required values in the dialog displayed:

1. Enter the Upper and Lower Frequencies to define the range:

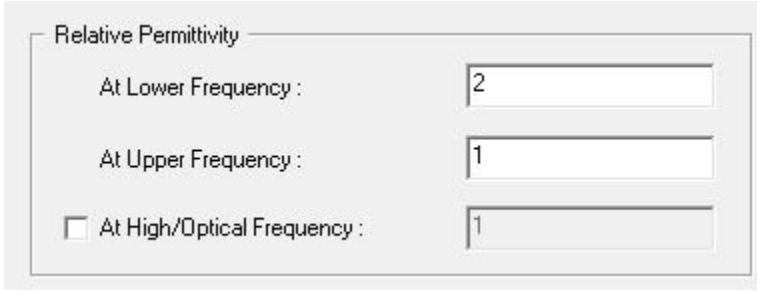


   HFSS assumes that the material's property values remain constant for frequencies below the Lower Frequency and above the Upper Frequency.

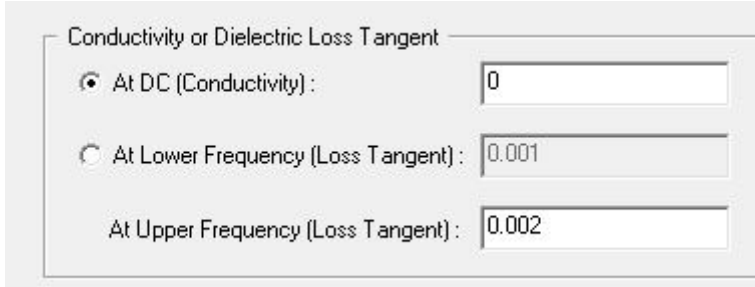
2. Enter the Upper and Lower Relative Permittivity of the material.

   If the permittivity of the material does not vary with frequency, enter the same value you

entered for the permittivity's lower frequency



3. If you need to specify a value for a High/Optical Frequency, check the box to enable the value field.
4. For Conductivity or Dielectric Loss Tangent, select the radio button for either At DC or for Upper and Lower Frequencies.



After you have entered the data for your selection, you return to the **View/Edit Material** window. New default function names appear in the material property text boxes. HFSS automatically created a dataset for each material property. Based on a varying property's dataset, HFSS can interpolate the property's values at the desired frequencies during solution generation.

To modify the dataset with additional points, see Modifying Datasets.

Note Neither the piecewise or the loss models ask for frequency dependent conductivity because there the constant sigma represents the DC loss and the frequency dependent loss tangent represents the polarization losses.

Related Topics

Piecewise Linear Input

Multipole Debye Model Input

Djordjevic-Sarkar

Enter Frequency Dependent Data Points

Technical Notes: Frequency-Dependent Material Properties

Technical Notes: Djordjevic-Sarkar Causal Dielectric Model

Technical Notes: Djordjevic-Sarkar Model Parameter Calculation

Technical Notes: Expressions for Permittivity and Conductivity

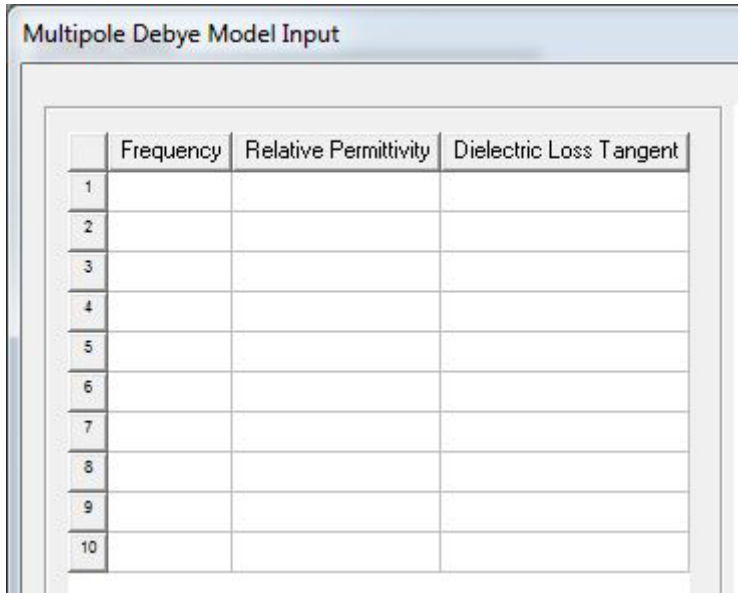
Technical Notes: Conductivity and Permittivity Limits

Modifying Datasets

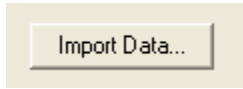
Assigning Frequency Dependent Material: Multipole Debye Model Input

Having selected Multipole Debye as the model for the frequency dependent material property, you enter the required values in the dialog displayed.

1. You can enter data either into the columns for Frequency (GHz), Relative Permittivity and Dielectric Loss Tangent The frequency can be ranged from 0 to unlimited. The relative permittivity is ranged from 1.0 to unlimited and loss tangent is ranged from 0 to unlimited. To get the good results, you should provide more than 5 frequency points.,



You can also import the data from a tab format text file.



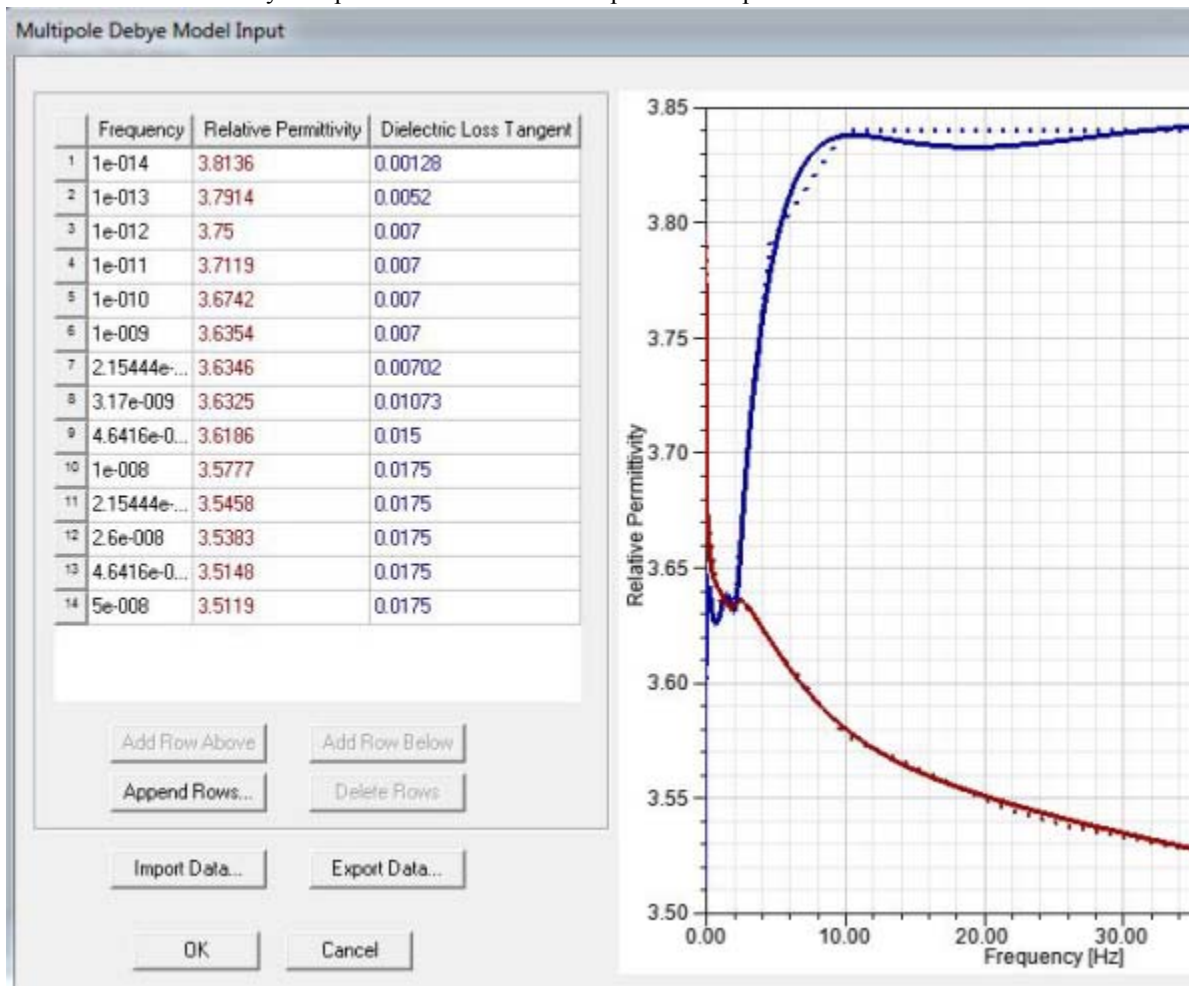
The import data will be shown in the data table, and then can be modified. You can export current input data to a file. The file is a tab file with extension .tab (same as dataset import/export file). Below is an example of the file format. Each row provides Frequency (assumed to be Hz), Permittivity, and Loss Tangent.

```
0.00001    3.8136   0.00128
0.00010    3.7914   0.00520
0.00100    3.7500   0.00700
```

```
0.01000   3.7119  0.00700
0.10000   3.6742  0.00700
1.00000   3.6354  0.00700
2.15444   3.6346  0.00702
3.17000   3.6325  0.01073
4.64160   3.6186  0.01500
10.0000   3.5777  0.01750
21.5444   3.5458  0.01750
26.0000   3.5383  0.01750
46.4160   3.5148  0.01750
50.0000   3.5119  0.01750
```

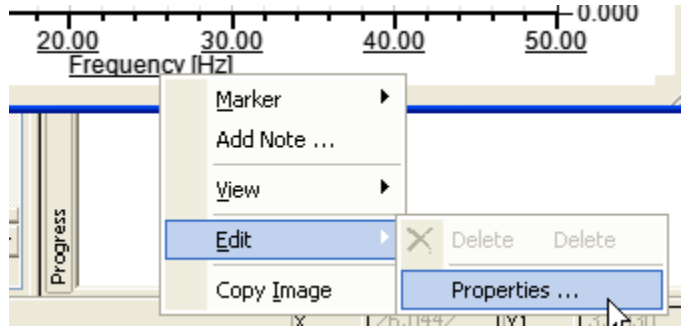
During the data input, the frequency dependent expressions for permittivity or loss tangent are calculated using Multipole Debye model, and are plotted for reference. The input data are also

linearly interpolated and drawn on the plot for comparison..

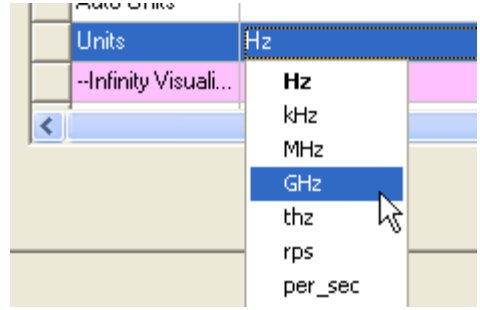


If desired, you can change or rescale the plot by selecting the Frequency label and right-click-

ing to display the Edit Properties menu.



Select the Scaling tab, find the Units row, and right click to select the desired scaling.



2. Based on this data the software dynamically generates frequency dependent expressions for relative permittivity and loss tangent through the Multipole Debye Model. The input dialog plots these expressions together with your input data through the linear interpolations.
   - The generated expressions provide the new value for the material properties of relative permittivity and loss tangent.
   - Both the expressions and data triples can be exported and re-imported.

   After you have entered the data for your selection, you return to the **View/Edit Material** window. New default function names appear in the material property text boxes. HFSS automatically created a dataset for each material property. Based on a varying property's dataset, HFSS can interpolate the property's values at the desired frequencies during solution generation.

Related Topics

Piecewise Linear Input

Debye Model Input

Djordjevic-Sarkar

Technical Notes: Multipole Debye Model

Technical Notes: Frequency-Dependent Material Properties

Technical Notes: Djordjevic-Sarkar Causal Dielectric Model

Technical Notes: Djordjevic-Sarkar Model Parameter Calculation

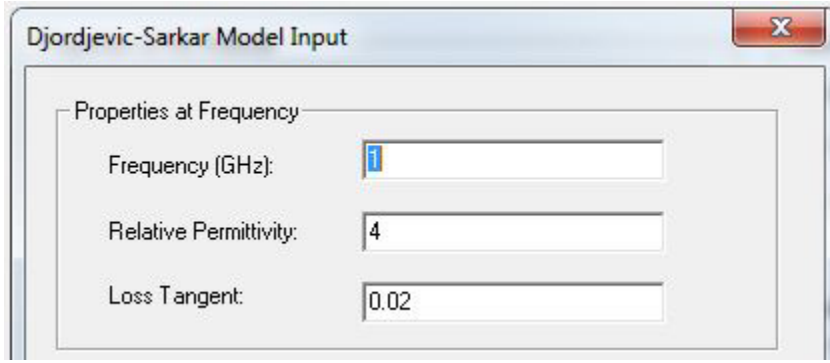
Technical Notes: Expressions for Permittivity and Conductivity

Technical Notes: Conductivity and Permittivity Limits

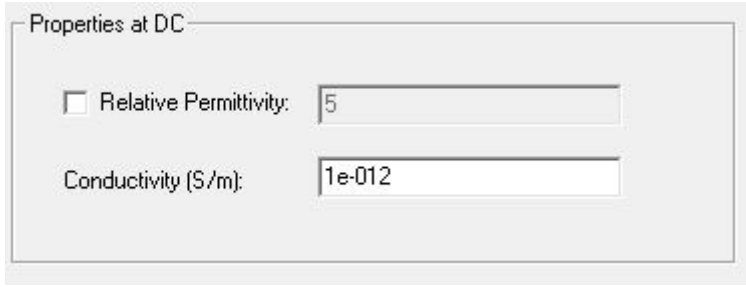
Assigning Frequency Dependent Material: Djordjevic-Sarkar Model Input

Having selected Djordjevic-Sarkar as the model for the frequency dependent material property, you enter the required values in the dialog displayed.

1. Enter the values for Properties at Frequency:



2. Enter the Properties at DC. This includes Conductivity and can include relative permittivity.



After you have entered the data for your selection, you return to the **View/Edit Material** window. New default function names appear in the material property text boxes. HFSS automatically created a dataset for each material property. Based on a varying property's dataset, HFSS can interpolate the property's values at the desired frequencies during solution generation.

Related Topics

Piecewise Linear Input

Debye Model Input

Multipole Debye Model Input

Technical Notes: Frequency-Dependent Material Properties

Technical Notes: Djordjevic-Sarkar Causal Dielectric Model

Technical Notes: Djordjevic-Sarkar Model Parameter Calculation

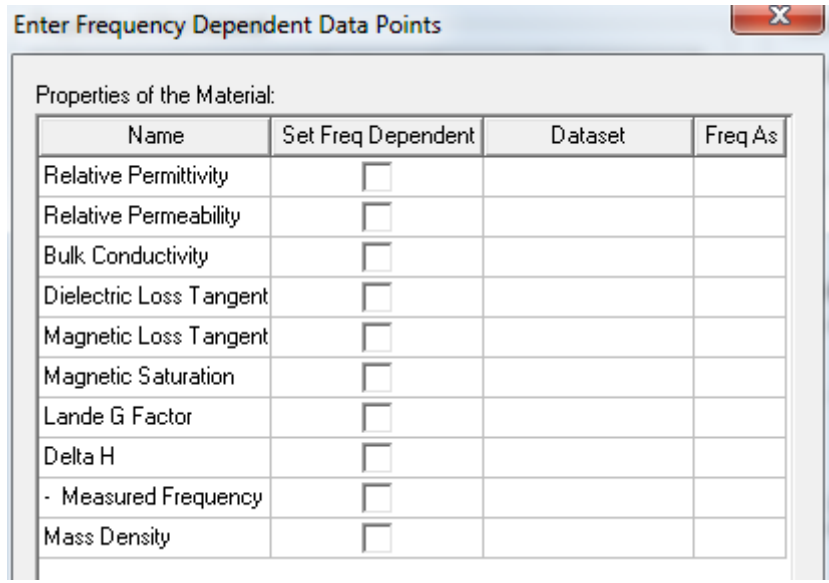
Technical Notes: Expressions for Permittivity and Conductivity

Technical Notes: Conductivity and Permittivity Limits

Modifying Datasets

Assigning Frequency Dependent Material: Enter Frequency Dependent Data Points

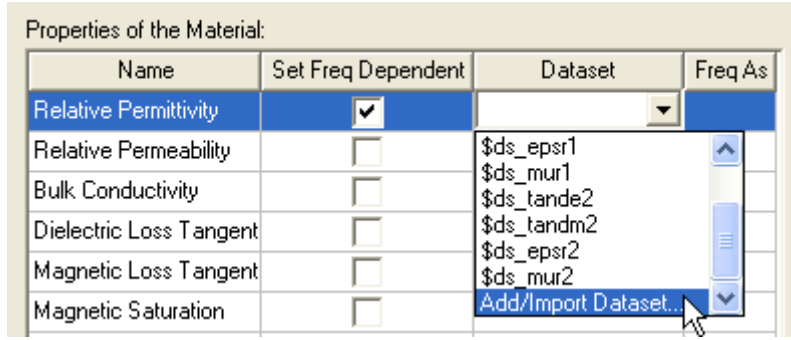
1. When you click **OK** on the on after selecting **Enter Frequency Dependent Data Points** on the **Frequency Dependent Material Setup** dialog box, the **Enter Frequency Dependent Data** points dialog box appears.



It shows a table with four columns:

- **Name**: the name of the selected material property.
- **Freq Dependent**: Check the box to indicate if the property is expressed as frequency-dependent dataset. If a property can not be set as frequency-dependent dataset, the cell is disabled.
- **Dataset column**: this is disabled unless **Freq Dependent** is checked or the property cannot be set as frequency dependent. When enabled, it contains a dropdown menu with a list

of existing datasets and the **Add/Import dataset..**.to add or import new dataset.



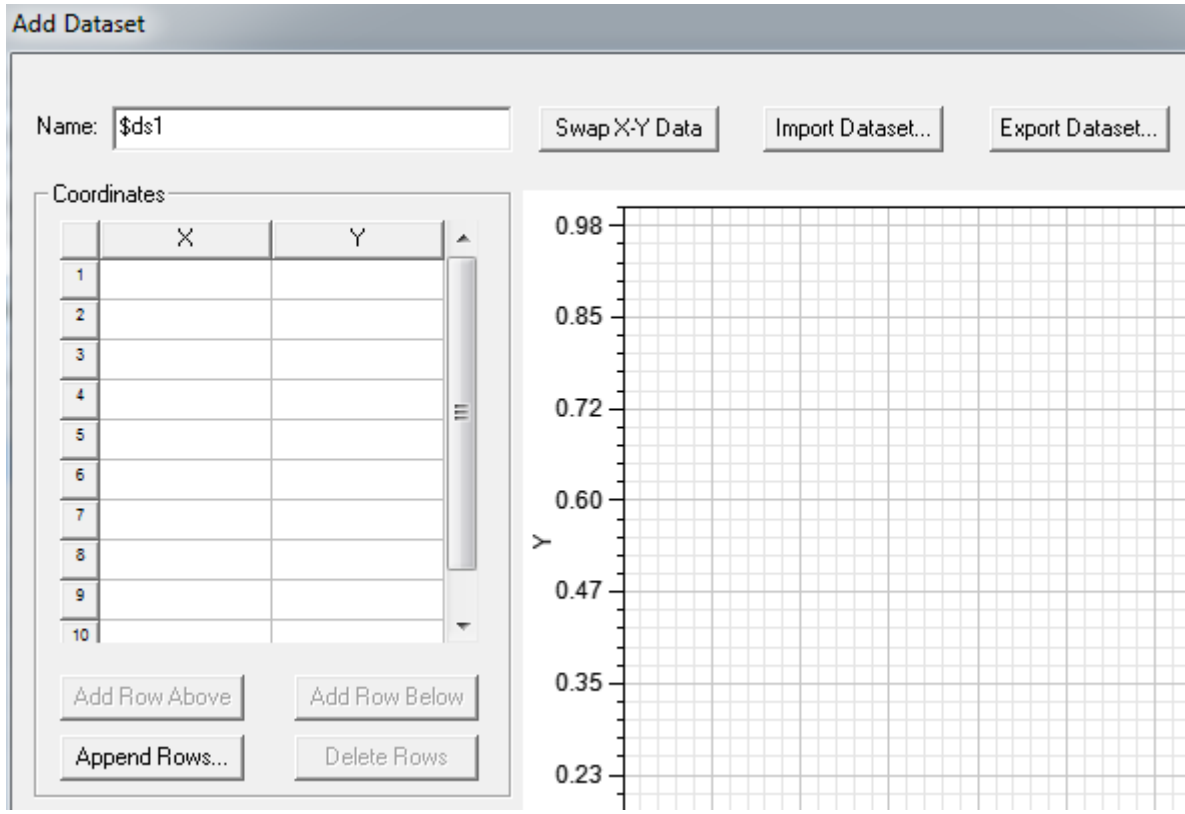
- **Freq As**: after a dataset is successfully imported or added, there are two choices available: "X datapoint" or "Y datapoint". The entered values are assumed to be in Hz.

2. If you select **Add/Import dataset**, the **Add Dataset** dialog appears.

   This contains the following fields:

   - The name field for the current dataset. The default is ds1.
   - The **Import from File** button. This opens a file browser for you to select an existing dataset.
   - The **Coordinates** table. This contains X and Y text fields in which you can enter data points where the frequency is assumed to be Hz. The values you add are interactively displayed on the graph to the right of the table. You can also Add rows above or below a

selected row, Delete rows, or Append a specified number rows.



3. After you have specified or imported the data points, and **OK** the dialog, the **Enter Frequency Data Points** dialog shows the **Dataset Name** and the **Freq As** Hzvalue.
4. After you **OK** the **Enter Frequency Dependent Data Points** dialog shows the new values.

   After you have entered the data for your selection, you return to the **View/Edit Material** window. New default function names appear in the material property text boxes. HFSS automatically created a dataset for each material property. Based on a varying property's dataset, HFSS can interpolate the property's values at the desired frequencies during solution generation.

   To modify the dataset with additional points, see Modifying Datasets.

Note Neither the piecewise or the loss models ask for frequency dependent conductivity because there the constant sigma represents the DC loss and the frequency dependent loss tangent represents the polarization losses.

Related Topics

Piecewise Linear Input

Debye Model Input

Multipole Debye Model Input

Djordjevic-Sarkar

Technical Notes: Frequency-Dependent Material Properties

Technical Notes: Djordjevic-Sarkar Causal Dielectric Model

Technical Notes: Djordjevic-Sarkar Model Parameter Calculation

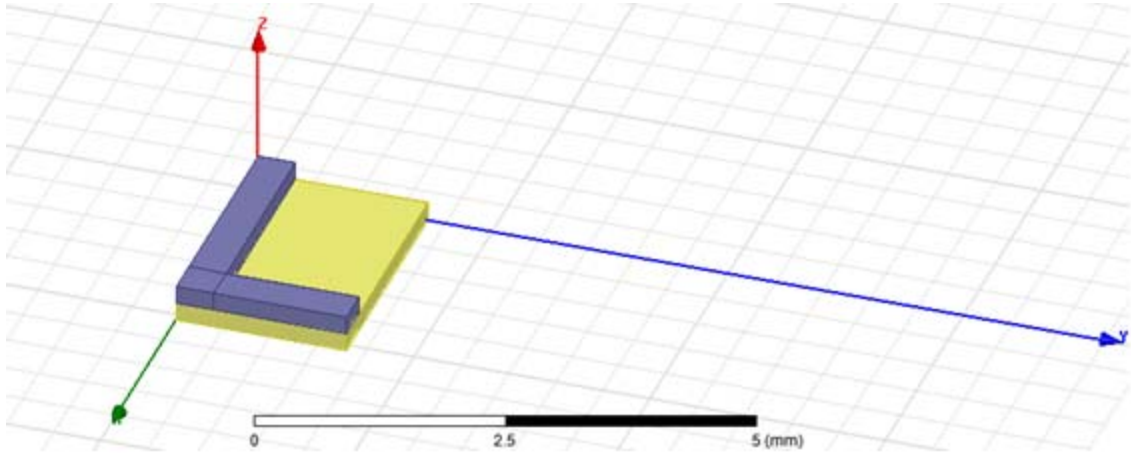
Technical Notes: Expressions for Permittivity and Conductivity

Technical Notes: Conductivity and Permittivity Limits

Modifying Datasets

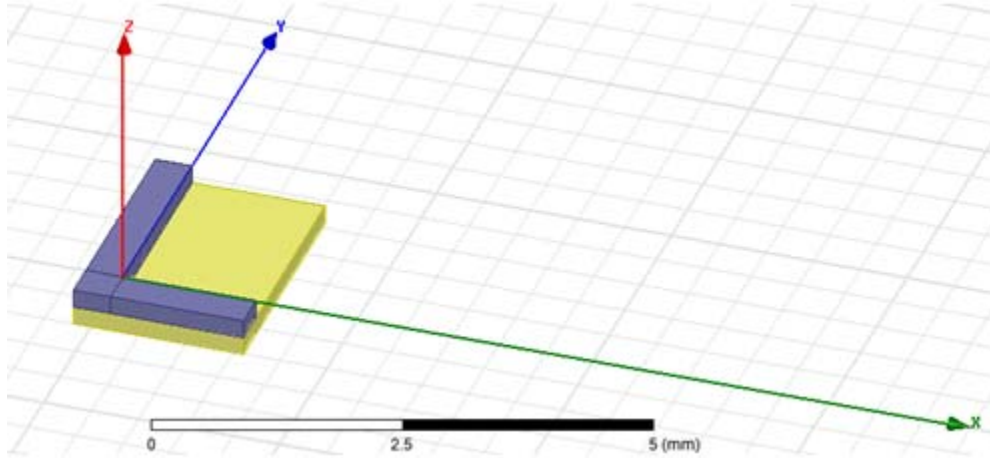
Spatially Dependent Materials in HFSS

HFSS supports spatially dependent materials, that is, materials whose properties change over their length (or width or thickness). Consider the L shaped strip lines in the figure below.



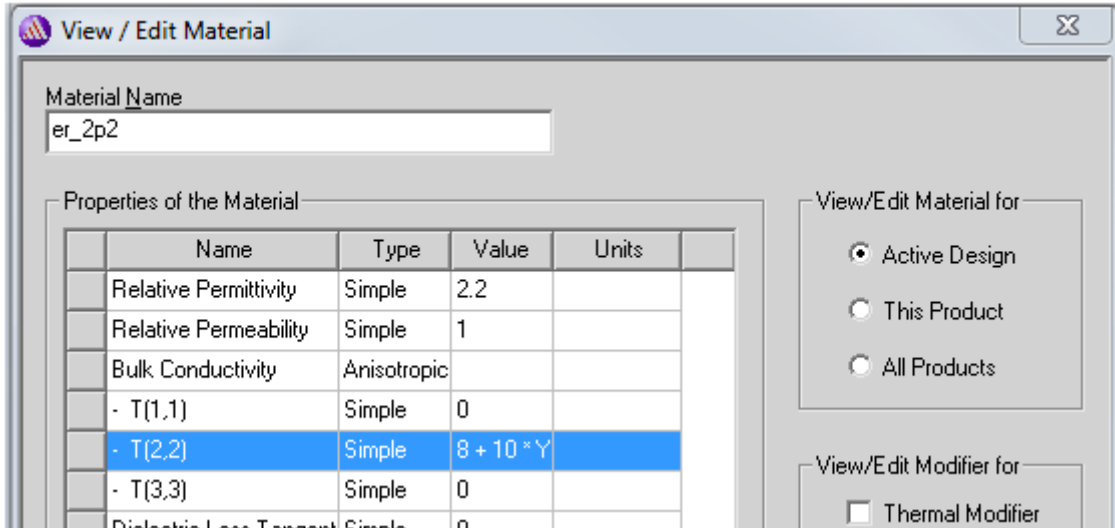
If the conductivity of the gray striplines changes over the length of the line, then you need to express the change as a function along one of the axes. However, in this case the two striplines are normal to each other. So, you need to use a different coordinate system for each of them or a different material.

Each spatially dependent object should have its own coordinate system as that would associate each with a separate origin point. The below images show the coordinate systems for each stripline with the Y origin at the start of the object.



So, the Bulk Conductivity could be something like 300 + 10 * Y to show increasing conductivity over the length of the object. Of course, the entire expression language is available so more complicated functions are usable as are project variables. Design variables are not supported at this time.

In the **View/Edit Material** dialog, you can assign properties using the Cartesian coordinate [X, Y, Z] to assign the spatial properties for a material.



Here the value of the Y dimension of the object composed of the material, based on that object's assigned coordinate system, affects the Bulk conductivity along the T[2,2] Tensor.

Note The spatial dependency can only be expressed in Cartesian coordinates.

Spatially dependent properties can still be either isotropic or anisotropic.

Spatially dependent properties will not be subject to solve time modifications like 'Automatic Causal Materials' transforms.

Related Topics

Spatially Dependent Boundaries in HFSS and HFSS-IE

Specifying Thermal Modifiers

To specify Thermal modifiers for a material:

1. In the View/ Edit materials dialog you must enable the View/ Edit Modifier checkbox for Thermal Modifier.

   This causes the Properties of the Material table to expand to include a Thermal Modifier column. By default, the Thermal Modifier property is set to **None**.

2. Selecting **Edit...** from the drop down menu, rather than **None** causes the **Edit Thermal Modifier** dialog to appear.

3. Select **Expression** radio button to display the **Parameters Modifier** text field or the **Quadratic** radio button to display the tabs for **Basic Coefficient Set** and **Advanced Coefficient Set**.

   - With **Expression** selected, you can write an equation for a thermal modifier in the Parameters **Modifier** text field.

     The expression whether to use Celsius or Kelvin is totally problem dependent. If a material thermal coefficient is defined as alpha/c_deg, then it is Celsius. On the other hand, if a material thermal coefficient is alpha/k_deg, then it is Kelvin.

   - Checking **Use temperature dependent data set** disables the Modifier text field. You can then use the drop down menu to select Add/Import Dataset. This lets you define the thermal modifier as a data set.

   - With the **Quadratic** radio button selected, in the **Basic Coefficient** tab, you can edit fields for the TempRef and units, and fields for C1 and C2 for the following equation:

   - P(Temp) = Pref[1+ C1(Temp - TempRef) + C2(Temp - TempRef)^2]

   - where the Pref is defined as the reference relative permittivity.

   - With the **Quadratic** radio button selected, in the **Advanced Coefficient Set** tab, you can edit fields for lower and upper temperature limits (TL and TU respectively) and select their units from the drop down.

   You can also edit the constant value limit for the thermal modifier values outside the limits. By default, these are automatically calculated. Uncheck the Auto Calculate TML and TMU to specify new values for thermal modifier lower (TML) and thermal modifier upper (TMU).

4. Click **OK** to accept the edits and return to the View/ Edit materials dialog.

Related Topics

Adding Datasets

View/ Edit materials dialog

Defining Material Properties as Expressions

When defining or modifying a material's properties, each material property value in the **View/Edit Material** window can be assigned a mathematical expression. Simply type the expression in the appropriate **Value** box. Expressions typically contain intrinsic functions, such as sin(x), and arithmetic operators, such as +, -, *, and /, but do not include project variables.

Note By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

Related Topics

Defining Variable Material Properties

Technical Notes: Expressions for Permittivity and Conductivity

Technical Notes: Conductivity and Permittivity Limits

Defining Functional Material Properties

Any material property that can be specified by entering a constant can also be specified using a mathematical function. This is useful when you are defining a material property whose value is given by a mathematical relationship — for instance, one relating it to frequency or another property's value. When defining or modifying a material's properties, simply type the name of the function in the appropriate **Value** box.

Note By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

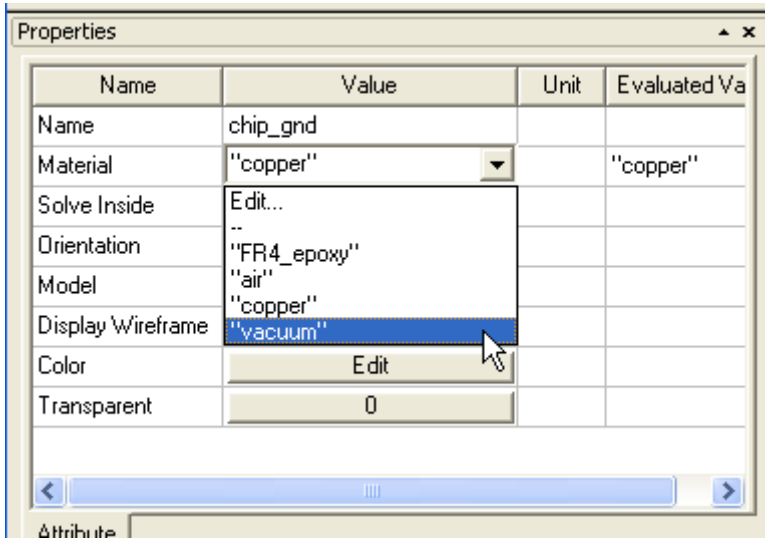
Related Topics

Defining Mathematical Functions

Defining Variable Material Properties

Assigning Materials from the Object Properties Window

The Properties dialog for each object includes a materials property. If you click on the current material property you see a drop down list that includes and Edit command and a list of materials in the current project. You can select from the list of current materials to assign the selected material to that object.



Related Topics

Assigning Materials

Viewing and Modifying Material Attributes

1. In the **Select Definition** window, select the material you want to view or modify, and then click **View/Edit Materials**.

   The **View/Edit Material** window appears. The material name and its property values are listed.

Note By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

2. Under **View/Edit Material for**, select:
   - **Active Design** to display properties used in the active design.
   - **This Product** to display properties commonly used by this product.
   - **All Products** to display all properties available. This enlarges the table of properties to show all properties possible. You can use the scroll bars or size the dialog to see all properties.
3. You can modify the material as follows:
   a. Provide a new name for the material in the **Material Name** text box.
   b. Under **Type**, specify whether a material property is **Simple, Anisotropic**, **Vector** and **Vector Mag**, or for Relative Permeability, **Non-linear**, as required for that property.

      For **Simple**, you provide a value or variable.

      For **Anisotropic**, you provide tensor values.

      For **Vector**, you provide a **Vector Mag**.

      For **Non-Linear**, you provide a Data Set.
   c. Provide new material property values in the **Value** boxes.
   d. Change the units for a material property.

   Note: Materials stored in the global material library cannot be modified.
4. Click **OK** to save the changes and return to the **Select Definition** window.

Warning If you modify a material that is assigned in the active project after generating a solution, the solution will be invalid.

Related Topics

Validating Materials

Defining Variable Material Properties

Copying Materials

Removing Materials

Export Materials to a Library

Validating Materials

HFSS can validate a material's property parameters. For example, it will check if the range of values specified for each material property is reasonable.

If a material's property parameters are invalid, an error message will appear in the lower-right corner of the **View/Edit Material** window. If the parameters are valid, a green check mark will appear there.

To validate the material attributes listed in the **View/Edit Material** window:

- Select a name from the **Properties of the Material** list, and then click **Validate Material**.

Related Topics

Copying Materials

Removing Materials

Export Materials to a Library

Sorting Materials

Filtering Materials

Working with Materials Libraries

Copying Materials

1. In the **Select Definition** window, select the material you want to copy, and then click **Clone Material**.
2. To modify the material's attributes, follow the directions for modifying materials.
3. Click **OK** to save the copy in the active project's material library.

Related Topics

Validating Materials

Copying Materials

Removing Materials

Export Materials to a Library

Sorting Materials

Filtering Materials

Working with Materials Libraries

Removing Materials

1. In the **Select Definition** window, select a material you want to remove from the active project's material library.
2. Click **Remove Material**.

   The material is deleted from the project material library.

Note

The following materials cannot be deleted:

- Materials stored in the global material library.
- Materials that have been assigned to objects in the active project.

In a project library, you may want to use the **Project>Remove Unused Definitions** command to remove selected materials definitions that your project does not require.

Related Topics

Validating Materials

Copying Materials

Export Materials to a Library

Sorting Materials

Filtering Materials

Working with Materials Libraries

Exporting Materials to a Library

1. In the **Select Definition** window, select the material you want to export.
2. Click **Export Material to Library**.

   The **Export to material library** file browser appears.
3. Click **PersonalLib** to export the material to a local project directory, accessible only to the user that created it.

   Click **UserLib** to export the material to a library that is shared by more than one user, usually in a central location.
4. Type the library's file name and then click **Save**.

Related Topics

Validating Materials

Copying Materials

Removing Materials

Sorting Materials

Filtering Materials

Working with Materials Libraries

Sorting Materials

You can change the order of the materials listed in the **Select Definition** window. You can sort the list of materials by name, library location, or material property value.

To change the order of the listed materials:

- Click the column heading by which you want to order the materials.

  If the arrow in the column heading points up, the material data will be listed in ascending order (1 to 9, A to Z) based on the values in the column you chose. If you want the material data to be listed in descending order (9 to 1, Z to A), click the column heading again. The arrow will point down.

Note By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

Related Topics

Validating Materials

Copying Materials

Removing Materials

Export Materials to a Library

Filtering Materials

Working with Materials Libraries

Filtering Materials

If you want to remove certain materials or material properties from the list in the **Select Definition** window, use the filter options under the **Material Filters** tab. You can filter out materials based on the product or library with which they are associated. You can also filter out material properties and types of material properties. And you can remove the filtering in order to see all available material properties.

To filter materials or material properties listed in the **Select Definition** window, using the choices in the **Materials** tab:

1. The text field under Libraries lists the libraries for the project. Selecting the listed library highlights it and cause the table to display the materials in that library.
2. Above the Libraries area, you can check or uncheck boxes to show or hide Project Definitions and All Libraries.
   - With both unchecked, nothing appears in the materials table. With both checked, the table shows all materials and highlights those used in the project.
   - With only Project Definitions checked, the materials table shows only the materials used in the project.
   - With All Libraries checked, the table displays all materials, but may not show all available properties.

To filter out or show additional material properties in the **Materials** tab:

1. Click the **Material Filters** tab.
2. Under **Filter Materials and Properties for**, select one of the radio buttons:
   - Click **Select All** to select all of the products listed. Click **Clear** to clear all product selections.
   - **Active Design** to display properties used in the active design.
   - **This Product** to display properties commonly used by this product.
   - **All Products** to display all properties available. Selecting this enlarges the table of properties shown under the **Materials** tab to show all properties possible. You can use the scroll bars or size the dialog to see all properties.
3. Click the **Materials** tab to save your selections.

   Click **Cancel** to revert back to the last saved selections.

Related Topics

Validating Materials

Copying Materials

Removing Materials

Export Materials to a Library

Sorting Materials

Working with Materials Libraries

Working with Material Libraries

There are two different kinds of materials libraries in HFSS, a system library and a user library.

Related Topics

Editing Libraries

Configuring Libraries

Working with the System Material Library

HFSS provides you with a global, *or system library* of predefined materials. Global materials in the system library are available in every HFSS project. They cannot be modified.

You can create a global system library that is stored in a common location and available to multiple users.

Working with User Material Libraries

You can create your own personalized global material library, or *user library*, that can be used in any HFSS project only by the user that created it. User-defined global materials can be modified at any time.

You can also create a personalized local user library that is used only in the active HFSS project.

Editing Libraries

There are two different methods of editing libraries.

- Using right-click on Materials in the project window to display the **Edit All Libraries** shortcut menu. Clicking displays the **Edit Libraries** window.

  Editing definitions from the project window does not modify the configured libraries for any particular design, since this is editing in general.
- Using **Tools>Edit Configured Libraries>Materials** from the menu bar takes the current design into account and adds any new libraries to the configured list for the design.

Configuring Libraries

Use **Tools>Configure Libraries** to display the **Configure Design Libraries** window. From this window you can view the available libraries for System, User, and Project, and which of these libraries has been configured. Set of selection arrows allows you to move a highlighted library to the Configured list. A checkbox permits you to specify a configured library as default.

Related Topics

Exporting Materials to a Library.

12'

Modifying the Model View

You can modify the view of contents in the **3D Modeler** window without changing their actual dimensions or positions.

What do you want to do?

Change the view

Show or hide design objects

Change object visualization

Change the background

Related Topics

Assigning Color to an Object

Assigning Transparency to an Object

Change the view

- Change the model view with Alt+double-click on areas in the modeler window.
- Apply an Orientation to the Current View
- Set the View Options
- Rotate the view.
- Pan the view.
- Zoom in or out
- Zoom to Selected Excitation
- Zoom in or out using Alt-Mousewheel
- Fit contents in the view window.
- Use Clip Planes

Related Topics

Modifying the Model View

Assigning Color to an Object

Assigning Transparency to an Object

Show or hide design objects

- Show or hide objects.
- Show or hide boundaries or excitations.
- Visualize or Select Nets

Related Topics

Modifying the Model View

Assigning Color to an Object

Assigning Transparency to an Object

Change object visualization

- Render objects as wireframes, flat-shaded, or smooth-shaded solids.
- Set the Surface Visualization
- Modify the view orientation.
- Modify the lighting.
- Set the projection view.

Related Topics

Modifying the Model View

Assigning Color to an Object

Assigning Transparency to an Object

Change the background

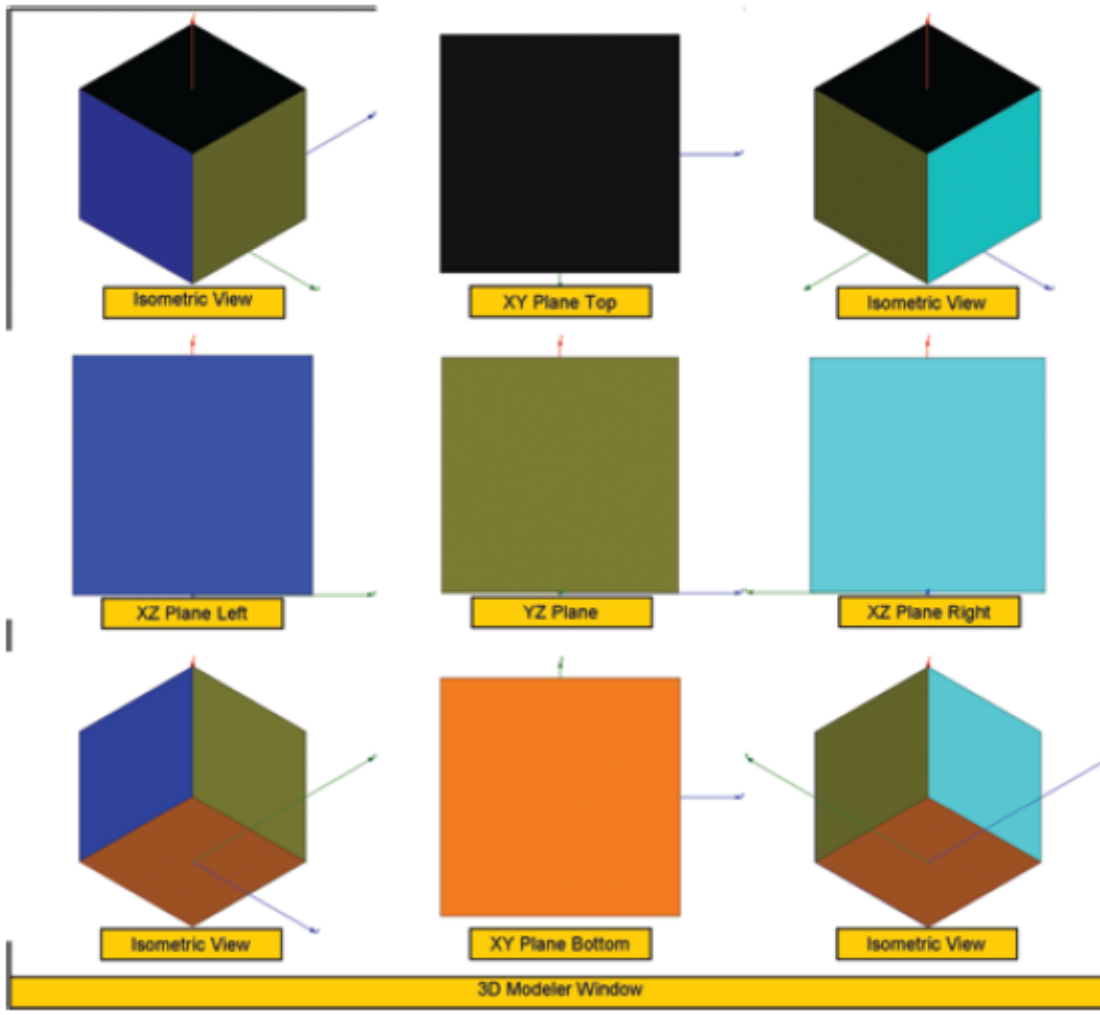
- Set the background color.
- Modify the appearance of the coordinate system axes.
- Modify the appearance of the grid.

Related Topics

Modifying the Model View

Changing the Model View with Alt+Double Click Areas

In the following figure, the orientation of the colored boxes shows the nine model orientations you can obtain by using **Alt + double click** in the corresponding area of the Modeler window.

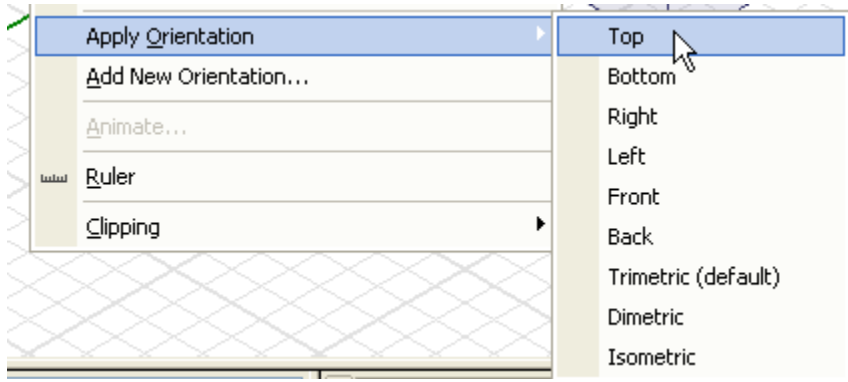


Related Topics

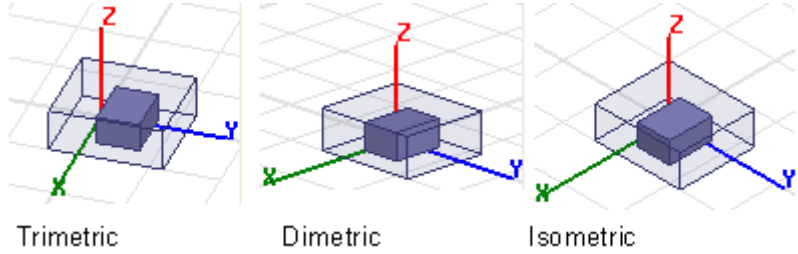
Rotate the view.

Apply an Orientation to the Current View

Right-click in the Modeler window to see the **View** shortcut menu with the **Apply Orientation** command.

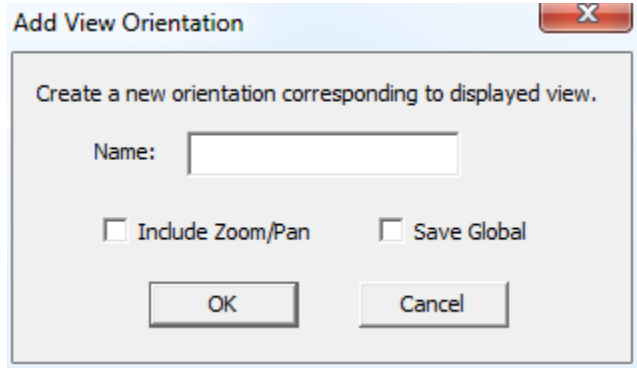


Possible orientations are Top, Bottom, Right, Left, Front, Back, Trimetric, Dimetric, and Isometric. These last three differ as shown below.



After applying an orientation, you can **View>Undo View:Orientation**.

You can also use the right-click **View** shortcut menu to **Add New Orientation**.



The Name that you assign will appear on the Apply Orientation command menu. You can specify whether to Include Zoom/Pan and whether to Save Global.

Related Topics

Change the model view with Alt-double click on areas in the modeler window.

View Options: 3D UI Options

Use the **View>Options** command to open the **3D UI Options** dialog. This lets you set defaults for the following view options:

- Stereo Mode (default, disabled)
- Drag Optimization (default, disabled)
- Show Ansoft Logo in Prints (default, disabled)
- Default Color Key Height (the maximum number of values displayed)

Where there is a selection option:

- Selection always visible (default, enabled)
- Set transparency of selected objects
- Set transparency of non-selected objects.

Default screen rotation about

- Model Center
- Current Axis
- Screen Center (default)

When you select one of these as the default, the **View>Rotate** selection menu changes to show that the Alt-Drag combination attaches to your selection.

Related Topics

Modeler options

Rotate the view.

Rotating the View

You can rotate the view relative to the Model Center, the Screen Center, or the Current Axis.

When you select one of these as the View>Options default, the **View>Rotate** selection menu changes to show that the Alt-Drag combination attaches to your selection.

To rotate the model center from the menus or icons:

1. Click **View>Interaction>Rotate>Rotate Model Center** or click the  . icon on the tool-bar.
2. Drag the mouse in the direction you want to rotate the view.

   The view rotates until you release the mouse button.
3. To exit **Rotate** mode, click **View>Interaction>Rotate>Rotate Model Center** again, or click the icon again or press **ESC**.

Hint Alternatively, rotate the view using one of the following methods:

- Hold down the **ALT** key as you drag the mouse.
- Right-click in the view window, and then click **View>Rotate** on the shortcut menu.

To rotate the view around the current axis:

1. Click **View>Interaction>Rotate>Rotate Current Axis**.
   - Alternatively, right-click in the view window, and then click **View>Interaction>Spin**.
   - Or, click the spin icon on the toolbar 
2. Drag the mouse left or right at the speed you want to spin the view.

   The view spins continually in the direction and at the speed you dragged the mouse.
3. To stop spinning the view, click in the view window.
4. To end **Spin** mode, click **Spin** again on the **View** menu or press **ESC**.

To rotate the view around the screen center:

1. Click **View>Interaction>Rotate>Rotate Screen Center** or click the rotate icon  on the toolbar.
2. Drag the mouse up and down at the speed you want to rotate the view.
3. To end **Rotate** mode, click **View>Interaction>Rotate>Rotate Screen Center**, or click the icon again or press **ESC**.

Note If Rotating is slow, especially for complex models, for some graphics cards, you can improve performance by setting **NVIDIA Control Panel>3D Settings>Manage 3D Settings Global Settings>Global Presets: Workstation App - Dynamic Streaming**

Related Topics

Pan the view.

Change the model view with Alt-double click on areas in the modeler window.

Panning the View

To move (pan) the view:

1. Click **View>Interaction>Pan**  .
2. Drag the mouse in the direction you want to pan the view.
   The view will pan until you release the mouse button.
3. To exit **Pan** mode, click **Pan** on the **View** menu again or press **ESC**.

Hint Alternatively, pan the view using one of the following methods:

- Hold down the **SHIFT** key as you drag the mouse.
- Right-click in the view window, and then click **View>Pan** on the shortcut menu.

Zooming In and Out

You can magnify (zoom in) or shrink (zoom out) the contents in the view window using hot keys or mouse zoom mode.

To zoom in using hotkeys:

- Press the plus sign (+) or (=) keys or press Ctrl-E keys.

  The view zooms in 5 percent.

To zoom out using hotkeys:

- Press the minus sign (-) key or press the Ctrl-F keys.

  The view zooms out 5 percent.

To zoom using the mouse.

1. Click **View>Interaction>Zoom**  or on the shortcut menu, **View>Zoom**.
2. To zoom in, drag the mouse towards the top of the view window. The objects in view expand as you drag.

   To zoom out, drag the mouse towards the bottom of the view window. The objects in view decrease in size as you drag.

   When zooming on a view of model objects the absolute size of the model does not change.

   When zooming on a 2D report, axis labels and ticks will adjust automatically during the zoom operation and will rescale to their final value after the zoom operation is complete.
3. To end **Zoom** mode, click **View>Interaction>Zoom** again or press **ESC**.

You can also zoom in or out using a mouse wheel.

Hint Alternatively, zoom in or out on the view using one of the following methods:

- Hold down the **ALT**+**SHIFT** keys as you drag the mouse.
- Right-click in the view window, and then click **View>Zoom** on the shortcut menu.

Note If Zooming is slow, especially for complex models, for some graphics cards, you can improve performance by setting **NVIDIA Control Panel>3D Settings>Manage 3D Settings Global Settings>Global Presets: Workstation App - Dynamic Streaming**

Related Topics

Zooming In or Out on a Rectangular Area

Zooming In or Out Using a Mouse wheel

Fitting Objects in the View Window

Zoom to Selected Exciation

Zooming In or Out on a Rectangular Area

To magnify or shrink a specific rectangular area in the view window:

1. Click **View>Interaction>Zoom In**  or **View>Interaction>Zoom Out**  .
   - Alternatively, right-click in the view window, and then click **View>Zoom In** or **View>Zoom Out** on the shortcut menu.
2. Use the mouse to draw a rectangle (or square) by selecting two diagonally opposite corners.

   This is the area of magnification that will be increased or decreased.

   The rectangular area is magnified or decreases in size.

   When zooming on a view of model objects, the absolute size of the model does not change.

   When zooming on a 2D report, axis labels and ticks will adjust after the zoom operation is complete.
3. To end **Zoom** mode, click **View>Interaction>Zoom In** or **View>Interaction>Zoom Out** or press **ESC**.

Related Topics

Zooming In and Out

Fitting Objects in the View Window

Zooming In or Out using a Mousewheel

By clicking in the Modeler window you can use the mousewheel to zoom in and out of the 3D Modeler window. The cursor location becomes the center relative to the zoom, rolling the wheel forward increases the zoom, and rolling the wheel backwards decreases the zoom. If the window includes a ruler display, this adjusts as you scroll in and out.

Note If Zooming is slow, especially for complex models, for some graphics cards, you can improve performance by setting **NVIDIA Control Panel>3D Settings>Manage 3D Settings Global Settings>Global Presets: Workstation App - Dynamic Streaming**

Related Topics

Using the Mousewheel to Scroll in the History Tree

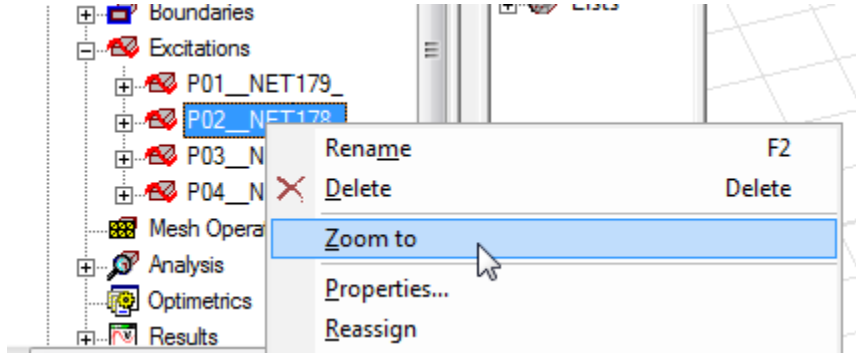
Zooming and Fitting Reports

Zoom to Selected Excitation

Zoom to is an option that can enlarge the view of an excitation. This can be very useful if you want to inspect any problem areas.

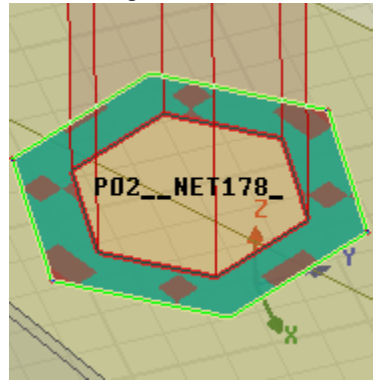
You can use the **Zoom to** option as follows.

1. Select the port under **Excitations** in the project tree to highlight it in the geometry.
2. Right-click to generate the short-cut menu as shown below.



3. Select the option **Zoom to**.

Notice that the option instantly increases magnification of the surface excited with the wave port as shown in the Figure below.



Related Topics

Modifying the Model View

Fitting Objects in the View Window

What do you want to do?

- Fit all objects or traces in a view window.
- Fit selected objects in a view window.

Fitting All Objects in a View Window

To fit all the views: click **View>Fit All>All Views**.

All view windows displaying the active design change to include all model objects.

To fit only the active view: click **View>Fit All>Active View**.

The view in the active Modeler window changes to include all model objects.

Hint Alternatively, fit all objects in the *active* view window using one of the following methods:

- Press **CTRL**+**D**.
- Right-click in the view window, and then click **View>Fit All** on the shortcut menu.

When **Fit All** is used in a report view, the window is automatically rescaled to fit all traces in the window and the axis label and ticks are rescaled.

Related Topics

Fitting a Selection in a View Window

Fitting a Selection in a View Window

1. When you are working on a model view, select the objects you want to fit in the view. When you are working on a report, select the traces you want to fit.
2. To fit the selection in the active view window: **Click View>Fit Selection>Active View**.
3. To fit the selection in every open view window of the active design: **Click View>Fit Selection>All Views.**

Hint Alternatively, fit the selection in the active view window by clicking **View>Fit Selection** on the shortcut menu.

Related Topics

Fitting All Objects in a View Window

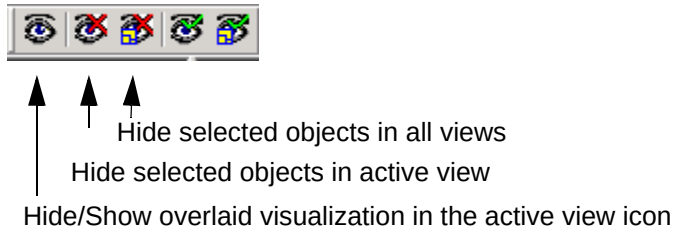
Hiding Objects from View

To hide selected objects.

1. Select the object you want to hide from view.
2. Click **View>Visibility>Hide Selection** and one of the following commands:
   - **All Views** to hide the selected object in every open view window.
   - **Active View** to hide the selected object in the active view window.

   You can also right-click in the modeler window for a short-cut menu and use the **View** commands for **Hide Selection**, or **Show Only Selection**.

   You can also use the **Hide** icons in the toolbar to hide selected objects in all views or the active view.



   The objects you selected are hidden.

If there are many objects, you may find it easier to hide objects using the **Active View Visibility** dialog.

Note Hiding boundaries also turns off a check for boundary overlaps during boundary assignment. In the case of very large models with many boundaries, hiding boundaries can prevent delays during boundary assignment. Full model validation will subsequently check for boundary overlaps.

Object visibility is saved with the project.

Related Topics

Showing Objects

Active View Visibility

Showing Objects

To show *one or more objects* that are currently hidden:

1. Click **View>Visibility>Active View Visibility**  .

   The **Active View Visibility** dialog box appears.
2. Select the tab for the objects you want to show or hide. The dialog contains tabs for 3D Modeler objects, Color Key objects, Boundaries, Excitations, and Fields Reporter objects.
3. Under the tab you need, select the **Visibility** option for the objects you want to show in the active view window.
   - For designs with large numbers of objects, you can resize the dialog for easier selection.
   - By default, objects are listed in alphabetical order. You can invert the order by clicking the Name bar above the Name fields. A triangle in the bar indicates the direction of the listing.
   - You can also use the **Name** field to type in an object name and apply the visibility via the **Show** and **Hide** buttons.

   The objects you select and designate as Visible (by selecting the property or using Show) reappear.

Note Hiding boundaries also turns off a check for boundary overlaps during boundary assignment. In the case of very large models with many boundaries, hiding boundaries can prevent delays during boundary assignment. Full model validation will subsequently check for boundary overlaps.

To show *all objects* that are currently hidden:

1. Click **View>Visibility>Show All**. and one of the following commands:
   - **All Views** to show all objects in every open view window
   - **Active Views** to show all objects in the active view window.

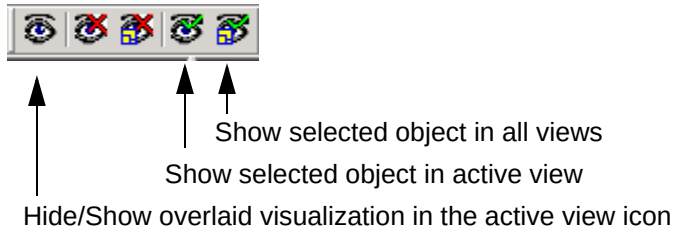
   The selected objects reappear.

To show *selected objects* that are currently hidden:

1. Select the object. Hidden items are selected once the node corresponding to them is clicked in the history tree.
2. Click **View>Visibility>Show Selection** and one of the following.
   - **All Views** to show selected objects in every open view window
   - **Active Views** to show selected objects in the active view window.

   You can also use the toolbar icons to **Show selected objects in all views** and **Show selected**

objects in active views.



The selected objects reappear.

Object visibility is saved with the project.

Related Topics

Hiding Objects

Active View Visibility

Active View Visibility Dialog

If there are many objects, it may be easier to show or hide objects using the **Active View Visibility** dialog

1. Click **View>Visibility>Active View Visibility**, or click the **Hide/Show** icon on the menu bar.

   The **Active View Visibility** dialog box appears.

2. Select the tab for the objects you want to show or hide. The dialog contains tabs for 3D Modeler objects, Color Key objects, Boundaries, Excitations, and Fields Reporter objects.
   - For designs with large numbers of objects, you can resize the dialog for easier selection.
   - By default, objects are listed in alphabetical order. You can invert the order by clicking the Name bar above the Name fields. A triangle in the bar indicates the direction of the listing.
   - You can also use the **Name** field to type in an object name and apply the visibility via the **Show** and **Hide** buttons.

3. Under the tab, clear the **Visibility** option for the objects you want to hide in the active view window.

   The objects you designate are hidden.

Note Hiding boundaries also turns off a check for boundary overlaps during boundary assignment. In the case of very large models with many boundaries, hiding boundaries can prevent delays during boundary assignment. Full model validation will subsequently check for boundary overlaps.

Object visibility is saved with the project.

Related Topics

Showing Objects

Hiding Objects

Rendering Objects as Wireframes or Solids

To render (display) *all* objects in the view window as wireframe outlines, flat-shaded solids, or smooth-shaded solids:

1. Click **View**>**Render** and click one of the following:
   - **Wireframe**.

     The objects in the view window are displayed as skeletal structures, enabling you to see all sides of the objects at one time.

     You can also use the **F6** key or the shade icon  to toggle the display to wireframe.
   - **Smooth Shaded**.

     The objects in the view window are displayed as shaded objects with smooth edges.

     You can also use the **F7** key or the shade icon  to toggle the display to smooth shaded.

To render *a single* object in the view window as a *wireframe outline*:

1. Select the object you want to render as a wireframe:
2. In the **Properties** dialog box, under the **Attribute** tab, select **Display Wireframe**.

Hint You can also press **F6** (Wire Frame) and **F7** (Smooth Shaded) to toggle between these two views.

Related Topics

Setting the Default View Rendering Mode

Setting the Surface Visualization

Setting the Default View Rendering Mode

To set a default rendering mode for all objects created in the active design and in future designs:

1. Click **Tools**>**Options**>**Modeler Options**.
2. Click the **Display** tab.
3. Select one of the following from the **Default View Render Mode** pull-down list.
   - **Wireframe**.

     The objects in the view window will be displayed as skeletal structures, enabling you to see all sides of the objects at one time.
   - **Smooth Shaded**.

     The objects in the view window will be displayed as shaded objects with smooth edges.
4. Click **OK.**

   The rendering mode will be applied to all new objects you create.

Related Topics

Rendering Objects as Wireframes or Solids

Setting the Curved Surface Visualization

HFSS allows you to specify the faceting for rendering true curves by using the **View>Render>Curved Object Visualization** command. There are two options for control--Maximum surface deviation and Maximum normal deviation. This resembles the Mesh surface approximation settings. Reduce either or both of the allowed deviations to improve the image quality. Improved image quality comes at the cost of increased CPU consumption. Changes apply to the current model until they are changed again. Any changes reset the default.

This release moves to an ACIS R21 faceting algorithm and has new defaults:

- Relative 0.002mm surface deviation and 30 degree normal deviation for 3D design
- Relative 0.002mm surface deviation and 2 degree normal deviation for 2D design

Legacy projects with non default settings are approximately converted to new settings with warning. Overall faceting time may be around 20% more but number of triangles should reduce.

The default gives satisfactory results (i.e. cpu/memory consumption vs. graphical display) for various model complexities.

When you change Curved Object Visualization settings and apply them to a design, those settings are saved with design unless you change it again. That means when you open the design again, it will apply saved visualization settings and NOT the default settings. Because this affects the CPU and memory required to open the project, typically, you should not save a project with other than the default settings.

To set the Curved Object Visualization for the active modeler window:

1. Click **View>Render>Curved Object Visualization**.

   This command displays the **Curved Object Visualization** dialog for the active modeler window. The dialog contains areas for setting the Maximum Deviation, and the Maximum Normal deviation.

2. Set the Maximum Deviation by first selecting from the radio buttons for **Ignore**, set as **Relative Deviation** or set as **Absolute Deviation**. Selecting the later two radio buttons enables the value field.

   When set as **Relative Deviation**, the actual surface deviation depends on the model size. For example, sphere with a radius of 10 has same number of facets as a sphere with a radius of 1. This means that CPU cost does not increase based on the model dimension.

   When set as **Absolute Deviation**, the maximum surface deviation for both the spheres will be approximately same since a bigger sphere has more facets than a smaller one. This means that the most CPU cost applies to the larger objects.

3. If you selected the radio buttons for Relative or Absolute Deviation for Maximum Deviation, enter a value in the field.

4. To change the **Maximum Normal Deviation**, enter a value in the text field. Units are degrees.

Note Wire bodies cannot be rendered with a Maximum Normal Deviation value less than 1 degree. When using a setting less than 1 degree all wire bodies will be rendered with a setting of 1 degree and all closed bodies will be rendered with the dialog box setting.

5. The **Save As Default** button lets you Save any values you change to the drop down menus for the fields.
6. The **Restore Default**s button lets you return to the original values. Any values you provided through **Save As Default** remain on the drop down menus for the fields for surface and normal deviations
7. Click **Apply** to apply the current values to the active modeler window, and **Close** or **Cancel** to close the dialog without changing settings.

Related Topics

Rendering Objects as Wireframes or Solids

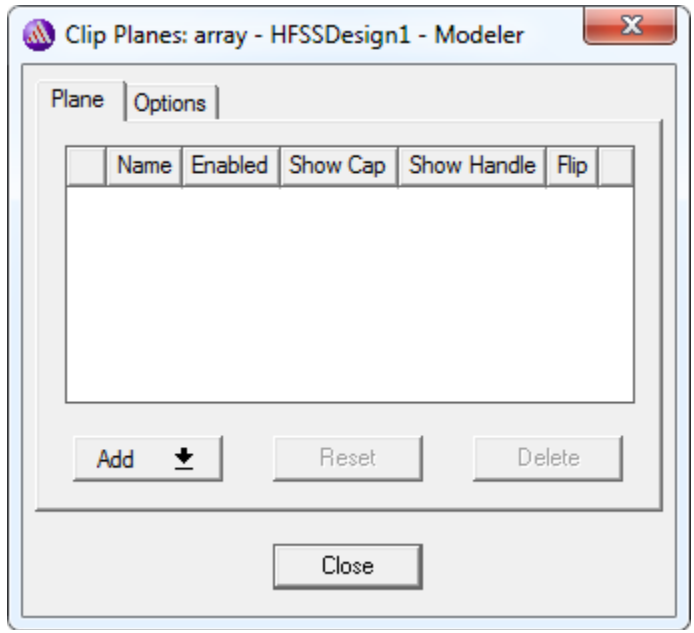
Using Clip Planes

The **View>Clipping>Clip Plane** command lets you define a clip plane that you can use to interactively make any desired cut-away view of a model. If you use **Edit>Copy Image** or **Modeler>Export>***image format* with the clip plane active, the image shows the clipped plane. When parts of the model are hidden by a clip plane, model selection works as though only the visible parts are present.

To add a clip plane:

1. Click **View>Clipping>Clip Plane**...

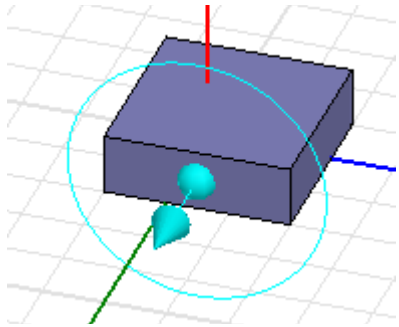
   This displays the **Clip Plane** dialog with the Plane tab selected.



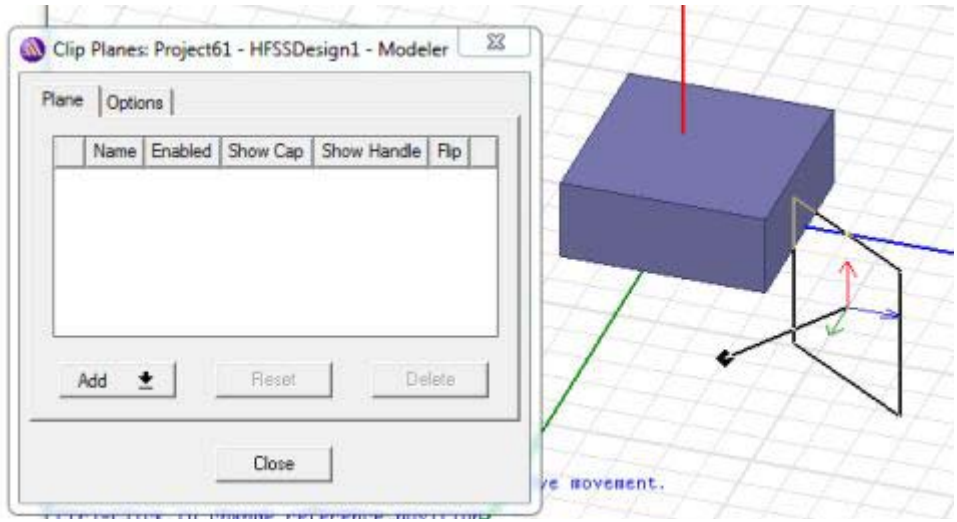
2. The **Add** button contains a drop down menu with choices for **Specify center, normal**, and **Use selection**. Of you want to use selection, you must first select a face or a cut plane.

- If you first select a face or cut plane, and then click **Add>Use Selection**, the clip plane is

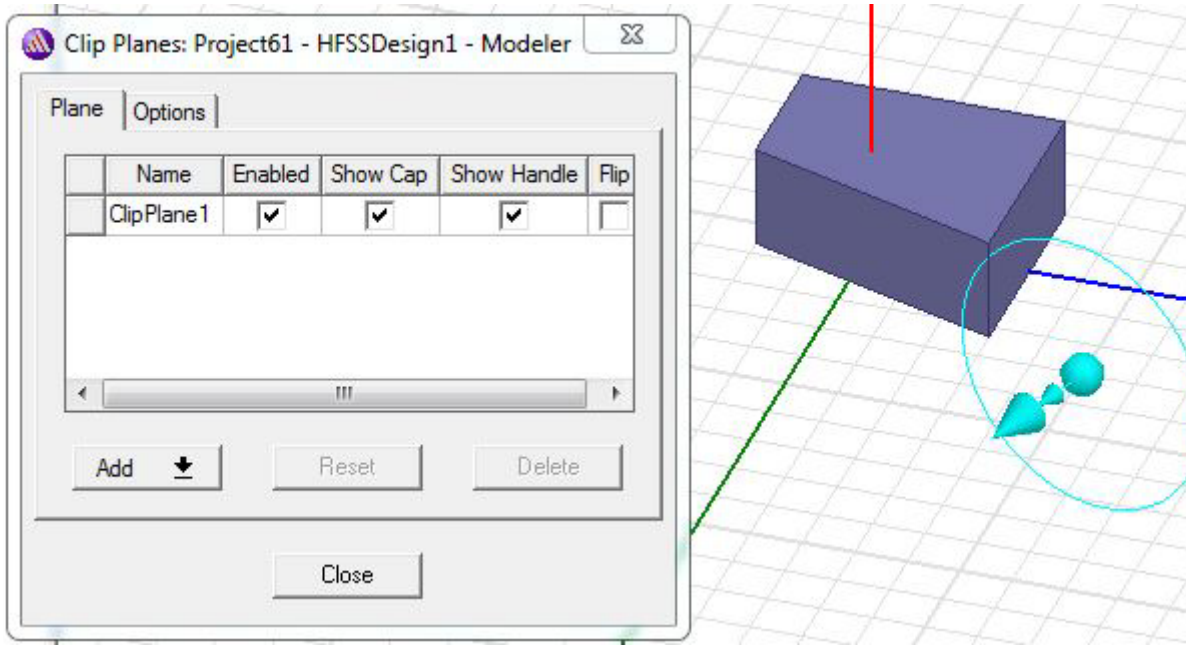
added on that face.



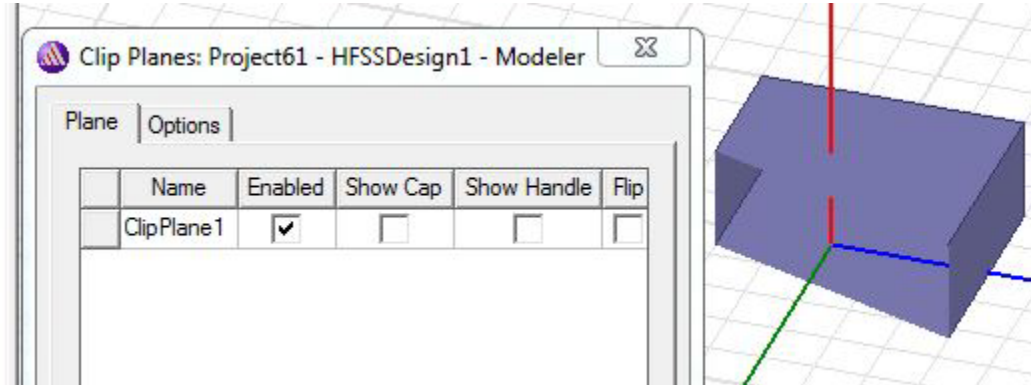
a. If you select **Add>Specify center, normal**, this launches a Measure dialog and enters a mode for you click to first define the start location, shown as a triad. When you move the cursor, a rectangle represents the clip plane, and a vector the current direction.



b. Click again to set the reference position.

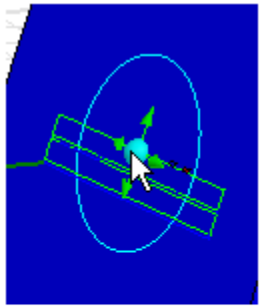


After the second click, the clip plane is active. The handle is visible as a circle with a sphere at the center, and an arrow pointing the normal for the plane. The **Clip Planes** dialog shows the clip plane name, that it is enabled, shows the cap (which is the plane surface), and shows the handle. The Flip selection lets you reverse the direction of the clip plane. If you uncheck Show cap and/or show handle, they disappear from the display.

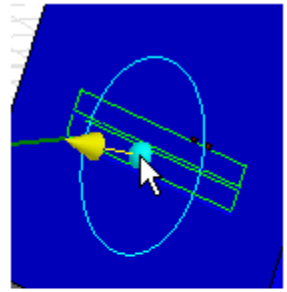


3. With Show Handle enabled, you can use the handle to manipulate the location and orientation of the clip plane. The handle changes appearance and function relative to the position of the

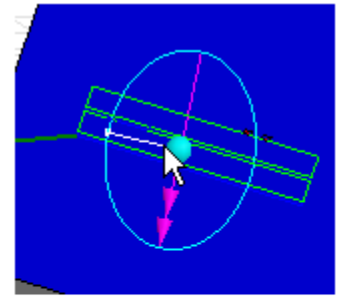
cursor. Dragging the cursor makes use of the current function.



Move Handle independently of clip plane. This lets you move the handle away from the model.



Move Handle while dragging clip plane



Rotate clip plane by dragging the cursor. The white line is parallel to the clip plane rotation. The double arrow line is the axis.

Move the cursor around the center of the handle to change the axis.

4. The **Options** tab for the **Clip Planes** dialog contains four options.
   - Force opaque for the unclipped portion.
   - Disable clip plane when drawing a new clip plane.
   - Plane handle color

     The button shows the current color. Click the button to display a color selection dialog. Select a default or custom color and click OK.
   - Plane handle radius.

     This slider lets you resize the radius of the handle to the most convenient size. The radius resizes dynamically. When you close and reopen the modeler window, the last selected size persists.

   You can save your choices as new defaults.

   Related Topics

   Copy Image

   Exporting Graphics Files

Modifying the View Orientation

To change the orientation of the view (the viewing direction) in the view window:

1. Click **View**>**Modify Attributes**>**Orientation**.

   A dialog box with orientation settings appears. It includes a table of names of six basic orientations (Top, Bottom, Right, Left, Front, Back), any additional orientations that you have added, and a section for adding new orientations. The table includes columns that show the input angles, and the equivalent vector components.
2. Apply a default orientation to the view or create and apply a new orientation.
3. Click **Apply to View** for the selected view to appear in the view window.

   You can use the **Reset View Orientation** button to restore the view to the original angle.
4. Click **Make Default** if you want the selected viewing direction to be the initial viewing direction when a **3D Modeler** window is opened, either in the current project or future projects.
5. Click **Close** to dismiss the dialog box.

   The orientation you set will be saved with the design. New orientations assigned to other designs after this point will not affect this orientation.

Related Topics

Applying a Default View Orientation

Applying a New Orientation

Removing an Orientation

Applying a Default View Orientation

To apply a default viewing direction to the active view window:

1. Click **View**>**Modify Attributes**>**Orientation**.

   A dialog box with orientation settings appears.
2. Click one of the orientation names listed in the viewing directions list.
3. Click **Apply**.

   The viewing direction will be applied to the active view window.

   You can use the **Reset View Orientation** button to restore the view to the original angle.
4. Click **Close**.

Related Topics

Applying a New Orientation

Removing an Orientation

Applying a New View Orientation

To create and apply a new viewing direction to the active view window:

1. Click **View**>**Modify Attributes**>**Orientation**.

   A dialog box with orientation settings appears.

2. To create a viewing direction that is based on a default viewing direction, click the existing orientation name in the viewing directions list.

   To create a viewing direction based on the current view in the **3D Modeler** window, click **Get Current View Direction**.

   - To modify the selected orientation's vector components, select **Input vector components** under **Add Orientation to List**, and then modify the values in the **Vx**, **Vy**, or **Vz** text boxes, and the Up vector boxes for Ux, Uy, and Uz.
   - To modify the selected orientation's input angles, select **Input angles** under **Add Orientation to List**, and then modify the values in the psi, phi and theta text boxes.

3. Type a name for the new orientation in the **Name** text box.
4. Click **Add/Edit**.

   The new orientation is added to the list of viewing directions.

5. Click **Make Default** if you want the new viewing direction to be the initial viewing direction when a **3D Modeler** window is opened in the current project or future projects.
6. Click **Close**.

Related Topics

Applying a Default View Orientation

Removing an Orientation

Removing an Orientation

To remove a viewing direction from the list in the orientation settings dialog box:

1. Click **View**>**Modify Attributes** >**Orientation**.

   A dialog box with orientation settings appears.

2. Click the viewing direction you want to delete from the list of names.
3. Click **Remove**.

   The viewing direction is removed from the list.

This operation cannot be undone.

Related Topics

Applying a Default View Orientation

Applying a New Orientation

Modifying the Lighting

You have the option to emit the following types of light on a design:

- *Ambient* lighting surrounds the model evenly with light. All objects are lit evenly in every direction by a color of light that you specify.
- *Distant* lighting directs a ray of light at the model in a direction you specify. By default, two distant light vectors are in effect for every new view window.

To modify the lighting:

1. Click **View**>**Modify Attributes>Lighting**.

   The **Lighting Properties** dialog box appears.
2. Select **Do Not Use Lighting** to turn off ambient and distant lighting.

   Clear this option to activate ambient and distant lighting.
3. To surround the model with light, click the **Ambient Light Properties** color button, and then select a color for the surrounding light from the **Color** palette.
4. To modify the distant light on a model, do one of the following:
   a. Add a new distant light by clicking **Add**.
   b. Copy an existing distant light that you intend to modify by first selecting it in the **Distant Light Vectors** table, and then clicking **Clone**.
   c. Select a default distant light to modify by selecting it in the **Distant Light Vectors** table.
5. For the selected distant light vector, specify the vector direction:
   a. To modify the direction by specifying Cartesian coordinates, do one of the following:
      - Enter the new Cartesian coordinates in the **X**, **Y**, and **Z** boxes.
      - Use the **Vx**, **Vy**, and **Vz** sliders to specify the Cartesian coordinates dynamically.
   b. To modify the direction by specifying the spherical coordinates, do one of the following:
      - Enter the new spherical coordinates in the ϕ and θ boxes.
      - Use the ϕ and θ sliders to specify the spherical coordinates dynamically.
6. To revert to the default ambient and distant light settings, click **Reset**.
7. Click **Save As Default** if you want the new lighting settings to be the defaults for all **3D Modeler** windows, either in the current project or future projects.
8. Click **OK** to dismiss the dialog box.

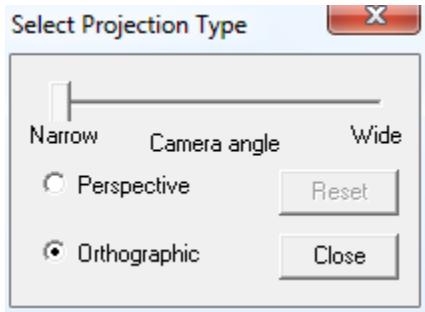
The lighting settings will be saved with the design. New lighting applied to other designs after this point, including new default settings, will not affect these lighting settings.

Setting the Projection View

To modify the projection of model objects (the camera angle) in the view window:

1. Click **View**>**Modify Attributes**>**Projection**.

   The **Select Projection Type** window appears:



2. Select **Perspective** to enable the slider to change the angle of the view.
   - Move the slider to the right to increase the proximity, or widen, the view. Move the slider to the left to decrease the proximity, or flatten, the view.

   Objects that are closer appear larger relative than objects that are farther away.
3. Select **Orthographic** to view the model without distortion.

   The slider is disabled because a distortion scale is no longer applicable.
4. Click **Reset** to return the model to its original view.
5. Click **Close** to accept the projection setting and dismiss the window.

   The **Select Projection Type** window closes. The last view you specified in the projection window remains visible in the view window.

The projection view you set will be saved with the design. New projection views assigned to other designs after this point will not affect this projection setting.

Setting the Background Color

To set the color of the background in the view window:

1. Click **View**>**Modify Attributes**>**Background color**.

   The **Select Background Color** window appears.
2. To assign a solid background color, do the following:
   a. Select **Plain Background**.
   b. Modify the background color in one of the following ways:
      - Click the **Background Color** button and then select a color from the **Color** palette.
      - Use the RGB sliders under **Change View Color Dynamically** to specify the color's red, green, and blue values.
3. To assign a background color that gradually changes from one color to another, do the following:
   a. Select **Gradient Background**.
   b. Specify the background color at the top and bottom of the view window in one of the following ways:
      - Under **Select Background Type**, click the **Top Color** button and select a color from the **Color** palette. Then click the **Bottom Color** button and select a color from the **Color** palette.
      - Under **Change View Color Dynamically**, click **Top Color** or **Bottom Color** and use the RGB sliders to specify the color's red, green, and blue values.
4. Click **Reset** to revert to the default background colors.
5. Click **Save As Default** if you want the new background color to be the background color for all **3D Modeler** windows in either the current project or future project.
6. Click **OK**.

The background color you set will be saved with the design. New background color settings assigned to other designs after this point, including new default settings, will not affect this design.

Modifying the Coordinate System Axes View

The coordinate system axes displays the x, y, z orientation from the origin point for the current working coordinate system.

What do you want to do?

- Show or hide the coordinate system axes.
- Show the coordinate system axes for selected objects.
- Enlarge or shrink the size of the coordinate system axes.
- Show or hide the triad axes.

Showing or Hiding the Axes

1. Click **View**>**Coordinate System**, then click one of the following:
   - **Hide** to hide the x-, y-, and z-axes in the active view window.
   - **Show** to display the x-, y-, and z-axes in the active view window.

Show the Axes for Selected Objects

1. Click **Tools**>**Options**>**Modeler Options**.

   This displays the Modeler Options dialog.
2. Select the **Display** tab.
3. Select **Show orientation of selected objects**.
4. Click **OK** to close the dialog.

Enlarging or Shrinking the Axes

1. Click **View**>**Coordinate System**, then click one of the following:
   - **Large** to display the x-, y-, and z-axes as extending to the edges of the active view window.
   - **Small** to display the x-, y-, and z-axes in a smaller size in relative to the edges of the active view window.

Showing or Hiding the Triad Axes

The triad is a secondary depiction of the coordinate system that appears at the lower right of the Modeler window. It shows the orientation of the currently selected working coordinate system. It can be shown or hidden separately from the selected coordinate system.

To show the triad:

1. Click **View**>**Coordinate System**>**Triad**, then click one of the following:
   - **Hide** to hide the triad x-, y-, and z-axes at the lower right of the active view window.
   - **Show** to display the triad x-, y-, and z-axes in the lower right active view window.
   - **Auto** to generally hide the triad axes.

Choosing Grid Settings

The grid displayed in the **3D Modeler** window is a drawing aid that helps to visualize the location of objects.

For Cartesian grids, the location of points on the grid are defined by intersections of planes that are perpendicular to and along the x-, y-, and z-axes. The division (the distance between neighboring parallel planes perpendicular to the same axis) can be set.

For polar grids, the location of points on the grid are defined by intersections of planes that are perpendicular to the local radius and angle coordinates. The division (the distance between neighboring parallel planes perpendicular to the same radius and angle) can be set.

Grid spacing is set according to the current project's drawing units.You can control the following aspects of the grid:

- Type (rectangular or circular)
- Style (dots or lines)
- Density
- Spacing
- Visibility
- Snap settings
- Grid plane

Setting the Grid Type

1. Click **View**>**Grid Settings**.

   The **Grid Settings** window appears.

2. Select a grid type for the active view window: **Cartesian** for a rectangular grid or **Polar** for a circular grid.

   The grid in the active view window is centered at the origin of the working coordinate system.

For Cartesian grids, you will define a coordinate by specifying its distance from the origin along each axis in the **X**, **Y**, and **Z** text boxes or its relative distance from the previously selected point in the **dX**, **dY**, and **dZ** text boxes.

For polar grids, you will define a coordinate by specifying its radius from the origin in the **R** text box and its angle from the x-axis in the **Theta** text box or its relative distance from the previously selected point in the **dR** and **dTheta** text boxes.

Related Topics

Choosing Grid Settings

Setting the Grid Style

1. Click **View**>**Grid Settings**.

   The **Grid Settings** window appears.

2. Select one of the following grid styles for the active view window:

   Dot Displays each grid point as a dot.

   Line Displays lines between grid points.

Related Topics

Choosing Grid Settings

Setting the Grid Density and Spacing

1. Click **View**>**Grid Settings**.

   The **Grid Settings** window appears.

2. If you want to change the density of the grid in the active view window as you zoom in or out on objects, do the following:

   a. Select **Auto adjust density to**.

   b. Specify a distance between grid points by typing a value in the **pixels** box.

      The default is set to 30 pixels, which is generally the best setting for displaying objects.

3. If you do not want the grid density to change when you zoom in or out, but instead want to specify a constant grid spacing, do the following:

   a. Clear the **Auto adjust density to** option.

   b. Specify the grid's spacing in the active design's units.

      If you selected a Cartesian grid type, type the values of **dX**, **dY**, and **dZ**. These values represent the difference between one grid point and the next in the x, y, and z directions, respectively.

      If you selected a polar grid type, type the values for **dR** and **dTheta**. **dR** represents the difference between each radius. **dTheta** is the difference between angles.

   The distance between grid points will increase and decrease proportionately as you zoom in and out in the active view window.

4. Click **OK**.

Related Topics

Choosing Grid Settings

Setting the Grid's Visibility

- To hide the grid, click the Grid toolbar icon:  . Click it again to show the grid.

Alternatively:

1. Click **View**>**Grid Settings**.

   The **Grid Settings** window appears.

2. Select **Grid Visible** to make the grid visible in the active **3D Modeler** window.

   Clear the selection to make the grid invisible.

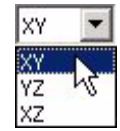
Related Topics

Choosing Grid Settings
Setting the Grid Plane

Setting the Grid Plane

To specify the plane on which you want to display the grid in the active view window, do one of the following:

- Click **Modeler>Grid Plane**, and then select a grid plane: **XY**, **YZ**, or **XZ**.
- Click a grid plane on the pull-down list on the 3D Modeler Draw toolbar:



Related Topics

Choosing Grid Settings

13

Defining Mesh Operations

In HFSS, mesh operations are optional mesh refinement settings that provide HFSS with mesh construction guidance. This technique of guiding HFSS's mesh construction is referred to as "seeding" the mesh. Seeding is performed using the **Mesh Operations** commands on the HFSS menu. Each mesh operation you define appears in the Project tree.

You can instruct HFSS to reduce the maximum size of tetrahedral elements on a surface or within a volume until they are below a certain value (length-based mesh refinement) or you can instruct HFSS to refine the surface triangle length of all tetrahedral elements on a surface or volume to within a specified value (skin depth-based mesh refinement). These types of mesh operations are performed on the *current mesh*, that is, the most recently generated mesh.

In a few circumstances, you may also want to create a mesh operation that modifies HFSS's surface approximation settings for one or more faces. Surface approximation settings are only applied to the *initial mesh*, that is, the mesh that is generated the first time a design variation is solved.

You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

You can also choose to override automatic choice of which mesher HFSS uses, by using the HFSS>Mesh Operations>Initial Mesh Settings command.

See the technical notes for more details about HFSS's application of mesh operations.

What do you want to do?

- Length Based Mesh Refinement

- Perform length-based mesh refinement on object faces.
- Perform length-based mesh refinement inside objects.
- Perform skin depth-based mesh refinement on object faces.
- Modify surface approximation settings for one or more faces.
- Specify automatic or specified model resolution for a selection.
- Specify the initial mesh settings

Related Topics

Plotting the Mesh

Specifying a Source for the Initial Mesh.

Technical Notes: The Mesh Generation Process

Technical Notes: Seeding the Mesh

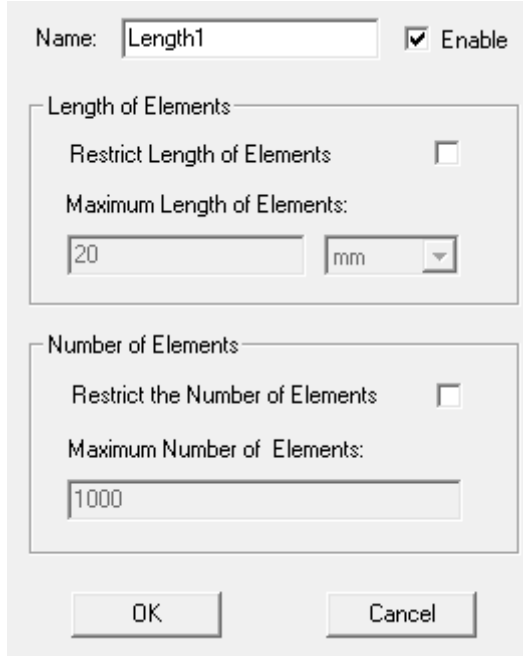
Technical Notes: Guidelines for Seeding the Mesh

Technical Notes: Surface Approximation Settings

Getting Started Guides: A Spiral Inductor

Length Based Mesh Refinement

The length based mesh refinement settings help control the length of the tetrahedral elements and the growth of the whole mesh. If you are not sure how big a refinement is needed, by way of mesh refinement you can set a ceiling on the mesh growth with the option **Restrict the Number of Elements**.



When you set a mesh operation on a selection of objects (more than one) and set a maximum length, it applies to the whole mesh. Even if you selected just one object, when you refine its boundary, elements are added both inside and outside of that object.

When there are no more elements to refine or when no more elements can be added, the refinement process stops because it has either met the growth limit or the element length target. However, there are more quality improvement swaps, smoothing, matching boundary issues performed at the finishing stage that might slightly alter the mesh size.

If you restrict the number of elements to say 10000, it will add 10000 elements. If you have multiple mesh operations each for 10000 elements, each mesh operation will add 10000 elements (assuming there was no element length targets).

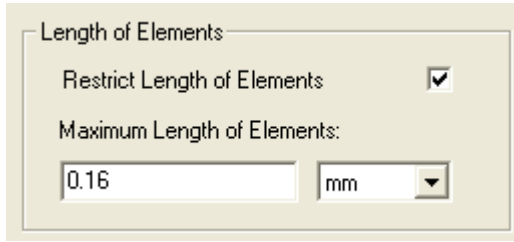
If you set a restriction of say 2000 elements on an object which creates 1100 elements after lambda refinement without any mesh operation, the starting mesh will contain more than 3000 elements.

Assigning Length-Based Mesh Refinement on Object Faces

1. Select the faces you want HFSS to refine.

   Alternatively, select an object if you want HFSS to refine every face on the object.
2. Click **HFSS>Mesh Operations>Assign>On Selection>Length-Based**.

   The **Element Length-Based Refinement** dialog box appears.
3. Type a name for the mesh operation in the **Name** text box or accept the default name.
4. To restrict the length of tetrahedra edges touching the faces:



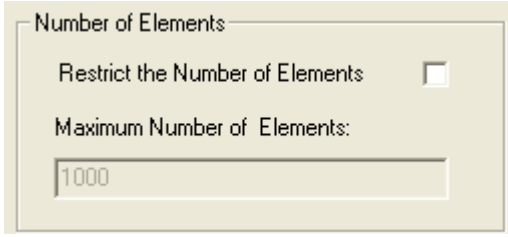
   a. Select **Restrict Length of Elements**.

   b. Type the maximum length of the tetrahedral edges touching the faces in the **Maximum Length of Elements** text box.

      HFSS will refine the element edges touching the selected faces until their lengths are equal to or less than this value.

      The default value is set to 20% of the maximum edge lengths of the bounding boxes of each selected face.

      For first order basis functions, a maximum length of $\frac{\sqrt{2}}{10}\lambda$ is recommended for radiation boundary surfaces.
5. To restrict the number of elements added during refinement of the faces:



   a. Select **Restrict the Number of Elements**.

   b. Enter the **Maximum Number of Elements** to be added.

   c. Click **OK**.

When the mesh is generated, the refinement criterion you specified is used. If the maximum number of elements is reached, some elements may exceed the requested maximum element length.

You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

Related Topics

Plotting the Mesh

Technical Notes: Length-Based Mesh Refinement

Technical Notes: Seeding the Mesh

Technical Notes: Guidelines for Seeding the Mesh

Assigning Length-Based Mesh Refinement Inside Objects

Applying Mesh Operations without Solving

Technical Notes: The Mesh Generation Process

Assigning Length-Based Mesh Refinement Inside Objects

To instruct HFSS to refine every face of an object and its interior:

1. Select the object you want HFSS to refine.
2. Click **HFSS>Mesh Operations>Assign>Inside Selection>Length-Based**.

   The **Element Length-Based Refinement** dialog box appears.
3. Type a name for the mesh operation in the **Name** text box or accept the default name.
4. To restrict the length of the tetrahedral element edges inside the object:
   a. Select **Restrict Length of Elements**.
   b. Type the maximum length of the edges inside the object in the **Maximum Length of Elements** text box.

      The default value is set to 20% of the maximum edge lengths of the bounding boxes of each selected object's faces.

      HFSS will refine the element edges inside the object until they are equal to or less than this value.
5. To restrict the number of elements added during the refinement inside the object:
   a. Select **Restrict the Number of Elements**.
   b. Enter the **Maximum Number of Elements** to be added.
   c. Click **OK**.

When the mesh is generated, the refinement criterion you specified will be used. If the maximum number of elements are reached, it may result in some elements exceeding the requested maximum element length.

You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

Related Topics

Plotting the Mesh

Technical Notes: Length-Based Mesh Refinement

Technical Notes: Seeding the Mesh

Technical Notes: Guidelines for Seeding the Mesh

Assigning Length-Based Mesh Refinement on Object Faces

Applying Mesh Operations without Solving

Technical Notes: The Mesh Generation Process

Assigning Skin Depth-Based Mesh Refinement on Object Faces

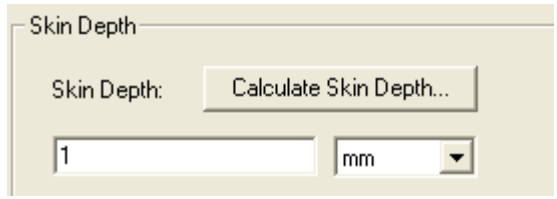
1. Select the faces you want to be refined.

Note It is possible to select a body and convert it to selecting all faces of the body. The user can use this method to select all faces and toggle a few faces out of selection. Selecting the whole body might select very large regions for refinement and increase the element count a lot.

2. Click **HFSS>Mesh Operations>Assign>On Selection>Skin-Depth-Based**.

   The **Skin Depth-Based Refinement** dialog box appears.

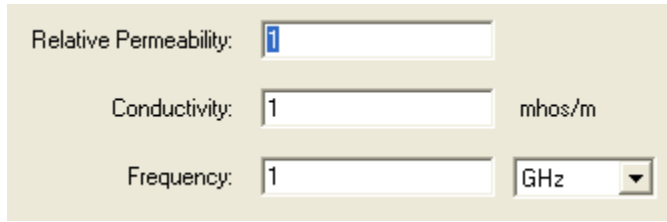
3. Type a name for the mesh operation in the **Name** text box or accept the default name.
4. Type the skin depth within which to refine the mesh in the **Skin Depth** text box.



   Alternatively, calculate the skin depth based on the object's material permeability and conductivity and the frequency at which the mesh will be refined:

   a. Click **Calculate Skin Depth**.

      The **Calculate Skin Depth** dialog box appears.



   b. Enter the material's **Relative Permeability** and **Conductivity**.

c. Specify the **Frequency** at which to refine the mesh.

d. Click **OK**.

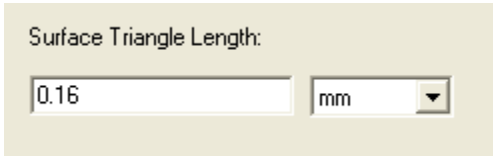
HFSS calculates the skin depth and enters its value in the **Skin Depth** text box.

5. In the **Number of Layers of Elements** text box, type the number of layers to add perpendicular to the object's surface.

   HFSS will add an equivalent number of mesh points to each layer. For example, if HFSS added 10 points to satisfy the **Surface Triangle Length**, it will add 10 points to each layer.

6. Type the maximum edge length of the surface mesh in the **Surface Triangle Length** text box.

   The default value is set to 20% of the maximum edge lengths of the bounding boxes of each selected face.



HFSS will refine the surface triangle mesh (the faces of the tetrahedra touching the surface) until their edge lengths are less than or equal to the specified value.

7. To restrict the number of elements added during refinement on the faces:

   a. Select **Restrict the Number of Surface Elements**.

   b. Enter the **Maximum Number of Surface Elements** to be added.

   c. Click **OK**.

When the mesh is generated, the refinement criteria you specified will be used. This operation will be approximately the same as having slabs of tetrahedra, but it is not guaranteed to prevent tetrahedra from crossing slab interfaces. Caution should be used with this mesh operation, as very thin layers may cause a reduction in mesh quality or unnecessarily cause the generation of a very large mesh. Further regions refined under this operation and its close neighbors do not participate in solution adaptive refinement. This is another reason to use this seeding operation with caution.

You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

Related Topics

Plotting the Mesh

Technical Notes: Skin Depth-Based Mesh Refinement

Technical Notes: Seeding the Mesh

Technical Notes: Guidelines for Seeding the Mesh

Applying Mesh Operations without Solving

Technical Notes: The Mesh Generation Process

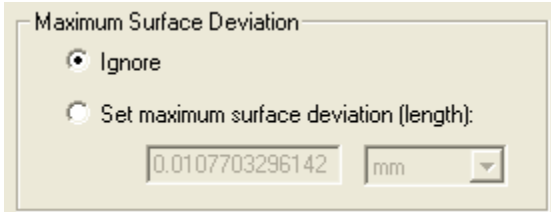
Modifying Surface Approximation Settings

HFSS applies surface approximation settings for selected objects when it generates the initial mesh. If you modify HFSS's default settings after the initial mesh has been generated, they will not affect the mesh for that design variation. Surface approximation makes sense for curved surfaces, for which the mesh will not exactly reproduce the surface shape. It also can be used to restrict the aspect ratio of triangles on planar surfaces.

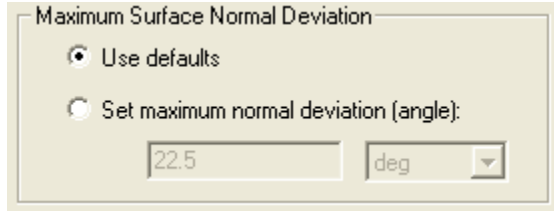
1. Select the faces for which you want to modify the surface approximation settings.

   Alternatively, select an object if you want to modify the surface approximation settings of every face on the object.
2. Click **HFSS>Mesh Operations>Assign>Surface Approximation**.

   The **Surface Approximation** dialog box appears with the **Surface Approximation** tab select.
3. Type a name for the group of settings in the **Name** text box or accept the default name.
4. Under **Maximum Surface Deviation**, do one of the following:
   - Select **Ignore** if you do not want to use surface deviation settings for the selected faces.
   - Select **Set maximum surface deviation (length)**, and then type the distance between the true surfaces of the selected faces and the meshed faces in the text box.

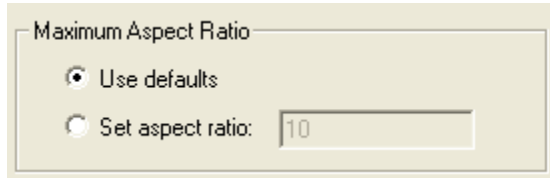


5. Under **Maximum Surface Normal Deviation**, do one of the following:
   - Select **Use defaults** if you want to use HFSS's default normal deviation setting for the selected faces, which is 22.5 degrees.
   - Select **Set maximum normal deviation (angle)**, and then type the angular distance between the normal of the true surface and the corresponding mesh surface in the text box.



6. Under **Maximum Aspect Ratio**, do one of the following:
   - Select **Use defaults** if you want to use HFSS's default aspect ratio settings for the selected faces, which are 10 for curved surfaces and 200 for planar surfaces. A ratio of 200 corresponds roughly to a triangle with an interior angle of one half degree.

- Select **Set aspect ratio**, and then type a value in the text box. This value determines the shape of the triangles. The higher the value, the thinner the triangles. Values close to 1 will result in equilateral triangles.



7. Select the **Surface Representation Priority for Tau Mesh**:

   In most cases, meshing is done by Tau Mesh. (See Initial Mesh Settings.) You can set the surface representation as normal or high.

   - Normal - use for normal situations for the Tau Mesh, allowing loose tolerances.
   - High - Use only on critical surfaces. (For example, a very small port in a large model.) The choice of "High" can help to mesh some difficult or small surfaces, but it can also be very expensive and can cause the mesher to fail.

8. Click **OK**.

   The settings will be applied to the initial mesh generated on the selected surface. The group of settings is listed in the project tree under **Mesh Operations**.

You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

Related Topics

Plotting the Mesh

Technical Notes: Surface Approximation Settings

Technical Notes: Guidelines for Modifying Surface Approximation Settings

Technical Notes: The Mesh Generation Process

Specifying the Model Resolution

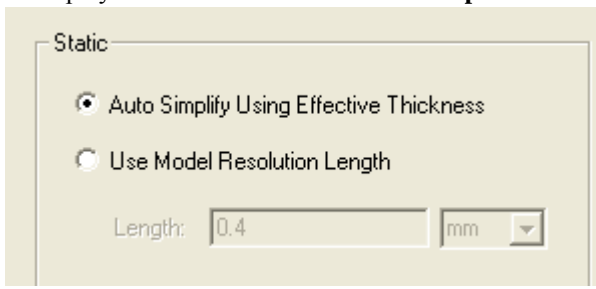
The model resolution parameter is used by the meshmaker to distinguish large features from small features in the model. This setting controls how large a feature must be to be resolved by the meshmaker. For example, if you set the model resolution length to 20 mm, any model features smaller than 20 mm are not represented in the mesh. Neither the model nor the model files are changed. The resolution only controls how the mesh for the model is represented.

While removing small details, if the meshmaker finds that the representation of the model is not accurate enough, it returns an error condition. The meshmaker then starts with the most accurate representation and prunes away the details smaller than the model resolution length. It returns an error if the specified model resolution length forces the final representation to deviate too greatly from the model.You can set Model Resolution on one or more objects to remove unnecessary details from the mesh representation. This can be used to reduce the mesh complexity of the selected objects.

A cautionary note is needed concerning the use of model resolution. It can sometimes make model faces fail to be represented in the mesh, which can effectively remove some of your boundary conditions. This can only happen if faces are closer together than the model resolution distance you specify.

1. Select the object or objects on which to specify a Model Resolution operation.
2. Click on **HFSS>Mesh Operations>Assign>Model Resolution**.

   This displays the **Model Resolution Mesh Operation** dialog.



   Alternatively, you can display the same dialog if you:

   a. Right-click on either **Mesh Operations** in the Project Tree, or right-click in the **3D Modeler** window to display the respective shortcut menu.
   b. Click on **Assign>Model Resolution** in the **Project Tree** menu or click on **Assign Mesh Operation>Model Resolution** on the shortcut menu.

   The **Model Resolution Mesh Operation** dialog contains text fields for the mesh operation Name and radio buttons with choices for the following

   - **Auto Simplify Using Effective Thickness**

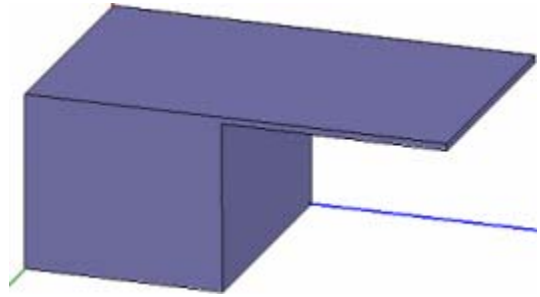
     The mesher calculates the resolution length based on each object's effective thickness. One mesh operation can be assigned to many objects, and each will be simplified based on its own dimensions. Use the **Auto Simplify** selection:

- To remove many details while retaining an object's overall shape and size.
- For objects of generally uniform thickness.
- To assign one mesh operation to many objects.

- **Use Model Resolution length**

  This enables fields for you to specify the resolution value and units. Use this selection for:

  - Tighter control of mesh accuracy.
  - Objects of non-uniform thickness. For example, the thin section of the object shown below might be lost with **Auto Simplify**.



3. After defining the operation, click **OK**.

   This adds the named Model Resolution operation under the **Mesh Operations** icon in the Project Tree.

Note Setting Model Resolution will invalidate any existing solutions.

When two objects in contact have different model resolution lengths, the smaller length will apply for the common regions.

Related Topics

Plotting the Mesh

Setting the Healing Options

Technical Notes: Model Resolution

Reverting to the Initial Mesh

The initial mesh is the mesh that is generated the first time a design variation is solved. It includes surface approximation settings, but does not include lambda refinement or defined mesh operations.

If you have modified the design setup, and do not want to use the existing current mesh, revert to the initial mesh prior to solving.

- On the **HFSS>Analysis Setup**>**Revert to Initial Mesh**.

Reverting to the initial mesh is useful when you want to evaluate how a different solution frequency affects the mesh generated during an adaptive analysis. You lose all solution data for a solve setup and all of its sweeps when you revert to the initial mesh for that setup. You can do this for all solve setups at once by selecting the command through Analysis in the menu system or project tree, or for a specific solve setup via its right mouse click menu in the project tree.

Related Topics

Plotting the Mesh

Technical Notes: The Mesh Generation Process

Applying Mesh Operations without Solving

If you want to refine the mesh on a face or volume, but do not want to generate a solution, do the following after defining mesh operations:

- Click **HFSS**>**Analysis Setup**>**Apply Mesh Operations**, or right click on the **Analysis** or **Setup** icon in the **Project** window to display the shortcut menu and click **Apply Mesh Operations**.

The same solve machine rules that apply to solving any other setup also apply here. The mesh operation will be sent to the default solve machine, or the HFSS **Server Setup** dialog may appear to allow you to interactively specify a solve machine if "Prompt for analysis machine when launching analysis" is selected under **Tools>Options>General Options>Analysis Options** tab.

- If a current mesh has been generated, HFSS will refine it using the defined mesh operations.
- If a current mesh has not been generated, HFSS will apply the mesh operations to the initial mesh.
- If an initial mesh has not been generated, HFSS will generate it and apply the mesh operations to the initial mesh.
- If the defined mesh operations have been applied to the selected face or object, the current mesh will not be altered.

Hint Define a new mesh operation rather than modify an existing mesh operation. HFSS will not re-apply a modified mesh operation.

Applying mesh operations without solving enables you to experiment with mesh refinement in specific problem regions without losing design solutions. You cannot undo the applied mesh operations, but you can discard them by closing the project without saving them, or by reverting to the initial mesh.

Related Topics

Technical Notes: The Mesh Generation Process

Plotting the Mesh

HPC and Analysis Options

Remote Analysis

Specifying Initial Mesh Settings

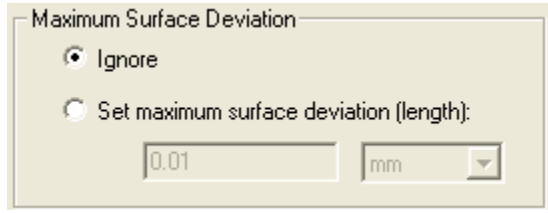
You can specify the initial mesh settings, including the surface approximation and the meshing approach. Initial Mesh Settings apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

For most designs, you can let HFSS automatically choose which of two meshing approaches to take. HFSS predicts which one gives the best results, balancing mesh reliability, speed, quality, size and design characteristics. In most cases, HFSS uses TAU mesh, rather than the classic mesh. In general, it looks for specific features (for example, stacks of large planar parallel facets with small gaps) and situations where the initial Tau mesh is 4 times larger than the Classic. In a few cases, you may decide to override the automatic choice and designate the mesher to use. To do so:

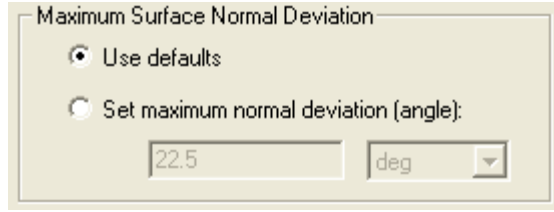
1. Select **HFSS>Mesh Operations>Initial Mesh Settings...** or in the Project tree, right-click on Mesh Operations, and select **Initial Mesh Settings** from the shortcut menu.

   The **Initial Mesh Settings** dialog appears with the **Surface Approximation** tab selected.

2. Under **Maximum Surface Deviation**, do one of the following:
   - Select **Ignore** if you do not want to use surface deviation settings for the selected faces.
   - Select **Set maximum surface deviation (length)**, and then type the distance between the true surfaces of the selected faces and the meshed faces in the text box.

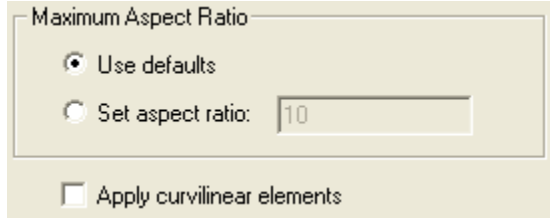


3. Under **Maximum Surface Normal Deviation**, do one of the following:
   - Select **Use defaults** if you want to use HFSS's default normal deviation setting for the selected faces, which is 22.5 degrees.
   - Select **Set maximum normal deviation (angle)**, and then type the angular distance between the normal of the true surface and the corresponding mesh surface in the text box.



4. Under **Maximum Aspect Ratio**, do one of the following:
   - Select **Use defaults** if you want to use HFSS's default aspect ratio settings for the selected faces, which are 10 for curved surfaces and 200 for planar surfaces.

- Select **Set aspect ratio**, and then type a value in the text box. This value determines the shape of the triangles. The higher the value, the thinner the triangles. Values close to 1 will result in well-formed, wide triangles.



5. You can also uncheck or check **Apply curvilinear elements**. For models with curved surfaces, this increases accuracy, though it costs more memory. In some situations, you may choose to uncheck the setting. See Rectilinear Elements and Curvilinear Elements and Modifying SAR Settings.
6. To make your choice the default, use the **Save as default** checkbox.
7. Select to the **Meshing Method** tab.

   It contains radio buttons for:

   - Auto (the default)-- HFSS automatically selects the mesher. In most cases, this will be TAU mesh.
   - TAU Mesh--this includes Surface representation choices for Strict or Tolerant. Strict performs stitching, resolves surfaces and contacts more accurately, but takes more time and may cause issues for dirty models. Tolerant uses a looser tolerance for surface representation, which may be better for dirty or very complex geometry.
   - Classic Mesh--this is based on the HFSS 11 mesher.
8. To make your choice the default, use the **Save as default** checkbox.
9. Click **OK** to apply your choices.

   The settings will be applied to the initial mesh generated.

Related Topics

Defining Mesh Operations

Deleting All Previously Assigned Mesh Refinements

To delete previously-assigned mesh refinement data:

- Click **HFSS>Mesh Operations>Delete All**.

Note This will not alter any existing meshes that have already made use of one or more mesh refinement operations. However, deleted mesh operations will not be used again after reverting to the initial mesh.

Viewing List of Mesh Operations

To list all mesh operations for the project, do the following from the project tree:

1. Right-click **Mesh Operations**.
   A shortcut menu appears.
2. Select **List** from the shortcut menu.
   The **Design List** dialog box appears, with the **Mesh Operations** tab displayed.
3. Click **Done** to close this dialog box.

Reassigning a Mesh Operation

To reassign a previously-assigned mesh setup, do the following in the project tree:

1. Select the object or objects to which you want to reassign the mesh operation.
2. Under **Mesh Operations**, right-click the mesh refinement you had previously defined. A shortcut menu appears.
3. Select **Reassign** from the shortcut menu.

   The mesh operation is reassigned to the currently selected objects.

Note This pertains to mesh setup operations only. However, if you want to see information about a mesh that has been generated, see Viewing Mesh Statistics.

To view the objects to which a mesh operation is assigned:

1. In the Project tree, select the mesh operation.
2. Right-click the mesh refinement you had previously completed. A shortcut menu appears.
3. Select **Show Assignment** from the shortcut menu.

   The objects to which the operation is assigned are selected.

Related Topics

Defining Mesh Operations

Viewing Mesh Properties

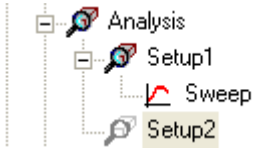
To view mesh properties, do the following in the project tree:

1. Under **Mesh Operations**, right-click the mesh refinement you had previously completed. A shortcut menu appears.
2. Select Properties from the shortcut menu.

14

Specifying Solution Settings

Specify how HFSS or HFSS-IE will compute a solution by adding a solution setup to the design. You can define more than one solution setup per design.



Each solution setup includes the following information:

- General data about the solution's generation.
- Adaptive mesh refinement parameters, if you want the mesh to be refined iteratively in areas of highest error.
- If you want to solve of a range of frequencies, you can add a frequency sweep.

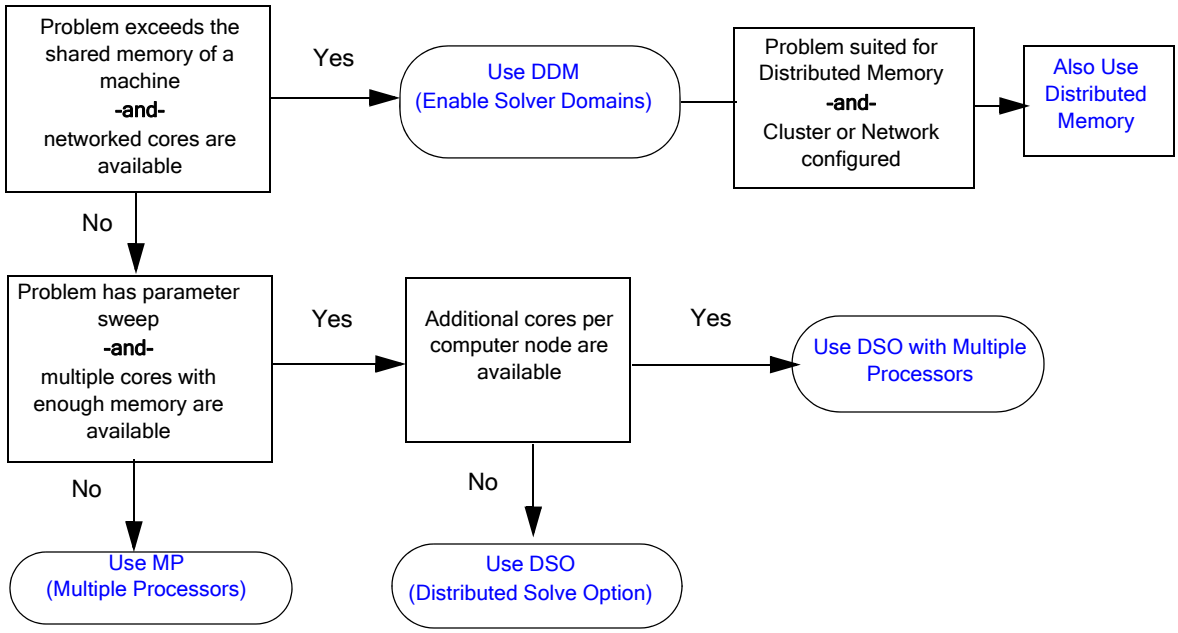
What do you want to do?

Add a Solution Setup to an HFSS Design

Add an Solution Setup to an HFSS-IE Design

Add a Solution Setup to an HFSS Transient Design

For large problems, depending the whether the solution involves a parameter sweep and your resources of shared machine memory, networked cores and cores per computing node, you can configure for different high performance computing solutions:



Additionally, a command line Large Scale DSO is used for 'large scale parallel' jobs, which either fail or scale poorly as Regular DSO jobs. A Large Scale DSO job does not support the output of full parametric results, but produces 'reduced' datasets corresponding to predefined Rectangular plots.

Related Topics

Defining Mesh Operations

Configure Distributed Memory Solutions for HFSS

Adding a Solution Setup to an HFSS Design

To add a new solution setup to a design:

1. Select a design in the project tree.
2. Click **HFSS**>**Analysis Setup**>**Add Solution Setup**  .
   - Alternatively, right click **Analysis** in the project tree, and then click **Add Solution Setup** on the shortcut menu.
   - If you have an existing setup, you can **Copy** and **Paste** it, and then edit parameters.
   - If you have already created a solution and you want to use an existing mesh, you can click **Add Dependent Solve Setup**.

   The **Solution Setup** dialog box appears. It is divided among the following tabs:

General	Includes general solution settings.
Options	Includes settings for lambda refinement, adaptive analysis and solution options, the Order of Basis setting, and whether to enable the use of solver domains, and whether to Save Fields and/or radiated fields only.
Advanced	Includes settings for mesh linking, absorbing boundaries on ports, and waveport adapt options.
Expression Cache	Includes a list of expressions (including post processing variables) that you can use for convergence for adaptive analysis.
Derivatives	If your design contains variables, they are listed here. HFSS can calculate derivatives for your variables.
Defaults	Enables you to save the current settings as the defaults for future solution setups or revert the current settings to HFSS's standard settings.

3. Click the **General** tab.
4. The **Enabled** checkbox on General tab permits to you to disable a setup so that it does not run when you select Analyze All.
5. For Driven solution types, do the following:
   a. Enter the **Solution Frequency** and select the frequency units from the pull down list.
   b. Optionally, select **Solve Ports Only**.

   For Eigenmode solution types, do the following:
   a. Enter the **Minimum Frequency** and the frequency units.
   b. Enter the **Number of Modes**. The number must be greater than 0 and less than 20.
6. If you are performing an adaptive analysis, enter **2** or more passes in the **Maximum Number of Passes** box, and then specify the remaining adaptive analysis parameters for HFSS.

   For Eigenmode solutions, if you are not performing an adaptive analysis, entering **0** will enable you to bypass the adaptive analysis process and just perform a frequency sweep. For driven problems HFSS always requires at least one adaptive pass. Entering **1** will also bypass adaptive anal-

ysis, generating a solution only at the solution frequency you specified.

7. The lower right corner also contains a button for HPC and Analysis options. Here you can select or create an analysis configuration.
8. Click **OK**.
9. Optionally, add a frequency sweep to the solution setup.

Related Topics

Add Dependent Solve Setup

Setting Adaptive Analysis Parameters for HFSS

Technical Notes: The HFSS Solution Process

Copying a Solution Setup

Renaming a Solution Setup

Add Dependent Solve Setup

For driven setups (not Eigenmode), to apply all settings from an existing setup to a child setup:

1. Select an existing setup in the project tree.
2. Right click on the setup in the project tree, and then click **Add Dependent Solve Setup** on the shortcut menu.

   A dependent setup icon appears, which has an altered graphic to distinguish it from the parent setup icon. The child setup name is "*parent_setup name*_1." All of the settings from the parent setup are copied to the child setup. The dependent setup uses the mesh from the parent setup. This is shown under the **Advanced** tab of the **Solution Setup** dialog, **Specifying a Source** for the initial mesh. You can add a dependent setup to another dependent setup, and form of the name shows the hierarchical dependence by appending "_1" to show further dependence.

If you intend to change any of the settings, you do this just as you would for a new setup.

The **Enabled** checkbox on General tab permits to you to disable a setup so that it does not run when you select Analyze All. If a solve is disabled, the **Analyze** command on the shortcut menu for the Project tree is disabled and the **Enable Setup** command is available.

Related Topics

Specifying Solution Settings

Renaming a Solution Setup

Do the following to rename a solution setup:

1. In the project tree, under **Analysis**, right-click the setup you want to rename.
   A shortcut menu appears.
2. Select **Rename** from the shortcut menu.
   The setup name text is highlighted in the project tree.
3. Type the new name for the setup, and press **Enter**.

You can also rename the solution setup by changing the text in the Name text box of the appropriate **Solve Setup** dialog box.

Related Topics:

Copying a Solution Setup

Copying a Solution Setup

Solution setups may be copied and pasted within a design or across designs of the same type. This is beneficial for setups having a large number of parameters to specify, or where minor changes to a setup are being evaluated.

Do the following to copy a solution setup:

1. In the project tree, under **Analysis**, right-click the setup you want to copy.

   A shortcut menu appears.

2. Select **Copy** from the shortcut menu.

   The setup parameters are copied to the clipboard.

3. In the project tree, right-click on the **Analysis** folder to receive the copied setup.

   A shortcut menu appears.

4. Select **Paste** from the shortcut menu.

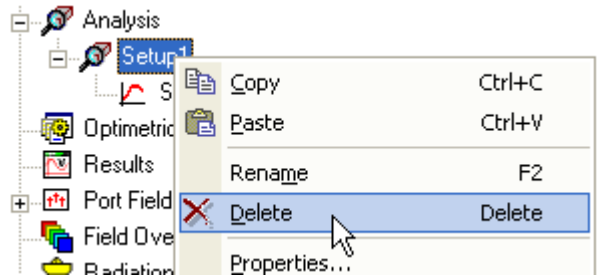
   The setup parameters are copied to the **Analysis** folder as a new setup.

Related Topics:

Renaming a Solution Setup

Deleting a Solution Setup

To delete an existing solution setup, select it in the Project tree, right-click to display the shortcut menu and select **Delete**.



Related Topics

Adding a Solution Setup

Setting the Solution Frequency

For Driven solution types.

For every Driven solution setup, specify the frequency and units at which to generate the solution. If you want to solve over a range of frequencies, define a frequency sweep. If a frequency sweep is solved, an adaptive analysis is performed only at the solution frequency.

- Under the **General** tab of the **Solution Setup** dialog box, enter a value for **Solution Fre-**

quency and select the frequency units from the pull down list.

Note For Fast sweeps, HFSS uses the solution frequency as the center frequency if it is within the frequency range (greater than the start frequency and less than the stop frequency.) Otherwise the middle of the frequency range is used as the center frequency.

Related Topics

Setting Adaptive Analysis Parameters for HFSS

Solving for Ports Only

For Driven solution types with ports.

To quickly compute only the 2D excitation field patterns, impedances, and propagation constants at each port:

- Under the **General** tab of the **Solution Setup** dialog box, select **Solve Ports Only**.

  This disables the remaining settings for **Maximum number of passes** and **Convergence per pass**.

HFSS calculates the natural field patterns (or modes) that can exist inside a transmission structure with the same cross-section as the port. These 2D field patterns serve as boundary conditions for the full 3D problem.

Related Topics

Port Field Display

Technical Notes: Port Solutions

Setting Adaptive Analysis Parameters for HFSS

Setting the Minimum Frequency

For Eigenmode solution types.

For every Eigenmode solution setup, specify the minimum frequency at which to search for eigenmodes. HFSS searches for the user-specified number of modes with a higher resonant frequency than the **Minimum Frequency** value.

- Under the **General** tab of the **Solution Setup** dialog box, type a **Minimum Frequency** and the frequency units.
- You can set the Minimum frequency as a variable by typing a name in the field and pressing Enter. This displays the **Add Variable** dialog for you to enter the value and units. Click OK to close the dialog. The variable is listed in the Setup and in the Design Properties.

Warning Because the minimum frequency is used to normalize some matrices, if the frequency is set too low, HFSS tries to solve a nearly-singular matrix, which may erode the accuracy of the calculations. As a general rule, do not enter a frequency less than 0.01 times the suggested, or default, value for **Minimum Frequency**.

Related Topics

Setting Adaptive Analysis Parameters for HFSS

Setting the Number of Modes

For Eigenmode solution types.

For every Eigenmode solution setup, specify the number of eigenmode solutions that the solver finds. If you enter **5**, the solver calculates the first 5 eigenmode solutions above the minimum frequency.

The Eigenmode solver can find up to 20 eigenmode solutions.

- Under the **General** tab of the **Solution Setup** dialog box, enter a value for **Number of Modes**.

Related Topics

Setting Adaptive Analysis Parameters for HFSS

Setting Adaptive Analysis Parameters for HFSS

When you set up an adaptive analysis, define the following parameters under the **General** tab of the **Solution Setup** dialog box:

- Enter the Solution Frequency and select the frequency units from the pull down list.
- Optionally, select Solve Ports Only.
- Maximum Number of Passes
- Maximum Delta S or Use Matrix convergence (for designs with ports). Here you can set matrix values for convergence, including maximum delta for Mag S and Phase S.
- Maximum Delta Energy for convergence per pass (for designs with voltage sources, current sources, incident waves, or magnetic bias).
- For Eigenmode solutions, specify Maximum Delta Frequency Per Pass and, if desired, Converge on Real Frequency Only.

Under the **Options** tab of the **Solution Setup** dialog box, you can edit the following settings:

- Lambda Refinement
- Maximum Refinement Per Pass
- Maximum Refinement
- Minimum Number of Passes
- Minimum Number of Converged Passes
- Order of Basis functions
- Enable the Direct Solver
- Enable Iterative Solver and associated Relative Residual Setting
- Enable Domain Decomposition

Under the **Advanced** tab of the **Solution Setup**, depending on the solution type, you can edit the following settings.

- Initial Mesh Options for mesh linking
- Port options (Maximum Delta Z_0, whether to Use Radiation Boundary on Ports and Min/Max Port Triangle settings)
- Linked Solve options, including no Link.
- Whether to **Save fields**, and/or whether to save radiated fields only. To view a port field display, you must save fields. Save fields options also occur for Discrete and Fast Sweeps.

Under the **Expression Cache** tab of the **Solution Setup**, you can create and manage expressions to use for adaptive convergence.

Under the **Derivatives** tab of the **Solution Setup**, you can:

- Specify which variables to use for calculating derivatives.

Related Topics

HFSS-IE Feature

Setting the Maximum Number of Passes

The **Maximum Number of Passes** value is the maximum number of mesh refinement cycles that you would like HFSS to perform. This value is a stopping criterion for the adaptive solution; if the maximum number of passes has been completed, the adaptive analysis stops. If the maximum number of passes has not been completed, the adaptive analysis will continue unless the convergence criteria are reached.

To set the maximum number of passes for an adaptive analysis:

- Under the **General** tab of the **Solution Setup** dialog box, enter a value for **Maximum Number of Passes**.

  For driven problems HFSS always requiring at least one adaptive pass. Entering **1** will bypass adaptive analysis, generating a solution only at the solution frequency you specified.

Note The size of the finite element mesh — and the amount of memory required to generate a solution — increases with each adaptive refinement of the mesh. Setting the maximum number of passes too high can result in HFSS requesting more memory than is available or taking excessive time to compute solutions.

Related Topics

Adding a Solution Setup to an HFSS Design

Adding an Solution Setup to an HFSS-IE Design

Setting the Maximum Delta S Per Pass

For designs with ports or Transient Solutions for Device Characterization.

The delta S is the magnitude of the change of the S-parameters between two consecutive passes. The value you set for **Maximum Delta S** is a stopping criterion for the adaptive solution. If the magnitude of the change of all S-parameters are less than this value from one iteration to the next, the adaptive analysis stops. Otherwise, it continues until the requested number of passes is completed.

To set the maximum delta S per adaptive pass:

- Under the **General** tab of the **Solution Setup** dialog box, enter a value for **Maximum Delta S**.

Delta S data is available only after HFSS completes two iterations of the adaptive analysis process.

Note Delta S is computed on the appropriate S-parameters - modal or terminal - after the S-parameters have been de-embedded and renormalized.

Related Topics

Adding a Solution Setup to an HFSS Design

Adding an Solution Setup to an HFSS-IE Design

Viewing the Maximum Magnitude of Delta S Between Passes

Technical Notes: Maximum Delta S

Setting the Maximum Delta Energy Per Pass

For designs with voltage sources, current sources, incident waves or magnetic bias or Transient Solutions for Field Visualization.
Not applicable to designs with ports.

The delta Energy is the difference in the relative energy error from one adaptive solution to the next. The value you set for **Maximum Delta Energy** is a stopping criterion for the adaptive solution. If the delta Energy falls below this value, the adaptive analysis stops. Otherwise, it continues until the convergence criteria are reached.

To set the maximum delta Energy per adaptive pass:

- Under the **General** tab of the **Solution Setup** dialog box, enter a value for **Maximum Delta Energy**.

Delta Energy data is available only after HFSS completes two iterations of the adaptive analysis process.

Related Topics

Viewing the Delta Magnitude Energy

Technical Notes: Maximum Delta Energy

Setting the Maximum Delta Frequency Per Pass

For Eigenmode solution types

The delta Frequency is the percentage difference between calculated eigenmode frequencies from one adaptive pass to the next. The value you set for **Maximum Delta Frequency Per Pass** is a stopping criterion for the adaptive solution. If the eigenmode frequencies change by a percentage amount less than this value from one pass to the next, the adaptive analysis stops. Otherwise, it continues until the maximum number of passes is completed.

To set the **Maximum Delta Frequency Per Pass**:

- Under the **General** tab of the **Solution Setup** dialog, enter a value for **Maximum Delta Frequency Per Pass**.

  Delta Frequency data is available only after HFSS completes two iterations of the adaptive analysis.

Related Topics

Specifying Convergence on Real Frequency Only

Specifying Convergence on Real Frequency Only

For Eigenmode solution types.

Selecting **Converge on Real Frequency Only** causes the percent difference calculation among a set of frequencies to be based only on the real parts of the frequencies; the imaginary parts of the frequencies are ignored.

- Under the **General** tab of the **Solution Setup** dialog box, select **Converge on Real Frequency Only**.

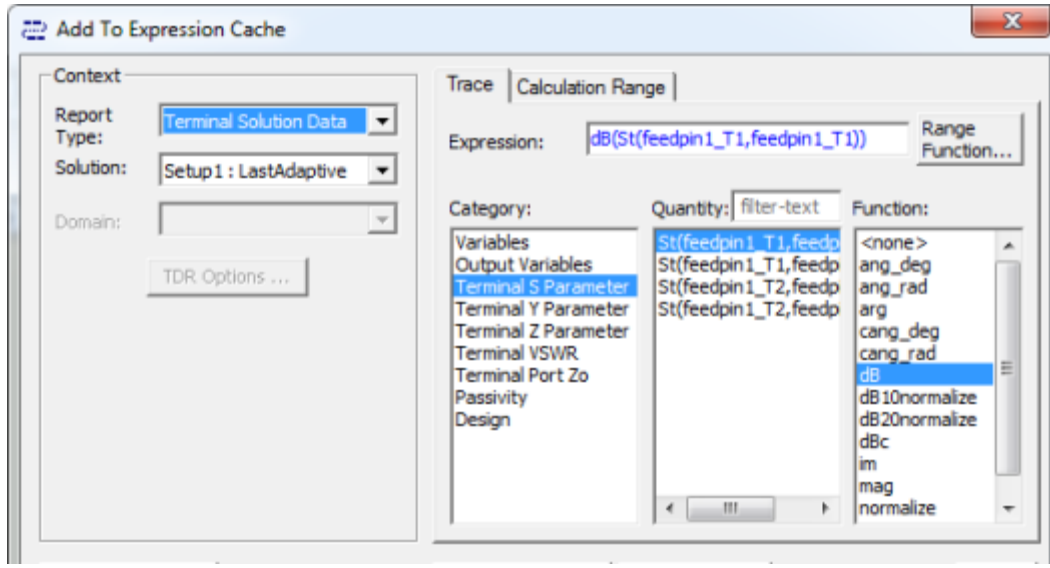
Specifying Expressions for Adaptive Convergence

You can specify additional convergence criteria through the use of expressions and output variables. The **Max Delta** or the **Max Percent Delta** defined for expression convergence represents the difference in values of the expressions between consecutive adaptive passes. If the difference in the value of the expression between consecutive passes is less than the **Max Delta** or the **Max Percent Delta** value this part of the convergence criteria is satisfied.

- For **driven solutions**, if the **Maximum Delta S**, **Maximum Delta E**, or alternate matrix convergence criteria are achieved in addition to any specified expression convergence criteria, the adaptive analysis stops. Otherwise, the solution continues until the requested number of passes is completed.
- For **eigenmode solutions,** if the **Maximum Delta Frequency Per Pass** criteria is achieved in addition to any specified expression convergence criteria, the adaptive analysis stops. Otherwise, the solution continues until the requested number of passes is completed.

To set expressions as **Convergence** criteria:

1. Double-click on the setup icon in the Project tree to open the **Solution Setup** dialog.
2. Click the **Expression Cache** tab in the **Solution Setup** and click the **Add...** button to open the **Add to Expression Cache** dialog.



3. Specify the context for the expression you define. For Modal Solution Data, you do not need context other than the Solution. Other selections require more context:
   - Emission Test also requires a digital signal.
   - Far Field field also requires a geometry such as an infinite sphere.
   - Fields or Near Fields also require a geometry such as a polyline.
4. Under the **Trace** tab, select from the Category, Quantity and Function lists to create expres-

sions.

Selecting a listed category lists the Quantities and Functions available for each category. If you have defined one or more output variables, you can see them listed as Quantities by selecting the **Output Variables** Category. The Output Variables button opens a dialog that lets you define additional output variables.

When you have created an expression, it appears in the Expression field of the **Trace** tab. If desired, you can use the Range Function button to select range functions to apply to the expression.

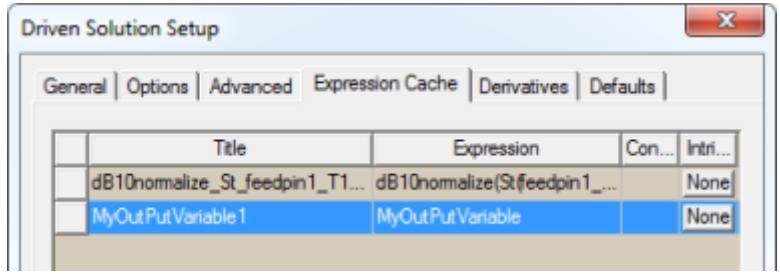
Under the **Calculation Range** tab, you can view the values of available sweep variables. Clicking the ellipsis [...] button in the Edit column opens a list of values.

5. When you have created an expression that you want to add to the cache, click the **Add Calculation** button.

   This adds the selected expression and the associated context to a table in the **Expression Cache** tab. You can select any additional expressions with contexts and add them in the same way.

6. When you have added the expressions you want, click **Done** to close the **Add to Expression Cache** dialog.

   The **Expression Cache** tab of the **Solution Setup** lists the expressions you have added as a table.



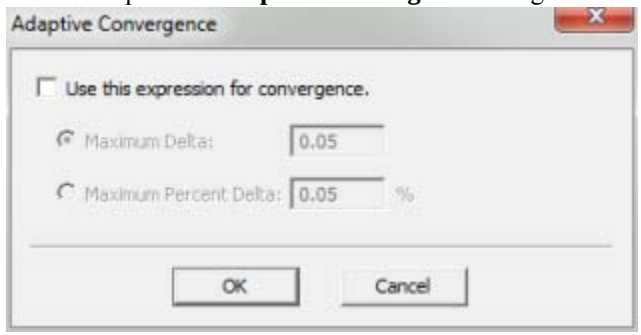
- The Title field is editable, by default showing the name as built from the expression, but removing underscores.
- The Expression field shows the full expression. If necessary, you can resize the **Solution Setup** dialog. You can also resize each column in the table.
- The Context column shows None for Modal solutions, or the appropriate geometry for Fields calculations.
- The Intrinsics column shows a clickable button that opens an **Edit Calculation Range** dialog. If the column button shows None you cannot edit the value. If the button shows variables, click the ellipsis [...] in the Edit column to display a list of the variable values that you can select. Click **OK** to close the **Edit Calculation Range** dialog and apply your selections to the Expression Cache.
- The Convergence column contains a checkbox that lets you designate post processing variables, such as for renormalization or deembedding. You can also use this feature to

assign variables to non-model objects (for example, the properties of a solid, sheet or lines used for field calculations). You can also assign a PP Variable to a coordinate system (CS) that is not associated with model objects.

To excite particular sources for particular output variables, you can assign a post process variable to the magnitude of sources you might want to edit, and assign the expression value in the Expression Cache. With PP Variables, you can control convergence based on several combinations of active sources.

7. To designate one or more expressions for convergence, click the field for the Convergence column for each expression.

   This opens an **Adaptive Convergence** dialog.



   Check **Use this expression for convergence** to enable the radio buttons. You can then specify the **Max Delta** between passes or the **Max Percent Delta** criteria. The **Max Delta** solves a potential issue if your expression is essentially zero and the numeric noise from pass to pass causes the maximum percentage delta to remain high. In that case adaptive refinement continues until you get to the maximum number of passes.

8. Click **OK** to close the **Solution Setup** dialog.

Related Topics

Viewing Convergence Data

Viewing the Output Variable Convergence

Specifying Output Variables

Specifying a Source for the Initial Mesh

You may choose to specify a source for the initial mesh from either the current design or another design. The source mesh should represent a geometrically equivalent model.

Note For more information see Mesh Linking and Mesh Link Requirements.

To specify a source for the initial mesh:

1. Under the **Advanced** tab of the **Solution Setup** dialog box, click the checkbox for **Import Mesh**.

   This displays the **Setup Link** dialog. By default, the **Source Project** checkboxes for **Use This**

Project and the **Source Design** checkbox for **Use This Design** are selected.

Note that when you setup a link to an external source, the **Lamda refinement** option is deselected under the **Options** tab to avoid over-refinement of the linked mesh.

If you uncheck **Use This Project**, you enable fields for selecting a different project and radio buttons to specify whether to save the source path relative to **The project directory of the selected project** or **This project**.

2. To specify a Source Project file click the ellipsis [...] button to open a file browser window.

   When you selected the project, click the **Open** button to accept the project file for the setup. You can use the checkbox to **Open as read only**.

   Use the radio buttons to specify whether to save the source path relative to **The project directory of the selected projec**t or **This project**.

   When you select a Project File, the Design field and the Solution field are filled in with default values, and the drop down menus contain any available Projects and solutions.

   The "Default" solution is the product dependent solution of the first Setup. That is the setup listed first in the source design's project tree (alphanumerical order). A product specific solution of this setup becomes the default solution. In most products, it is LastAdaptive. In a Transient solution type, it is "Transient."

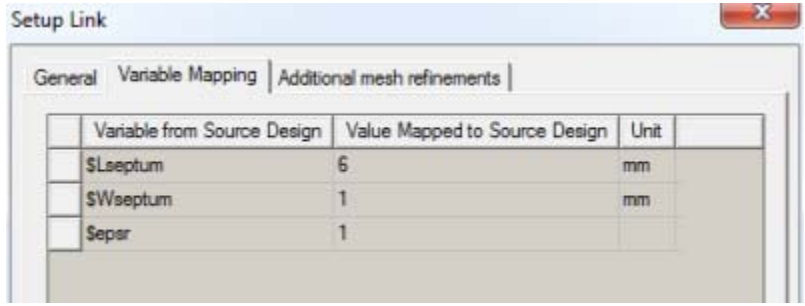
Note The solution in the source design must provide data for the target design's adaptive frequency as well as its sweeps. That is, the adaptive frequency for the target design must be included in the sweep in the source design.

If necessary, you can open the source design and add an appropriate frequency point to an existing sweep.

3. For the Source Design, if you leave **Use This Project** checked, you can uncheck **Use This Design** to enable the a dropdown menu for to select from other available designs.
4. For the **Source Solution** field, you can use the dropdown menu to select from other possible solutions.
5. Use the checkbox specify whether to **Simulate the source design as needed** to produce the mesh.
6. Use the checkbox to specify whether to **preserve the source design solution**. Note that in the **Extractor mode**, the source project will be saved upon exit. **Extractor mode** means that the software is opened during the link solely for the purpose of solving.
7. The second tab, **Variable Mapping,** lets you view any variables contained in the Project you select.

   When there are variables in the source design, you can choose to "map" these variables to constant values, expressions or variables in the target designs. Variable mapping becomes more important when the datalink type requires source and target design to be geometrically identi-

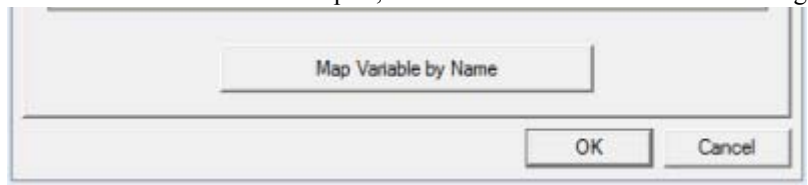
cal and source design is geometrically parameterized.



When a variable in the target design has the same name of a source design variable but the 2 variables are not mapped, the **Parameter** tab will become the active tab with the following message box popped.

You can exit the **Setup Link** dialog directly by selecting "Accept Setup and Exit Dialog" or reexamine the parameter mapping by selecting "Return to Setup Dialog".

In the **Setup Link** dialog, for linked designs with variables of the same name, you can click **Map Variables by Name** to automatically map same named variables. A variable that does not have a same named counterpart, and retains its value in the source design.



Click **OK** to proceed without the variable mapping.

8. You can select the **Additional mesh refinements** tab to specify the following:
   - For Mesh Operations, you can select radio buttons to either **Apply mesh operation in target design on the imported mesh**, or **Ignore mesh operations in target design** (default).
   - For Port Adapt, you can check **Perform port adapt in target solve setup**.
9. Click **OK** to accept the setup and close the **Setup** dialog box.

Related Topics

Clear Linked Data

Setting Lambda Refinement

Clearing Linked Data

Linked data can be mesh, field or some other post-processing data that the source design generated. The target design for the link caches these data internally to minimize the need to activate the source design.

If you have previously setup links to a design, the **HFSS>Analysis Setup** menu contains an option to **Clear Linked Data**. This removes the linked data for all links in a design, therefore invalidating

the solutions. You can also clear linked data through **HFSS>Results>Clean Up Solutions**, which displays a dialog that includes options that let you selectively delete linked data only, or as part of other deletions.

Clearing linked data for some link types requires HFSS to revert to the initial mesh. Thus in some cases, this command removes the current mesh of the target design.

Related Topics

Deleting Solution Data

Mesh Linking

You can obtain a broadband mesh by linking between designs as follows:

- Link a mesh from an Eigenmode solution to a Driven Modal or a Driven Terminal design.
- Use a cascade of mesh links within the same design where each link adapts the mesh at a different frequency.

Related Topics

Mesh Link Requirements

Specifying a Source for the Initial Mesh.

Mesh Link Requirements

If you want to reuse a mesh from an existing source design, make sure that the geometries in the target and source are identical. This is a requirement for a mesh link. There are two ways to generate geometrically equivalent models.

- Copy the entire source design and paste it in the **Project Manager** window to create the target design.
- Use **Import from Clipboard** as follows:
  1. In the model window of the source design, press **CTRL+A** to select all the objects.
  2. Right-click on the Modeler field and select **Edit>Copy** from the short-cut menu.
  3. On the target design, go to **Modeler>Import from Clipboard** to paste in the design.

After you paste using one of the above methods, create a solution setup in the target, select **Import Mesh** on the **Options** tab, and link the mesh from the **Setup Link** dialog box.

Note For more information about linking the mesh, see Specifying a Source for the Initial Mesh.

Setting Lambda Refinement

Lambda refinement is the process of refining the initial mesh based on the material-dependent wavelength. It is recommended and selected by default.

HFSS performs solution adaptive mesh refinement to produce accurate results. Lambda Refinement is necessary to avoid false convergence and achieve accuracy to the field data especially for

electrically (approximately greater than 2*lambda) large problems. The Lambda refinement process helps to start with the optimal balance between element count and element size. For efficient convergence, the initial mesh should meet some element electrical size requirements as determined by lambda refinement. Typically, a few elements per wavelength are needed for accurate results and since wavelength depends upon the material, the element length target differs for each material. You can specify the size of the element target by which HFSS refines the mesh in the Lambda Target field or select the recommended Use Default Value. The default value in the Lambda Target field also depends upon the Order of the Basis Functions. For example, if the lambda target is 0.3333, the initial mesh is refined such that on each solid the length of the elements are approximately smaller than 0.333*wavelength.

If you use the **Advanced** tab to link to a mesh from another design or project, **Do Lamda Refinement** is unchanged.

To specify the size of target value by which HFSS will refine the mesh:

1. Under the **Options** tab of the **Solution Setup** dialog box, select **Do Lambda Refinement**.

   This enables the **Target** field and the **Use free space lambda** check box.

2. Enter a value for the wavelength in the **Target** field or accept the defaults.

   The **Target** defaults depend on the Order of Basis function selections. For example, for Driven solutions and a First Order basis function, the default target is 0.3333, which means that HFSS will refine the mesh until most element lengths are approximately one-third wavelength.

   For eigenmode solutions and a First Order basis function, the default target is 0.2

   If you change the Order of Basis functions in the **Solution Setup** dialog, the default changes automatically. Setting the Order of Basis affects the default value of the Lambda Refinement in the Solution setups as follows.

Zero order:	driven 0.1,	eigenmode 0.1
First order:	driven 0.3333,	eigenmode 0.2 (as is)
Second order:	driven 0.6667,	eigenmode 0.4
Mixed order	driven 0.6667	eigenmode 0.6667

3. If you want the initial mesh to be refined based on the wavelength in free space, select **Use free space lambda**. Material-dependent lambda refinement will be deactivated.

Note Changing the Lambda refinement target invalidates any solutions that were performed with the previous lambda refinement.

If an object is highly conductive, very little energy will penetrate into it, so it does not need a dense mesh. In such cases, free space mesh is applied.

Related Topics

Setting the Max Order of Solution Basis

Specifying a Source for the Initial Mesh

Setting Lambda Refinement for HFSS-IE

Lambda refinement is the process of refining the initial mesh based on the material-dependent wavelength. It is recommended and selected by default.

To specify the size of wavelength by which HFSS-IE will refine the mesh:

1. Under the **Options** tab of the **Solution Setup** dialog box, select **Do Lambda Refinement**.

   This enables the **Target** field and the **Use free space lambda** check box.
2. Enter a value for the wavelength in the **Target** field or accept the defaults.

If you want the initial mesh to be refined based on the wavelength in free space, select **Use free space lambda**. Material-dependent lambda refinement will be deactivated.

Note Changing the Lambda refinement target invalidates any solutions that were performed with the previous lambda refinement.

If an object is highly conductive, very little energy will penetrate into it, so it does not need a dense mesh. In such cases, free space mesh is applied.

Related Topics

Adding a Solution Setup to an HFSS Design

Adding an Solution Setup to an HFSS-IE Design

Setting the Percent Maximum Refinement Per Pass

The value you set for percent **Maximum Refinement Per Pass** determines how many tetrahedra are added at each iteration of the adaptive refinement process. The tetrahedra with the highest error will be refined. The default value is 30%.

To set the percent refinement per adaptive pass:

- Under the **Options** tab of the **Solution Setup** dialog box, enter a value for percent **Maximum Refinement Per Pass**.

Related Topics

Technical Notes: Percent of Tetrahedra Refined Per Pass

Adding a Solution Setup to an HFSS Design

Adding an Solution Setup to an HFSS-IE Design

Setting the Maximum Refinement

This specifies the maximum number of tetrahedra that can be added during an adaptive pass. By default, this unchecked, to that there is no maximum. If you enable the **Maximum Refinement**, the initial value is 1000000.

To set a new value for the Maximum Refinement:

1. Under the **Options** tab of the **Solution Setup** dialog box, click the **Maximum Refinement** checkbox to enable the text field.
2. Enter the number of tetrahedra for **Maximum Refinement**.

You can also control these values in the docked properties window that displays when you select the setup with the **View>Properties** enabled. With the properties window displayed. you can click the checkbox for **Use Max Refinement** to apply the value in the Max Refinement text field.

Related Topics

Adding a Solution Setup to an HFSS Design

Adding an Solution Setup to an HFSS-IE Design

Setting the Minimum Number of Passes

An adaptive analysis will not stop unless the minimum number of passes you specify has been completed, even if convergence criteria have been met.

- Under the **Options** tab of the **Solution Setup** dialog box, enter a value for **Minimum Number of Passes**.

Note For a solve setup with zero passes, no sweeps, and that is not ports only, validation produces a warning message.

Related Topics

Adding a Solution Setup to an HFSS Design

Adding an Solution Setup to an HFSS-IE Design

Setting the Minimum Number of Converged Passes

An adaptive analysis will not stop unless the minimum number of converged passes you specify has been completed.

- Under the **Options** tab of the **Solution Setup** dialog box, enter a value for **Minimum Converged Passes**.

  The convergence criteria must be met for at least this number of passes before the adaptive analysis will stop.

Related Topics

Adding a Solution Setup to an HFSS Design

Setting Matrix Convergence Criteria

For designs with ports.

You can specify different stopping criteria for specific entries in the S-matrix. This is done in the **Matrix Convergence** dialog box. The adaptive analysis will continue until the magnitude and phase of the entries change by an amount less than the specified criteria from one pass to the next, or until the number of requested passes is completed.

To set the matrix convergence:

1. Under the **General** tab of the **Solution Setup** dialog box, select **Use Matrix Convergence**.
2. Click **Set Magnitude and Phase**.

   The **Matrix Convergence** dialog box appears.

3. Select one of the following from the **Entry Selections** pull-down list:

All	Sets all of the matrix entries at once. (The default).
Diagonal/Off-Diagonal	Sets all of the diagonal matrix entries at once, all off-diagonal matrix entries at once, or both diagonal and off diagonal entries at once.
Selected Entries	Sets individual matrix entries that you will select.

   For the selection **All**, enter the convergence criteria for the **Maximum Delta (Mag S)** and the **Maximum Delta (Phase S)** in the fields to the right.

   For the selection **Diagonal**/**Off-Diagonal**, first check **Diagonal Entries**, **Off-Diagonal Entries**, or both, to enable the convergence criteria field or fields. Then enter the convergence criteria for the **Maximum Delta (Mag S)** and the **Maximum Delta (Phase S)** in the fields to the right.

   For both of these **Entry Selections**, you can set the **Ignore Phase when Mag is less than** value. See the note on technical issues under step 4 c.

4. If you chose **Selected Entries**, the Matrix Convergence dialog displays some new fields:
   - a table showing columns for Matrix Entry 1, Matrix Entry 2, and the Delta Mag and Delta Phase.
   - **Entry 1** and **Entry 2** fields which contain drop down lists of ports and associated modes (or terminals).
   - an **Insert** button with which to move selections from the port list selections to the table

   To select the desired ports and mode (or terminal) pairs, do the following:

   a. Select Entry 1 and Entry 2 from their drop down lists.

   b. In the **Magnitude** box, enter the maximum change in magnitude from pass to pass from the Entry 1 to Entry 2.

   c. In the **Phase** box, enter the maximum change in phase, in degrees, from pass to pass from Entry 1 to Entry 2.

      Note: When the Mag S becomes small (near to zero) its phase becomes indefinite and insignificant due to mathematical issues. In HFSS 10 and 11 there was a 0.05 magnitude threshold that caused that Phase Margin to be discarded. However, some users may want to continue running additional passes to stabilize phase margins even when the S-parameter magnitude is below this threshold. For this reason, the magnitude threshold has been removed. This calls for caution if you want to see a good phase convergence for S_{12} which is near to 1. If you set a small delta_phi with small delta_s, after a couple of passes, S_{12} will converge, but S_{11} never, since S_{11} ~0 and its phase changes with the mesh noise because the phase is indefinite. In other words, under these conditions the adaptive process never stops, so you should abort it.

   d. Click **Insert.**

      The entries appear in the table above. If you have selected multiple entries, all combinations of matrix entry1 and matrix entry2 populate the table.

Selecting a Row in the table enables the Delete button, if you need to remove a row from the table.

Clicking in the Delta Mag and Delta Row fields of the selected row enables editing in those fields.

5. Click **OK** to close apply the values and close the dialog.

Related Topics

Viewing the Magnitude Margin

Viewing the Phase Margin

Viewing Delta (Mag S)

Viewing Delta (Phase S)

Setting the Order of Basis Functions

You can change the basis functions HFSS uses to interpolate field values from nodal values.

- Under the **Options** tab of the **Solution Setup** dialog box, select **Order of Basis**.

  This can be First Order (the default), Zero Order, Second Order, or Mixed order.

  Setting the Order of Basis functions affects the default value of the Lambda Refinement in the Solution setups as follows.

Zero order:	driven 0.1,	eigenmode 0.1
First order:	driven 0.3333,	eigenmode 0.2 (as is)
Second order:	driven 0.6667,	eigenmode 0.4
Mixed order	driven 0.6667	eigenmode 0.6667

The Zero order option is useful when a model requires a mesh that produces more than 100,000 tetrahedra, but the model size is small compared to wavelength. The higher order options solve progressively more unknowns for each tetrahedron. Mixed order uses higher order where more accuracy is required, and lower order where fields are weaker.

Warning If you select **Zero Order Solution Basis**, all tetrahedra in the model must have edge lengths less than $1/20^{th}$ wavelength. Thus, this option is usually selected in combination with a specific lambda refinement setting.

Related Topics

Technical Notes: Basis Functions

Setting Lambda Refinement

Enable Iterative Solver

Enable Iterative Solver

The iterative solver provides an alternative to the multi-frontal solver when a matrix is well-conditioned for an iterative solution. The iterative solver significantly reduces memory usage, and it can also provide a savings in the solution time for large simulations.

When you select the **Enable Iterative Solver** option, HFSS automatically invokes the iterative solver when it decides that the matrix is conditioned well enough to take advantage of the iterative approach. HFSS uses the multi-frontal solver if the matrix does not meet this requirement.

For more detail, see the technical notes for Iterative Matrix Solver.

To enable the Iterative solver:

1. On the **Solution Setup** dialog, **Options** tab, select the Iterative Solver radio button.

   This enables the Relative Residual checkbox.
2. Enter a value for the Relative Residual. The residual measures the convergence of the iterative solver. The default value is 1E-4.

Note The Iterative Solver is not available for zero order basis solutions.

Related Topics

Technical Notes: Iterative Matrix Solver

Enable Domain Decomposition

If a problem is too large to solve on one machine HFSS can automatically partition a design into domains that can be solved by separate processes. Before enabling solver domains, you must have the HPC license option, and you must have allocated at least three distributed machines to the solve pool. The number of domains that the solver creates will not exceed N-1, where N is the number of machines listed in the pool (The first machine in the pool acts as the head node and is responsible for domain assembly, mesh refinement, and solution management). If more machines are present in the solve pool than are needed, HFSS creates the number of domains that leads to increased overall solver efficiency. Consequently, some machines remain idle if the problem size does not justify their use. See the Technical Note on domain decomposition for further details.

Note You can use the Domain Decomposition method and setup HPC to solve a large problem such as a helical antenna. HFSS automatically apportions the helical antenna design into domains and solves them by separate processes. See the discussion associated with the helical antenna example project that can be accessed with **File > Open Examples > Antenna**.

Domain use can be invoked for a solve when

- The **Domain Decomposition** radio button under the Solution Setup Options tab is checked.
- You have the HPC License.
- You have provided at least three distributed machines in the pool.
- The solver determines that the problem is large enough (the mesh has enough tets) to bother with domains.
- The design includes IE Regions and/or FEBI Radiation Boundaries.

If you have configured your installations properly, the Domain solve can use Distributed Memory solutions. Memory used by the MPI-enabled HFSS solver is therefore limited by the set of machines that are available rather than the shared memory available on any single machine. This allows you to simulate larger structures than before and to optimally reconfigure the cluster of machines for the problem at hand. If an HFSS problem involves solver domains or a finite array, then frequency sweeps will not be done using DSO. Also, DSO for Optimetrics will not be allowed.

Restrictions on solver domains are that the design and analysis setup cannot include:

- The design cannot contain master and slave boundaries.
- Eigenmode solution type.
- Fast frequency sweeps.

If any condition is not met, the problem is solved with the non-domain solver defined under the solution options. When these conditions are met:

- The solver chooses the number of domains to use, based on the machines available. The number of domains chosen will likely be close to the maximum. The maximum is one less than the total because the first machine in the pool is used to control domain iterations.
- The solver creates the domain meshes of roughly the same size
- Domain meshes are created every time the global mesh changes (before each adaptive pass)

Note

- You provide resources for the distributed solve by adding machines to the distributed machine pool. A machine can appear in the pool more than once. You should use this capability to maintain a balanced load.
- Because the domains should be roughly the same size, you should provide balanced resources.

Related Topics

Distributed Right-Hand-Side Solver in Domain Decomposition

Balancing Resources for Solver Domains

Technical Notes: Domain Decomposition Method

Distributed Right-Hand-Side Solver in Domain Decomposition

For versions prior to HFSS16 in the domain decomposition solver, domains are distributed across separate machines and each right hand side (RHS) or excitation is solved one at a time. For designs with several excitations, this process can take much time. In HFSS16 a distributed right hand side (RHS) solver can use extra machines (if available) and speed up simulation.

Suppose a design with 10 excitations requires 5 domains. To simulate such a design you must enter at least 6 machines in the distributed list. If you enter less than 12 machines, Distributed RHS launches 6hf3ds and the simulation process is identical to that in the previous versions. These 6 hf3d solvers are treated as one group. If you enter 12 machines, you will notice 12 hf3ds (two groups) in the Task Manager and each group simultaneously solves 5 distinct excitations. If each

machine is distinct, the simulation is faster compared to the single group simulation. More groups are automatically created when more machines are entered.

The distributed RHS solver is triggered when there are at least 4 excitations and adequate number of machines available. The minimum number of excitations per group is 2. For finite array domains, distributed RHS is enabled when there are adequate number of machines such that each engine solves at most 2 parents. For regular rectangular arrays, number of engines per group is 11. Thus you can enter at least 22 machines to enable this new feature. For optimal performance, we recommend using distinct machines in each group.

Related Topics

Configuring Distributed Analysis

Selecting an Optimal Configuration for Distributed Analysis

Balancing Resources for Solver Domains

Technical Notes: Domain Decomposition Method

Balancing Resources for Domains

The Domain Decomposition option for large models requires that you provide resources for the distributed solve by adding machines to the distributed machine pool.

Because the domains should be roughly the same size, you should provide balanced resources. As an example of balancing the load, suppose that machine called Patriot has 256 GB and another called Cutlass has 64 GB. If you want up to five domains, it makes sense to enter Patriot in the pool four times and Cutlass once.

Under Analysis Configuration Options, you should set the **Desired RAM Limit (Mem_Limit_Soft)** to the memory desired for each individual domain.

In the example above, a good choice for Mem_Limit_Soft would be 64 GB.

Remember:

- A machine can appear in the pool more than once. The user should use this capability to maintain a balanced load.
- The first machine in the pool is used to control domain iterations.
- Under the Analysis Configuration Machine tab, the **Number of Cores** is also set per domain.

In the example above, a machine named Patriot has 16 processors and a machine named Cutlass has 8 processors. Four domains will go on Patriot, so **Number of Processors** should be set to 4. Four processors on Cutlass will be unused.

Recommendation: it is more important to use memory efficiently than to use all the processors.

Related Topics

Configuring Distributed Analysis

Selecting an Optimal Configuration for Distributed Analysis

Distributed Right-Hand-Side Solver in Domain Decomposition

Technical Notes: Domain Decomposition Method

Domains with FEBI Radiation Boundaries or IE Regions

The solve uses domain solvers if there is a FEBI radiation boundary and/or at least one IE Region.

- Solve sequentially using domains if solver domains are not enabled
- Solve distributed using domains if solver domains are enabled, even if mesh not big enough to split into solver domains

Number of domains

- One or more for FEM region (depending on whether using solver defined domains)
- One or more for FEBI surfaces and dielectric IE Regions, depending on size and connectivity
- One or more for metallic IE Regions, depending on size and connectivity
- With solver defined domains, determines number of machines

Related Topics

Enable Use of Solver Domains

Assign IE Regions

Assigning Radiation Boundaries

Technical Notes: Radiation Boundaries

Use Radiation Boundary on Ports

If the design includes waveports, the **Use Radiation Boundary on Ports** option is enabled under the **Advanced** options tab of the **Solution Setup** dialog box.

- If you select this setting, edges which are assigned to ABC and touch a port have an radiation boundary condition applied during the port solution.
- If you do not select the setting, a perfect conducting boundary condition is used during the port calculations.

In most cases this setting has a limited effect on the overall fields or post processed quantities.

Port Options

If the design includes waveports, the **Port Options** options appear under the **Advanced** options tab of the **Solution Setup** dialog box. These options include:

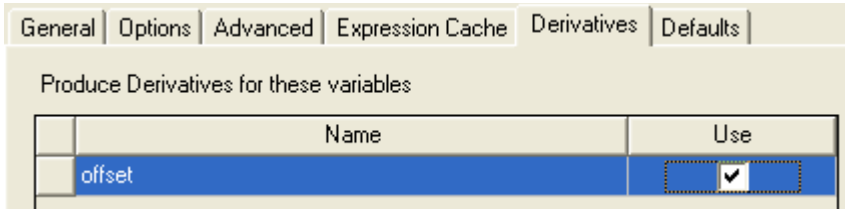
- **Maximum Delta Zo** - change to Zo specified as a target percentage. The default is 2%.
- Use Radiation Boundaries on Ports
- **Set Triangles for Wave Port -** unchecked by default.

  If you check **Set Triangles for Wave Port**, the **Minimum** and **Maximum** fields are enabled. You can edit the default values of 100 for the minimum and 500 for the maximum.

  For designs with lumped ports, this option is not active. Higher numbers of triangles would not benefit a solution setup in this case.

Produce Derivatives for Selected Variables

The Derivatives feature produces derivatives of S-parameters and related matrix quantities such as Y or Z. You can select design properties and project variables as variables of differentiation. If your design has appropriate candidate variables, the **Derivatives** tab of the Solution setup lists them. You enable the derivative calculation by checking the appropriate boxes.



To have HFSS calculate derivatives for design or project variables:

1. Open the **Derivatives** tab of the **Solution Setup**
2. For the desired variable, click the checkbox in the Use column.

You can view the computed derivatives in the Reporter. You can also use them through the Tune Reports command for interactive exploration of small variations in the design, without the need to solve again. Furthermore, for Optimetrics, the SNLP optimizer will take advantage of the derivatives when they are available, which can reduce the number of simulations needed to find the optimum.

This feature has the following limits on use:

- HFSS Transient projects do not support derivatives.
- HFSS Eigenmode projects do not support derivatives.
- Derivatives are not computed in a ports-only solution.
- Parameters of differentiation may not affect Floquet ports.
- Parameters of differentiation many not affect deembedded Lumped ports.
- Derivatives are not supported for any solution employing Domain Decomposition Method (DDM), which includes FEBI radiation boundaries, IE Regions, Finite Arrays.

Related Topics

Setting Adaptive Analysis Parameters

Derivative Tuning for Reports

Examples: Tune a Coax Fed Patch

Technical Notes: Overview of the Technical Approach for Derivatives in in HFSS

Adding a Frequency Sweep

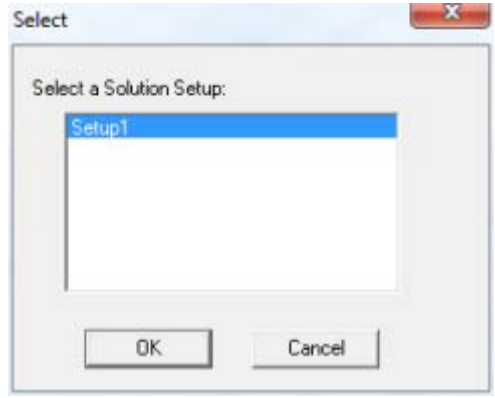
For Driven solution types.

To generate a solution across a range of frequencies, add a frequency sweep to the solution setup. HFSS or HFSS-IE performs the sweep after the adaptive solution. If an adaptive solution is not requested, the sweep is the only solution generated. You can also disable a sweep, so that you can run only the adaptive solution (or a ports-only solution) without the sweep, then later reactivate the sweep definition.

To add a frequency sweep:

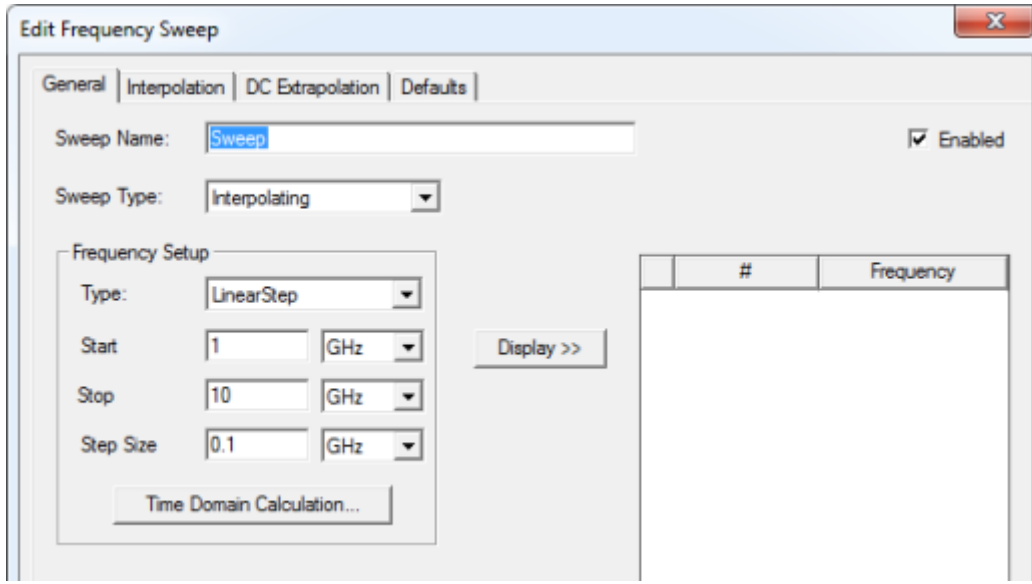
1. Click **HFSS**>**Analysis Setup**>**Add Frequency Sweep**  .

   A dialog appears which lists the solution setups.



2. Select the solution setup to which the sweep applies and click **OK**. To bypass this dialog right-click on the desired setup in the Project tree and select **Add Sweep** from the shortcut menu.

The **Edit Frequency Sweep** dialog box appears.



3. Specify the following sweep parameters:
   - Sweep type - Discrete, Fast, or Interpolating (the default). Each selection affects the tabs and options available.
   - Frequency Setup.
4. If you plan to perform a Full-Wave SPICE analysis, click Time Domain Calculation tool to obtain assistance determining a suitable frequency sweep range for the solutions. Also see the Requirements for Full-Wave SPICE.
5. If you expect to use the same settings often, you can got to the Defaults tab and click the **Set Defaults** button. To use previously saved settings, click the **Use Defaults** button.
6. Click **OK**.

   Once you have created a sweep, an icon for the sweep appears in the Project tree under the associated setup. Selecting the sweep icon causes the docked properties dialog to show the name, start, stop, step size, and sweep type. It also includes an **Enabled** checkbox.

Edit, Copy and Paste Existing Sweeps.

You can select an existing sweep, use the **Edit** commands to **Copy** it, and then and **Paste** the sweep into the Project tree. (By default, the copy is named Sweep*n*, where *n* increments with each new sweep.) You can edit the new copies of the sweep to make desired changes. For example, you can change a specific parameter, or for a distributed solve, you could assign different start and end points for each copy of the setup.

The **Paste** command for sweeps is design sensitive (that is, you cannot paste between Driven and Eigenmode designs) and context sensitive (for example, a sweeps can only be pasted in a setup.)

Dependent setups are pasted along with the copied setup. You are warned if the dependent setup is already in the design and setup is not pasted again.

Note For a solve setup with zero passes, no sweeps, and that is not ports only, validation produces a warning message.

Related Topics

Disabling a Frequency Sweep

Technical Notes: Frequency Sweeps

Selecting the Sweep Type

For Driven solution types.

Specify the type of sweep you want to perform in the **Edit Sweep** dialog box. Choose one of the following sweep types:

Fast Sweep Options	Generates a unique full-field solution for each division within a frequency range. Best for models that will abruptly resonate or change operation in the frequency band. A Fast sweep will obtain an accurate representation of the behavior near the resonance. Fast sweeps are disabled if an anisotropic boundary condition is present.
Discrete Sweep Options	Generates field solutions at specific frequency points in a frequency range. Best when only a few frequency points are necessary to accurately represent the results in a frequency range.
Interpolating Sweep Options (default)	Estimates a solution for an entire frequency range. Best when the frequency range is wide and the frequency response is smooth, or if the memory requirements of a Fast sweep exceed your resources. All discrete basis solutions are solved prior to interpolating sweeps because it is possible that an interpolating sweep can re-use already solved frequencies from a discrete sweep. For Time Domain Reflectometry plots (TDR), you must use an interpolating sweep. When you select Interpolating sweeps in the **Edit Sweep** dialog, the **Interpolation** options tab is activated. This lets you specify a maximum number of solutions, and other interpolation values.

If you expect to use the same option settings often, you can click the **Set Defaults** button. To use previously saved settings, click the **Use Defaults** button.

Related Topics

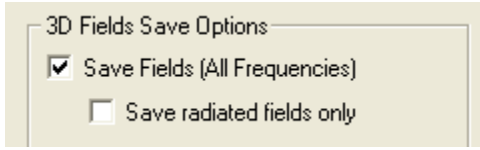
Adding a Frequency Sweep

Technical Notes: Frequency Sweeps

Options for Discrete Sweeps

For Discrete sweeps, the **Edit Sweep** dialog options you can set include

- Sweep Name
- Frequency Setup
- Whether to **Save Fields** (for all Frequencies). By default, all frequencies are saved. (This field is disabled under a Solve Ports Only setup. You can view port fields for the discrete frequencies under the port field display in the project tree.)
- For Discreet sweeps, you can specify whether you want to save the 3D fields.



If you expect to use the same settings often, you can click the **Set Defaults** button. To use previously saved settings, click the **Use Defaults** button.

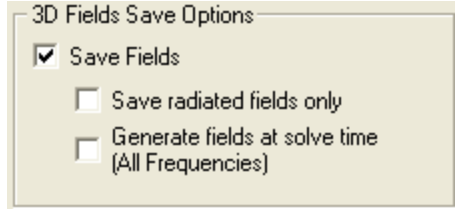
Related Topics

Adding a Frequency Sweep

Options for Fast Sweeps

For Fast sweeps, the **Edit Sweep** dialog options you can set include:

- Sweep Name
- Frequency Setup
- For Fast Sweeps, you can select Save Fields, and whether to Generate fields at solve time (All frequencies).



By default, all fields are saved. (This field is disabled under a Solve Ports Only setup. You can view port fields for the discrete frequencies under the port field display in the project tree.)

Selecting **Save Fields** enables the checkbox for **Generate Fields for all frequencies while solving**. By default, fields are not generated. If you have more than 100 frequencies, checking the box generates a warning that disk space use may be excessive.

If you select this option, HFSS solves the fast sweep and then computes the fields at each freq in the sweep, and saves them. This has two effects: (a) the simulation takes longer due to saving the fields, (b) Post processing is much faster.

Since this option is exercised at solve time, it doesn't apply to existing solutions. Enabling this

option for a previously solved sweep and re-solving will access the previously solved data and generate the requested fields.

- DC Extrapolation options (appearing depending on the Design Settings for DC Extrapolation)

If you expect to use the same settings often, you can click the **Set Defaults** button. To use previously saved settings, click the **Use Defaults** button.

Related Topics

Adding a Frequency Sweep

Options for Interpolating Sweeps

For Interpolating sweeps, the **Edit Sweep** dialog options you can set include:

General tab

- Sweep Name
- Frequency Setup

Interpolation tab

- Max Solutions
- Error Tolerance
- Click the **Advanced Options...** button to open the Interpolating Sweep Advanced Options dialog.

DC Extrapolation tab (appearing depending on the Design Settings for DC Extrapolation)

- DC Extrapolation option

If you expect to use the same settings often, you can click the **Set Defaults** button on the Defaults tab. To use previously saved settings, click the **Use Defaults** button.

Related Topics

Adding a Frequency Sweep

DC Extrapolation options

If the Design Setting for DC Extrapolation is set to Advanced, the **DC Extrapolation** tab appears for Fast and Interpolating sweeps. Otherwise, an effective DC value is calculated.

1. Setting the Initial Frequency to 0 enables DC extrapolation for a sweep.
2. Enter a value for the **Minimum Solved Frequency**. This value represents the smallest frequency in the sweep for which a full solution is generated. The default is 100 Mhz.

Note The Minimum Solved Frequency that you enter is solved independently of the step size.

If you expect to use the same settings often, you can click the **Set Defaults** button. To use previously saved settings, click the **Use Defaults** button.

Related Topics

Options for Interpolating Sweeps
Options for Fast Sweeps

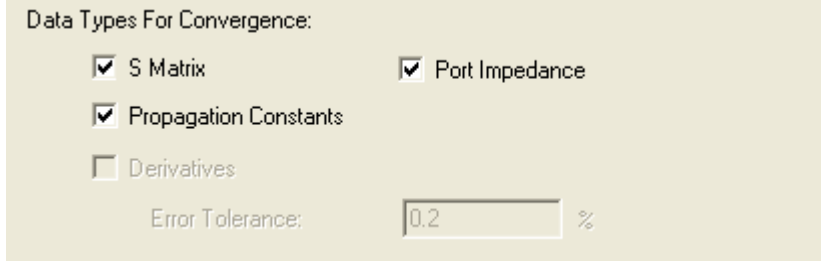
Setup Interpolating Sweep Advanced Options

For HFSS and HFSS-IE Interpolating sweeps, the **Setup Interpolating Sweep Advanced Options** dialog lets you specify the following settings for a sweep:

- The **Minimum Solutions** value is the minimum number of converged solutions that will be solved for the frequency range with respect to the interpolation basis. For example, if this value is three, and the basis element is 1, then once the sweep reaches convergence it simulates at two extra frequencies. If the sweep used a higher basis element setting, that is taken into account. This resembles the minimum number of converged adaptive passes in a regular simulation. Setting a minimum number of solutions can eliminate non-physical S-parameter spikes and oscillations. For interpolating sweeps the default is 0. To change the value: type a new value in the Minimum Solutions box.

- Specify a **Minimum Number of Sub Ranges**. This number acts as an initial condition on the sweep to force initial even breakup of the null range into sub ranges. The end points and middle of each subrange will be solved. This controls the points at which the interpolating sweep is broken up and prevents redundant effort caused by neighboring interpolating sweeps solving the same point. For example, the 1GHz to 4GHz and the 4GHz to 9 GHz sweeps do not both solve the 4 GHz data point.

- Whether to use all or selected entries in the matrix of data types for the convergence. To choose, click the **Select Entries** button to display the Interpolation Basis Convergence dialog.
- The Data Types for Convergence. The defaults selections differ depending on the solution setup.



For Driven modal, Network Analysis 3D Solution Interpolating sweeps:

- S-Matrix - checked
- Port Impedance - unchecked.
- Propagation constants - checked.

For Driven terminal, Network Analysis 3D Solution Interpolating sweeps:

- S-Matrix - checked
- Port Impedance - checked.
- Propagation constants - checked.

For Driven modal, Network Analysis ports-only, interpolating

- S-Matrix - unchecked
- Port impedance - checked
- Propagation constants - checked

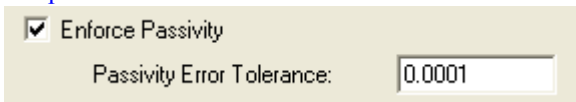
For Driven terminal, Network Analysis ports-only, interpolating

- S-Matrix - unchecked
- Port impedance - checked
- Propagation constants - checked

If the design contains them, you can select **Derivatives**. If you select **Derivatives**, you can also set the **Error Tolerance**, overriding the tolerance specified in the Edit Sweep dialog..

Note If a driven setup's ports-only setup changes and then the problem type switches between driven modal and driven terminal, HFSS resets the interpolation basis data types for the interpolating sweep.

Enforce Passivity -- This applies to driven modal and Terminal solutions. Select this checkbox to enable passivity enforcement of an interpolating sweep at the requested sampling frequencies with the specified error tolerance. For instance, tolerance of .001 means simulation should attempt to reduce passivity to less than 1.001 at all sampling frequencies. You can view passivity using the reporter.



When using passivity, you may want to increase the Max Solutions value.

During an interpolating sweep, HFSS first adds basis points (solution frequencies) until the convergence goal is reached, and then tests the sampling frequencies using the converged sweep for whether they meet the specified passivity requirement. The sampling frequency having the worst passivity violation is selected for solving. After solving for a passivity violation frequency, the solver adds the newly solved data to the interpolation basis.

If necessary, the solver returns to solving for convergence, limited by the maximum number of solutions for the sweep and resumes passivity checking after reestablishing convergence. Messages in the Solution Profile indicate when the interpolated solver is working towards convergence or working to resolve passivity. If any of the bases violates the passivity, the current sweep terminates with a warning message.

Enforce Causality -- This applies to Terminal solutions. Causality means that an output signal cannot start to change before the input signal changes. A design should follow "Best Practices for

Accurate and Causal Broad Band Frequency Sweeps." In addition, you can select "Enforce Causality" so the sweep fits (rather than interpolates) the s-parameter results to a set of causal basis functions, thereby ensuring the extraction of a causal model..

Note You should follow recommended guidelines whenever setting up a numerical model in HFSS (provide correct material definitions, including information about frequency dependent behavior, and defining appropriate boundary conditions). Also when extracting s-parameter models for signal integrity designs, where providing passive and causal results is important, besides using this option, see "Best Practices for Accurate and Causal Broad Band Frequency Sweeps".

Note If you start and stop an interpolated sweep and change the convergence value and/or passivity tolerance, you may generate a non-repeatable sequence of frequencies in the basis. Both regular and passivity frequencies are used when re-evaluate convergence.

If you change sampling frequency delta subsequent passivity behavior may be totally different, since it is evaluated at new sampling frequencies.

Related Topics

Adding a Frequency Sweep

Options for Interpolating Sweeps

Best Practices for Accurate and Causal Broad Band Frequency Sweeps

Setting the Error Tolerance

For Interpolating sweeps.

The **Error Tolerance** value is the maximum relative difference allowed between two successive interpolation solutions. The default 0.5 percent for interpolating sweeps is usually satisfactory.

To set the error tolerance for an Interpolating sweep:

1. Open the **Edit Sweep** dialog box (by either viewing the properties of an existing Sweep or by Adding a Frequency sweep to an existing Setup).
2. Type a value in the **Error Tolerance** box.

If you expect to use the same settings often, you can click the **Set Defaults** button. To use previously saved settings, click the **Use Defaults** button.

Related Topics

Options for Interpolating Sweeps

Setting the Maximum Number of Solutions

For Interpolating sweeps.

The **Max Solutions** value is the maximum number of solutions that will be solved for the frequency range. For fast sweeps and for interpolating sweeps the default is 250. The value must be greater than 3. To change the value:

1. Open the **Edit Sweep** dialog box (by either viewing the properties of an existing Sweep or by Adding a Frequency sweep to an existing Setup).
2. Type a value in the **Max Solutions** box and click **OK**.

If you expect to use the same settings often, you can click the **Set Defaults** button. To use previously saved settings, click the **Use Defaults** button.

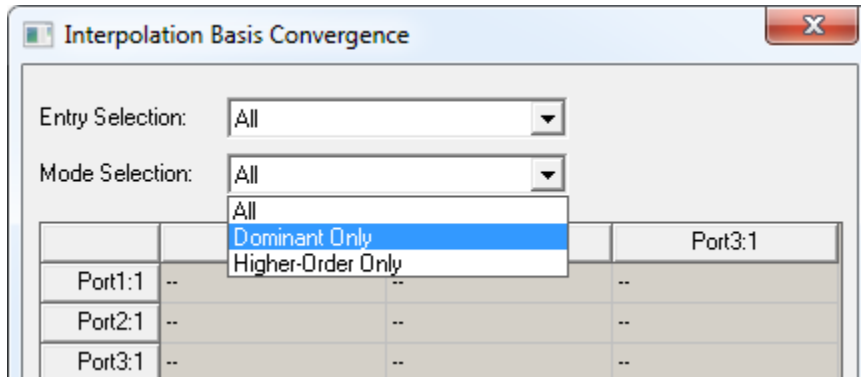
Note HFSS automatically subdivides the interpolating sweep range so that no single subrange gets too many basis elements. The effect is that you can now (if appropriate) request hundreds of basis elements in the Max Solutions box for interpolating sweep setup, without incurring any basis seeding performance penalty.

Related Topics

Options for Interpolating Sweeps

Interpolation Basis Convergence

From the Setup Interpolations Basis dialog, select the **Use Selected Entries** radio button to enable the **Select Entries** button. Select this to display the **Interpolation Basis Convergence** dialog. This dialog permits you to specify the convergence basis.



1. Select one of the following from the **Entry Selections** pull-down list:

All	Selects all of the matrix entries at once. (The default).
Diagonal	Selects all of the diagonal matrix entries at once.
Off-Diagonal	Selects all of the off-diagonal matrix entries at once.

2. If you chose **All**, **Diagonal**, or **Off-Diagonal**, you may fine-tune the matrix entry selection

process by selecting one of the following options from the **Mode Selection** pull-down list:

All	Selects all of the mode matrix entries. Select in conjunction with **All**, **Diagonal**, or **Off-Diagonal** entry selections.
Dominant Only	Selects only the dominant mode matrix entries. Select in conjunction with **All**, **Diagonal**, or **Off-Diagonal** entry selections.
Higher Order Only	Selects only the higher-order mode matrix entries. Select in conjunction with **All**, **Diagonal**, or **Off-Diagonal** entry selections.

As you select the waveports for convergence, you use the **Set**, **Clear**, and **Clear All** buttons in connection with the **Entry Selection** and **Mode Selection** settings. These buttons are enabled when the waveport matrix state and selection settings permit them do something. For example, The **Clear** button is not enabled until there are entries in the waveport matrix to clear and those selections are permitted by the entry selection. The **Set** button is not enabled unless the available mode selections permit entries to be set.

You can also select individual entries in the waveport matrix by clicking on grid cells. This action displays a dropdown menu that lets you select ON or "-".

3. Specify **Entry Selections** and **Mode Selections** as desired and click **SET**, or click individual waveport cells and select ON.

   In the table location corresponding to the selection, the dash in the display is replaced by ON. For example, selecting the first element in the row list and the fourth element in the column list, and then **Set Selection** places an ON in the first row, fourth column. You can select one entry at a time via the dropdown in the matrix cell, or clear the entire table with the **Clear All** button. You can also **Clear** only the entries specified by the **Entry** and **Mode** selection settings (such as off-diagonal, higher order).

4. Click **OK** to close apply the selections and close the dialog.

Related Topics

Setup Interpolations Basis

Options for Interpolating Sweeps

Specifying the Frequency Points to Solve

You can specify the following types of frequency points to solve within a frequency sweep:

Linear Step	A linear range of frequency points in which you specify a constant step size.
Linear Count	A linear range of frequency points in which you specify the number, or count, of points within the frequency range.
LogScale	A logarithmic range of frequency points in which you specify a frequency range and a samples number.
Single Points	Individual frequency points. *For Discrete sweeps*.

Select the type of frequency point entry from the **Type** pull-down list **The Edit Sweep** dialog contains a Time Domain Calculation tool that you can use to help calculate frequency step sizes and maximum frequencies, particularly if you intend to perform Full-Wave Spice analysis.

For Discrete sweeps, HFSS always solves the frequencies from highest to lowest.

Related Topics

Change the Value of an Existing Frequency Point

Specifying Single Frequency Points

Deleting Frequency Points

Insert Frequency Points

Specifying Frequency Points to Solve

Specifying Frequency Points with a Linear Step Size

1. In the **Edit Sweep** dialog box, click **Linear Step** in the **Type** pull-down list.
2. In the **Start** text box, type the starting frequency of the frequency sweep.

   HFSS solves the solution beginning with the frequency entered in the **Start** box and ending with the frequency entered in the **Stop** box. For Fast and Interpolating sweeps, setting the initial value to 0 enables the DC Extrapolation option.
3. In the **Stop** text box, type the ending frequency of the frequency sweep.
4. In the **Step Size** box, type the difference between frequency points.

   HFSS will solve the frequency point at each step in the specified frequency range, including the start and stop frequencies.

   For example, specifying **10** for the start frequency, **20** for the stop frequency, and **2.5** for the step size for a Discrete sweep instructs HFSS to compute a solution for frequencies of 10, 12.5, 15, 17.5, and 20.

   To view a table of the frequencies and count, click the **Display** button. If the list of frequencies is longer than the visible display, you can use a scroll bar to view the list to the end of the count.
5. For Fast sweeps, select **Save Fields** if you want to save the calculated 3D field solutions associated with all port modes at the chosen frequencies.

   For Discrete sweeps, select **Save Fields (All Frequencies)** if you want to save the calculated 3D field solutions associated with all port modes at the chosen frequencies.

   If want to save the fields for just one or a few Discrete sweep frequencies, click the Display button to show a table of frequencies, and select **Single Points** from the **Type** pull-down list. This disables the **Save Fields** checkbox for the entire sweep, and adds a **Save Fields** column with a checkbox for each frequency. You can select the **Save Fields** check box for the desired frequency, or click the **Save Fields** column header to check or uncheck all frequencies.

For Discrete sweeps, HFSS always solves the frequencies from highest to lowest.

Related Topics

Specifying Frequency Points to Solve

Specifying Single Frequency Points

Deleting Frequency Points

Insert Frequency Points

Change the Value of an Existing Frequency Point

Specifying Frequency Points to Solve

Specifying a Linear Count of Frequency Points

1. In the **Edit Sweep** dialog box, click **Linear Count** in the **Type** pull-down list.
2. In the **Start** text box, type the starting frequency of the frequency sweep.

   HFSS solves the solution beginning with the frequency entered in the **Start** box and ending with the frequency entered in the **Stop** box. For Fast and Interpolating sweeps, setting the initial value to 0 enables the DC Extrapolation option.
3. In the **Stop** text box, type the ending frequency of the frequency sweep.
4. In the **Count** text box, type the number of points in the sweep. The count value includes the start and stop values.

   HFSS will divide the frequency range into the count you specify and solve each frequency point in the count.

   To view a table of the frequencies and count, click the **Display** button. If the list of frequencies is longer than the visible display, you can use a scroll bar to view the list to the end of the count.
5. For Fast sweeps, select **Save Fields** if you want to save the calculated 3D field solutions associated with all port modes at the chosen frequencies.

   For Discrete sweeps, select **Save Fields (All Frequencies)** if you want to save the calculated 3D field solutions associated with all port modes at the chosen frequencies.

   If want to save the fields for just one or a few Discrete sweep frequencies, click the Display button to show a table of frequencies, and select **Single Points** from the **Type** pull-down list. This disables the **Save Fields** checkbox for the entire sweep, and adds a **Save Fields** column with a checkbox for each frequency. You can select the **Save Fields** check box for the desired frequency, or click the **Save Fields** column header to check or uncheck all frequencies.

For Discrete sweeps, HFSS always solves the frequencies from highest to lowest.

Related Topics

Specifying Frequency Points to Solve

Specifying Single Frequency Points

Deleting Frequency Points

Insert Frequency Points

Change the Value of an Existing Frequency Point

Specifying Frequency Points to Solve

Specifying a Logarithmic Spaced Frequency Sweep

For Discrete sweeps.

1. In the **Edit Sweep** dialog box, click **LogScale** in the **Type** pull-down list.
2. In the **Start** text box, type the starting frequency of the frequency sweep.

   HFSS solves the solution beginning with the frequency entered in the **Start** box and ending with the frequency entered in the **Stop** box. For Fast and Interpolating sweeps, setting the initial value to 0 enables the DC Extrapolation option.
3. In the **Stop** text box, type the ending frequency of the frequency sweep.
4. In the **Samples** text box, specify the number of frequency points to sample.

   HFSS assigns the sampled points using intervals based on a logarithmic scale.

   To view a table of the frequencies and count, click the **Display** button. If the list of frequencies is longer than the visible display, you can use a scroll bar to view the list to the end of the count.
5. Select **Save Fields (All Frequencies)** if you want to save the calculated 3D field solutions associated with all port modes at the chosen frequencies.

For Discrete sweeps, HFSS always solves the frequencies from highest to lowest.

Related Topics

Specifying Frequency Points to Solve

Specifying Single Frequency Points

Deleting Frequency Points

Insert Frequency Points

Change the Value of an Existing Frequency Point

Specifying Frequency Points to Solve

Specifying Single Frequency Points

For Discrete sweeps.

To specify Single Frequency points:

1. In the **Edit Sweep** dialog box, click **Single Points** in the **Type** pull-down list.

   The **Frequency setup** changes to single point mode.
2. In the **Single** text box, type a desired frequency point, and then select the frequency units.

   This enables the **Insert** button.

3. Click **Insert**.

   The point is added to the **Frequency** column to the right.

   A check mark in the **Save Fields** column indicates that the fields for the point will be saved. Optionally, click the check box in the **Save Fields** column.

4. Repeat steps 2 - 4 for each frequency point you want to solve.

For Discrete sweeps, HFSS always solves the frequencies from highest to lowest.

Related Topics

Change the Value of an Existing Frequency Point

Deleting Frequency Points

Insert Frequency Points

Specifying Frequency Points to Solve

Change the Value of an Existing Frequency Point

To change the value of an existing frequency point in a discrete sweep in the **Edit Sweep** dialog:

1. Click the **Display** button in the **Edit Sweep** dialog to show the frequency and count table for the sweep. If the list of frequencies is longer than the visible display, you can use a scroll bar to view the list to the end of the count.
2. Select **Single Points** from the **Type** pull-down list in the **Edit Sweep** dialog.

   The **Frequency setup** changes to single point mode.
3. Either select the text field in the Frequency column and edit an existing value field directly, or:
   a. Edit the value in the **Single** Text box.
   b. Select the Frequency row that you want to change.

      This enables the **Change** button.
   c. Click the **Change** button to replace the selected Frequency row value with the Single Text box value.
4. Select **Save Fields** if you want to save the calculated 3D field solutions associated with all port modes at that frequency.
5. Repeat for changing additional points.

Related Topics

Specifying Single Frequency Points

Deleting Frequency Points

Insert Frequency Points

Specifying Frequency Points to Solve

Deleting Frequency Points

1. To delete a single frequency point from a sweep, you must click the **Display** button in the **Edit**

Sweep dialog to show the frequency and count table for the sweep. If the list of frequencies is longer than the visible display, you can use a scroll bar to view the list to the end of the count.

You can select multiple frequencies with SHIFT-click or CTRL-click, and delete all.

2. Select **Single Points** from the **Type** pull-down list in the Edit Sweep dialog.

   The **Frequency setup** changes to single point mode.

3. Select the row containing the frequency you do not want to solve.

   This enables the **Delete** button.

4. Click **Delete**.

Related Topics

Specifying Single Frequency Points

Inserting Frequency Points

Change the Value of an Existing Frequency Point

Specifying Frequency Points to Solve

Inserting Frequency Points

For Discrete sweeps, you can insert, change, or delete specific frequency points that you want to solve in the frequency range. They can be inserted individually to a new display, or to an existing display of uniform frequency points.

1. To pre-specify an initial uniform range of frequency points, use the Frequency setup of the **Edit Sweep** dialog to define the linear step, linear count, or log scale range you want, and click the **Display** button to list the count and frequency of those points as a table. For longer frequency tables, you can use a slider bar to scroll to through the complete list to see counts and values.

2. Select **Single Points** from the **Type** pull-down list.

   The **Frequency setup** changes to single point mode. If you previously specified a range of frequency points, and clicked the **Display** button, the frequency table retains those points. This selection also enables the **Insert** button between the **Frequency setup** and the frequency table. The frequency points table now includes another column that contains a **Save Field** checkbox for each row.

   Note that selecting a row highlights it and enables the Delete button under the **Insert** button. Clicking the cursor in the Frequency columns of the frequency points table makes their values editable.

3. In the **Single** text box, type a desired frequency point in the frequency units.

   This enables the **Insert** and **Change** button.

4. Click **Insert**.

   This inserts the new point in ascending order for the list and the count for each row adjusts. Clicking Change would replace the frequency value in the selected row.

5. Select the **Save Fields column** header if you want to save the calculated 3D field solutions associated with all port modes at that frequency. You can also select, or deselect the checkbox

to specify individual frequencies.

6. Add or edit additional frequency points, and then click **OK** to save the sweep.

For Discrete sweeps, HFSS always solves the frequencies from highest to lowest.

Related Topics

Specifying Single Frequency Points

Deleting Frequency Points

Change the Value of an Existing Frequency Point

Specifying Frequency Points to Solve

Choosing Frequencies for Full-Wave SPICE

If you plan to perform a full-wave SPICE analysis, use the **Time Domain Calculation** dialog box to help determine a suitable frequency sweep range for the solution.

To perform the calculation of suitable frequencies to solve:

1. In the **Edit Sweep** dialog box, click **Time Domain Calculation**.

   The **Time Domain Calculation** dialog box appears.

2. Type a minimum rise time value in the **Signal Rise Time** box.

   This value represents the time scale that will characterize the rate of change of the input time signal, which will be applied in the circuit simulator.

3. Type a value in the **Time Steps Per Rise Time** box.

   The time sampling increment for the entire signal is calculated using

   $$\Delta t = \frac{\tau}{N_\tau} \quad (1)$$

   where

   - Δt is the time sampling increment.
   - τ is the signal rise time.
   - N_τ is the number of time steps per signal rise time.

4. Type a value in the **Number of Time Points** box.

   Note that the input time signal duration is determined using $N \times \Delta t$, where N is the number of time points.

5. Click **Calculate**.

   - HFSS now determines the **Maximum Frequency** using

     $$F_{max} = \frac{0.5}{\Delta t} \quad (2)$$

   - where F_{max} is the maximum frequency.

     HFSS determines the **Frequency Step Size** using $\frac{F_{max}}{N}$.

6. Click **OK** to transfer the data to the frequency sweep fields in the **Edit Sweep** dialog box.

Related Topics

Guidelines for Calculating Frequencies for Full-Wave SPICE

Requirements for Full-Wave SPICE

Guidelines for Calculating Frequencies for Full-Wave SPICE

Keep the following guidelines in mind when you set up the calculation for the suggested frequency step size and maximum frequency:

- The maximum frequency should be at least five times the inverse of the rise and fall times. If the specified frequency band is too wide, an HFSS frequency sweep may have convergence problems. If this happens, try to decrease the maximum frequency until the solution converges.
- It is recommended, though not required, that the minimum frequency be less than the maximum frequency divided by the number of frequency steps. It is usually recommended to have at least 500 frequency steps. A higher number will slightly improve the full-wave SPICE solution accuracy, but will also increase CPU and memory requirements to solve the problem. For most cases, using 1000 frequency steps provides a good trade-off between the accuracy and computational requirements.

Warning Occasionally, HFSS can fail to solve for the minimum frequency during a Discrete or Interpolating frequency sweep due to a failure of the port solver to converge. If this happens, try to increase the minimum frequency until the solution process completes successfully. However, the minimum frequency should be as low as possible because the low-frequency response determines the steady-state time response.

- The suggested frequency sweep ranges are estimates. You may have a pulse with a wider frequency content and HFSS's recommended frequency sweep range may miss some of the high frequencies.

Requirements for Full-Wave SPICE

The Full-Wave Spice requirements are as follows:

1. The design problem type in which the solution data panel is opened must be driven terminal.
2. In the Matrix Data panel, for non-imported data, the view type for the solution data must be "Terminal Data" (not "Modal Data").
3. The data must be interpolating, or it must be discrete.
4. If the data is discrete:
   a. It must either be native terminal data, or it must be inferred as terminal. For instance, in a Touchstone file, the comment line "! Terminal data exported" will cause HFSS to interpret the data as terminal, while the comment line "! Modal data exported" will cause HFSS to interpret the data as modal. If HFSS finds neither comment line, it assumes that the data is terminal.
   b. At least 20 frequency points must be provided.

c. HFSS must be able to generate an interpolation basis that converged with <= 0.5% error using no more than 100 basis elements.

Disabling or Enabling a Frequency Sweep

To disable a sweep definition without deleting it:

1. Expand the tree hierarchy under the **Analysis** icon in the project tree.
2. Expand the tree hierarchy under the icon for the analysis setup that includes the sweep.
3. Right-click on the icon for the sweep definition. In the shortcut menu that appears, select **Disable Sweep**.

   You can also disable a sweep by selecting **Properties** from the shortcut menu to open the **Edit Properties** window. In the upper right corner of the **Edit Properties** window, uncheck the Enabled box. Then click OK

To reactivate the sweep, open the shortcut menu again and select **Enable Sweep**.

You can also reactivate the sweep by selecting **Properties** from the shortcut menu for the sweep to open the **Edit Properties** window. Check the **Enabled** box in the upper right corner, and click OK to apply the change and close the window.

Disabling and Enabling an Analysis Setup

To disable an Analysis setup definition without deleting it:

1. Expand the tree hierarchy under the **Analysis** icon in the project tree.
2. Right-click on the icon for the setup definition. In the shortcut menu that appears, click **Disable Setup**.

   You can also disable an analysis setup by selecting **Properties** from the shortcut menu to open the **Edit Properties** window. In the upper right corner of the **Edit Properties** window, uncheck the Enabled box. Then click OK

To reactivate the analysis setup, open the shortcut menu again and select **Enable Setup**.

You can also reactivate the analysis setup by selecting **Properties** from the shortcut menu for the sweep to open the **Edit Properties** window. Check the **Enabled** box in the upper right corner, and click OK to apply the change and close the window.

Specifying the Number of Processors

If you want to use more than one processor, see HPC and Analysis Options,.

Specifying the Desired RAM Limit

The **Desired RAM Limit** setting requests a restriction on the amount of memory the MPS solver used in HFSS may allocate before it must stop solving on-core—solving processes entirely in memory—and start solving off-core. In off-core mode, the solver creates temporary solution files to which it spills, or shifts, data from memory, instead of relying on the operating system to start disk swapping. The location of these temporary files is specified in the General Options (**Tools>Options>General Options,** Project Options tab). The MPS solver is finely tuned at handling its own memory, and can optimize loading only those blocks of memory required for its immediate needs.

Using this option may help to keep the entire solver from being swapped out in the normal course of process management on your computer. This kind of control may be especially important when multiple solvers are running on the same machine. Of course, if the total memory requirement of all processes grows large enough, the operating system will be forced into disk swapping.

Note The Iterative solver doesn't go off-core for efficiency reasons. Because the Desired RAM Limit (the soft memory limit) is used for off-core run, it is not appropriate for the iterative solver. Instead, the iterative solver uses the Maximum RAM Limit (the hard memory limit). Once the limit is exceeded, iterative solver will issue an "Out of memory" error message.

Regardless of this setting, processes are limited to 2GB of address space (3GB with appropriate operating system and boot settings) on 32 bit operating systems and 4TB of address space on 64 bit operating systems - no matter how much physical memory is installed.

Note Regardless of the Desired RAM Limit setting, if allocation fails, the HFSS solver will automatically switch to off-core mode.

In case you receive an error message regarding insufficient memory on a 64-bit operating system, you may have reached a point where the sum of physical RAM plus available swap space exceeds the minimum amount of RAM needed by the off-core solver. Even for the off-core solver, the RAM usage cannot be made arbitrarily small. In that case you can consider increasing the swap space (the virtual memory) in the settings of your system.

To specify the **Desired RAM Limit** of the machine on which HFSS is installed:

1. Click **Tools>Options>Setting HPC and Analysis Options**.
2. Make the Local machine Active.
3. In the **HPC and Analysis Options** dialog click the Edit button.

   This opens the **Analysis Configuration** dialog.
4. Select the Options tab, and select **Desired RAM Limit (GB)**.
5. Type a value for the amount of available memory, in giggabytes, in the text box. This setting affects all HFSS projects for the current configuration for the user and machine.

Specifying the Hard Memory Limit

The **Maximum RAM Limit** setting specifies the absolute limit on the amount of memory that the solver can use. If the solver attempts to allocate more memory than this setting, the solution process will terminate abnormally, and incorrect error messages may appear.

To specify the **Maximum RAM Limit** for the current user and machine:

1. Click **Tools>Options>Setting HPC and Analysis Options**.
2. Make the Local machine Active.
3. In the **HPC and Analysis Options** dialog click the Edit button.

   This opens the **Analysis Configuration** dialog.
4. Select the **Options** tab and **Maximum RAM Limit (GB)**.
5. Type a value for the maximum amount of memory, in megabytes, in the text box.

   The solver will only use the amount of memory specified and no more. This setting affects projects for the Local configuration.

Note ANSYS recommends that you use the **Desired RAM Limit** setting if you wish to limit the RAM that the MPS (default) solver may allocate. However, the Iterative solver doesn't go off-core for efficiency reasons. Because the Desired RAM Limit (the soft memory limit) is used for off-core run it is not appropriate for the iterative solver. Instead, the Maximum RAM Limit (the hard memory limit) is used. Once the limit is exceeded, iterative solver will issue an "Out of memory" error message.

Note Allocation of greater than 2 GB of RAM on 32-bit hardware platforms is only possible with the appropriate operating system and boot settings, even if more than 2 GB are physically installed.

Distributed Memory Solutions with HFSS

When properly configured, and with Domain Decomposition selected as a Setup option, the HFSS solver uses the industry standard Message Passing Interface ("MPI") and can perform solutions that distribute memory use across machines in a cluster or network. Memory used by the MPI-enabled HFSS solver is therefore limited by the set of machines that are available rather than the shared memory available on any single machine. This allows you to simulate larger structures than before and to optimally reconfigure the cluster of machines for the problem at hand.

To use the distributed memory solution in HFSS you will need to install HFSS and MPI software from one of the supported third party vendors on all the machines you intend to use. You may need to set passwords depending on the MPI vendor for authentication on the machines. Settings within HFSS are used to turn on distributed memory solutions and define the list of machines you intend to use. If an HFSS problem involves solver domains or a finite array, then frequency sweeps will not be done using DSO. Also, DSO for Optimetrics will not be allowed.

Detailed instructions about how to get distributed memory HFSS solutions up and running are outlined in the following sections.

- Installation Requirements for Distributed Memory Solutions with HFSS
- Setting up HFSS and Running Distributed Memory Solutions
- Select the MPI Vendor for HFSS
- Running HFSS Distributed Memory Solutions from the Command line
- Discussion of HFSS Distributed Memory Solution
- Interconnects for HFSS Distributed Memory Simulation
- Authentication on Linux (RSH & SSH) for HFSS
- Troubleshooting for HFSS Distributed Memory Solutions

Installation Requirements for Distributed Memory Solutions with HFSS

You must install HFSS on all the machines you intend to use during the HFSS distributed memory solution process. The installation locations for HFSS must be identical and the machines must be uniform: all 64 bit or all 32 bit; all Windows or all Linux. The machines must all use the same interconnect.

In addition, on Windows, you must install one of the supported versions of MPI from either Platform Computing (default) or Intel. Be sure to install the same version of MPI on all machines in your cluster. (Solving on a single Windows machine does not require MPI installation. And users running on Linux do not need to install MPI manually).

Platform	MPI Software
Linux Intel & AMD	Platform MPI 8.1 Intel MPI 4.0.1
Windows Vista, Windows 7 (32 and 64 bit)	Platform MPI 8.1 Intel MPI 4.0.1

You will need to set the password you want to use for your MPI runs on all the machines in the cluster. You can either use the batch command provided by ANSYS to set your password or refer to the MPI vendor's documentation.

Related Topics

Distributed Memory Solutions with HFSS

Setting up HFSS and Running Distributed Memory Solutions

Select the MPI Vendor for HFSS

Running HFSS Distributed Memory Solutions from the Command line

Discussion of HFSS Distributed Memory Solution

Interconnects for HFSS Distributed Memory Simulation

Authentication on Linux (RSH & SSH) for HFSS

Troubleshooting for HFSS Distributed Memory Solutions

Setting up HFSS and Running Distributed Memory Solutions

After setting up your HFSS project normally you will need to set the list of machines used for the distributed memory solution process and configure the distributed memory interface.

The distributed machine list is selected under Tools>Options>HPC and Analysis Options, and created under the Analysis Configuration dialog.

Distributed memory solutions are turned using the **Domain Decomposition** radio button under the Solution Setup Options tab.

If an HFSS problem involves solver domains or a finite array, then frequency sweeps will not be done using DSO. Also, DSO for Optimetrics will not be allowed. Distributed frequency solutions are also known as "DSO" solutions.

Related Topics

Distributed Memory Solutions with HFSS

Installation Requirements for Distributed Memory Solutions with HFSS

Select the MPI Vendor for HFSS

Running HFSS Distributed Memory Solutions from the Command line

Discussion of HFSS Distributed Memory Solution

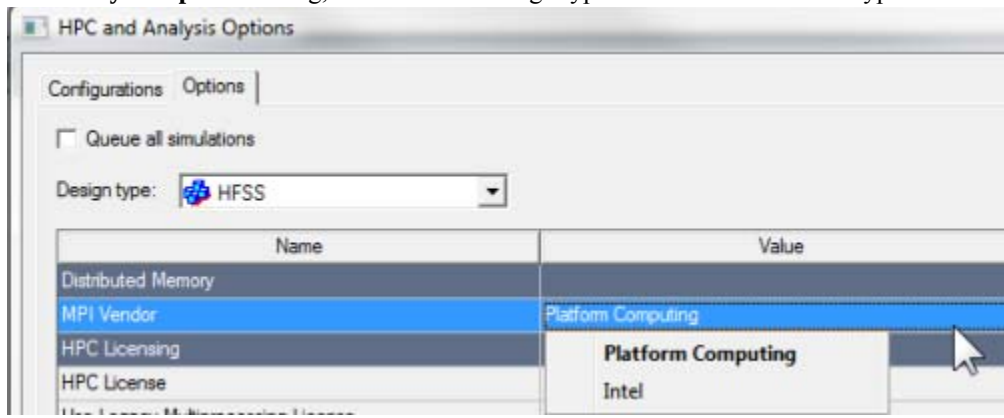
Interconnects for HFSS Distributed Memory Simulation

Authentication on Linux (RSH & SSH) for HFSS

Troubleshooting for HFSS Distributed Memory Solutions

Select the MPI Vendor for HFSS

After installing MPI on your machine from a particular vendor such as Platform Computing or Intel you need to set which type of MPI you are using in HFSS. Go to the **Options** tab of the **HPC and Analysis Options** dialog, and select the design type to set the MPI Vendor type.



Related Topics

Distributed Memory Solutions with HFSS

Installation Requirements for Distributed Memory Solutions with HFSS

Setting up HFSS and Running Distributed Memory Solutions

Running HFSS Distributed Memory Solutions from the Command line

Discussion of HFSS Distributed Memory Solution

Interconnects for HFSS Distributed Memory Simulation

Authentication on Linux (RSH & SSH) for HFSS

Troubleshooting for HFSS Distributed Memory Solutions

Running Distributed Memory Solutions from the Command line for HFSS

You can run distributed memory HFSS solutions from the command line using the -BatchSolve option. Set the distributed memory solve setup option before running the simulation and use the "BatchSolve" flag with the "Distributed" and MachineList" options. For example:

```
hfss -BatchSolve -Distributed -MachineList list="machine1,
machine2" TheProject.hfss
```

This simulates "TheProject.hfss" as a distributed memory solution on machines "machine1" and "machine2."

Note the distributed memory solve setup option can be turned on via scripting if desired.

Related Topics

Distributed Memory Solutions with HFSS

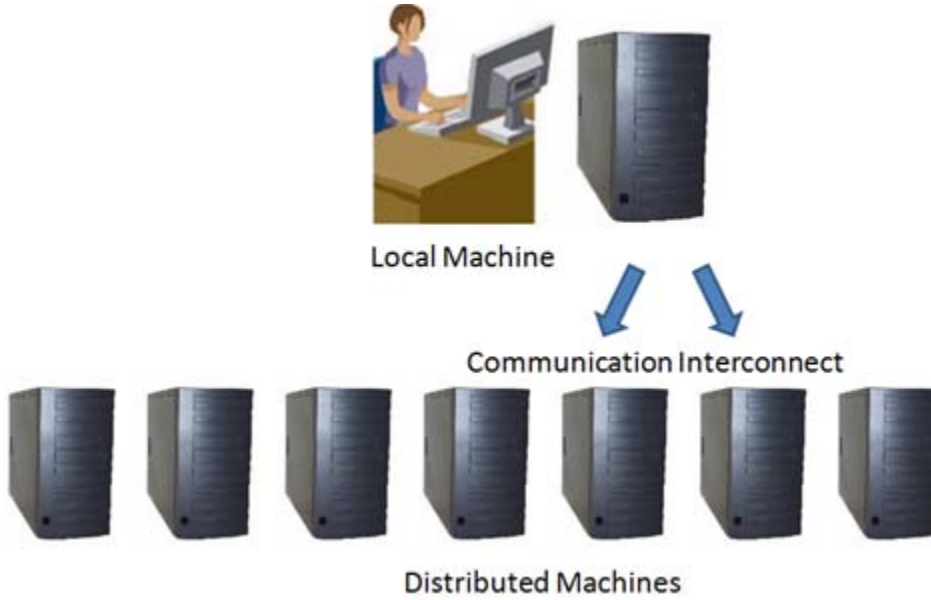
Installation Requirements for Distributed Memory Solutions with HFSS

Discussion of HFSS Distributed Memory Solutions

Each machine or "compute node" is connected to the other nodes via a "communication interconnect" and relies on the message passing library (MPI) to exchange data and synchronize computational tasks. Ethernet, Myrinet and Infiniband are common communication interconnects. Each node is identified by a unique integer ID or rank number. The local machine is known as the "Rank-0 node" and is the master. The Rank-0 machine has many tasks including:

- Management of all communication with the HFSS user interface. None of the distributed machines communicate directly with the user interface but pass all information through the Rank-0 machine.
- Mesh generation. The mesh is generated only on the Rank-0 machine.
- Disk access. None of the distributed machines access their local discs. The simulation mesh, intermediate and solution data are passed to and from the distributed machines using MPI.
- Distribution and control of computational tasks on the distributed machines. The simulation process is dynamic and the Rank-0 machine will determine which of the distributed machines has memory available to distribute tasks accordingly.

- Post-processing of the HFSS Distributed Memory Solution



The algorithms used in the distributed memory version of the HFSS solver engine resemble those used in the non-distributed memory version. The matrix solution algorithms in the distributed memory version of HFSS have been adjusted to use slightly more memory so that larger problems can be simulated in less time.

The MPI enabled HFSS solver engine is not multi threaded. (The regular HFSS engine is multi threaded.) If a particular machine has multiple cores and enough memory you can define this machine several times in the distributed machine list ("doubling up") to take advantage of the extra cores. Certain portions of the MPI enabled HFSS solver will be multi threaded in the future.

During the "Matrix Assembly" and "Matrix Solve" steps of the solution process the HFSS engine attempts to distribute memory use evenly. At various points in the matrix solution process the software will poll the machines in the cluster and determine which machine has the most memory available and then reserve a block of memory on that machine. If a particular machine does not have a large block of memory available the memory use on that machine will grow only slowly. If none of the machines in the cluster have sufficient memory the solution process will terminate and an error message will be posted to the HFSS message window.

Many factors affect solution time. In general, the solution time will decrease as the number of compute nodes increases. However, parallel efficiency decreases as the ratio of communication to computation increases so to some extent you need to match the size of the problem to the size of parallel machine. Simulating small structures on a large cluster will not be efficient and may take longer and use significantly more memory than if the structure was simulated on a single machine.

Network interconnect speed and topology can affect performance significantly in homogeneous clusters. Performance can degrade if machines are "doubled up" to the point of causing memory

bus contention or if the cluster is significantly inhomogeneous and certain faster machines need to wait for slower machines to catch up to synchronization points in the solution process.

Related Topics

Distributed Memory Solutions with HFSS

Installation Requirements for Distributed Memory Solutions with HFSS

Setting up HFSS and Running Distributed Memory Solutions

Select the MPI Vendor for HFSS

Running HFSS Distributed Memory Solutions from the Command line

Interconnects for HFSS Distributed Memory Simulation

Authentication on Linux (RSH & SSH) for HFSS

Troubleshooting for HFSS Distributed Memory Solutions

Interconnects for HFSS Distributed Memory Simulation

To obtain the best possible performance we recommend the use of a network interconnect that supports communication speeds greater than 1000MB/sec or higher. Some high performance interconnects plug into a PCI (Peripheral Component Interconnect), PCI-X (extended), or PCIe (PCI Express) slot on the system.

HFSS-IE 15 supports the following network interconnects:

Platform	Interconnects
Win32	Ethernet/GiGE
Win64	Ethernet/GiGE (default), Myrinet, Infiniband
Linux	Ethernet/GiGE (default), Myrinet, Infiniband

Ethernet/GiGE is the default interconnect on all platforms. You can choose one of the alternate interconnects by setting the ANSOFT_MPI_INTERCONNECT environment variable to "myri" for Myrinet and "ib" for Infiniband.

Interconnect variants are supported on Linux. Set the ANSOFT_MPI_INTERCONNECT_VARIANT to the desired interconnect variant. For example, set "ANSOFT_MPI_INTERCONNECT_VARIANT=silverstorm" to use the silverstorm variant.

Related Topics

Distributed Memory Solutions with HFSS

Installation Requirements for Distributed Memory Solutions with HFSS

Setting up HFSS and Running Distributed Memory Solutions

Select the MPI Vendor for HFSS

Running HFSS Distributed Memory Solutions from the Command line

Discussion of HFSS Distributed Memory Solution

Authentication on Linux (RSH & SSH) for HFSS

Troubleshooting for HFSS Distributed Memory Solutions

Authentication on Linux (RSH & SSH) for HFSS

An important step in using a high performance cluster is setting up authentication across machines in such a way that the machines can be accessed without a password. By default HFSS uses SSH authentication on Linux to spawn commands on the remote machines but also supports RSH. The selection of which to use is made on the **Options** tab of the Tools/Options/HPC and Analysis Options dialog.

Related Topics

Distributed Memory Solutions with HFSS

Installation Requirements for Distributed Memory Solutions with HFSS

Setting up HFSS and Running Distributed Memory Solutions

Select the MPI Vendor for HFSS

Running HFSS Distributed Memory Solutions from the Command line

Discussion of HFSS Distributed Memory Solution

Interconnects for HFSS Distributed Memory Simulation

Troubleshooting for HFSS Distributed Memory Solutions

Troubleshooting for HFSS Distributed Memory Simulations

A number of things can prevent distributed memory solutions from completing successfully. This section provides suggestions to debug problems.

It is often a good idea to set up and run a small simulation with two processes on a single machine before moving to a large cluster. Using a single machine will allow you to verify that HFSS and MPI are installed correctly while eliminating problems arising from remote installation, authentication and firewall settings.

Many problems occur because the MPI software cannot start and run due to authentication and firewall issues. Please check with your MPI vendor and their end user documentation for information about how to verify that authentication and firewall settings are correct.

If you are using MPI on Windows from Platform Computing you can test whether MPI will run by using the "mpidiag" utility:

- From a command prompt browse to the Platform Computing binaries located at

  `<HFSS_Installation_Directory>\common\fluent_mpi\multiport\mpi\_win64(win32)\pcmpi\bin`
- Enter "mpidiag -s <name_of_machine> -at" to run an authentication test.
- Run the authentication tests in both directions, i.e. both to and from all target machines. This will verify that MPI passwords and firewall settings are correct.

  The tests need to be run in both directions because firewalls may allow communication in one direction but not the other.

Platform Computing's implementation of MPI on Windows requires that you enter a password on each machine in the cluster to run MPI solutions.

- From a command prompt browse to the Platform Computing binaries located at

```
<HFSS_Installation_Directory>\common\fluent_mpi\multiport\mpi\_
win64(win32)\pcmpi\bin
```

- To set the password run "mpidiag -s <name_of_machine> -cache -at" and enter the password at the prompt.

Recall that HFSS must be installed in the same directory on all machines in the cluster and that the cluster must be uniform (i.e. all Linux machines, all 32 bit Windows machines, or all 64 bit Windows machines)

\Verify that the version of the third party MPI software is identical on all the machines and that it is listed in the table of supported versions above.

Verify that the machine names are correct and that all the machines can be reached on the network.

Related Topics

Distributed Memory Solutions with HFSS

Installation Requirements for Distributed Memory Solutions with HFSS

Setting up HFSS and Running Distributed Memory Solutions

Select the MPI Vendor for HFSS

Running HFSS Distributed Memory Solutions from the Command line

Discussion of HFSS Distributed Memory Solution

Interconnects for HFSS Distributed Memory Simulation

Authentication on Linux (RSH & SSH) for HFSS

15

Running Simulations

After you specify how HFSS is to compute the solution, you need to begin the solution process. In general, the **Analyze** command applies to the selected setup and associated sweeps, if any, or to a select sweep. To use this command, right-click on a setup or sweep in the Project tree, and click the command on the context menu. The **Analyze All** command applies to all enabled setups, dependent setups, and sweeps at or below the level invoked in the Project tree. To use this command, either click **HFSS>Analyze All** or right-click on the Analysis icon in the Project tree and select **Analyze All**.

What do you want to do?

- Solve a single setup with or without sweeps
- Solve a specific sweep
- Enable a queue so that multiple simulations can run sequentially as resources become available.
- Run more than one simulation, whether multiple setups, or multiple sweeps under a single setup, or setups with dependencies.
- Monitor queued simulations
- Configure and run remote analysis
- Configure and run distributed analysis
- RSM Integration with Job Management UI
- Large Scale DSO for Parametric Analysis
- High Performance Computing (HPC) Integration
- Monitor the solution process
- Change a solution priority for system resources
- Abort an analysis
- Re-solve after modifying a design

- Re-solve after ANSYS Workbench Feedback

Related Topics

Running an Optimetrics Analysis

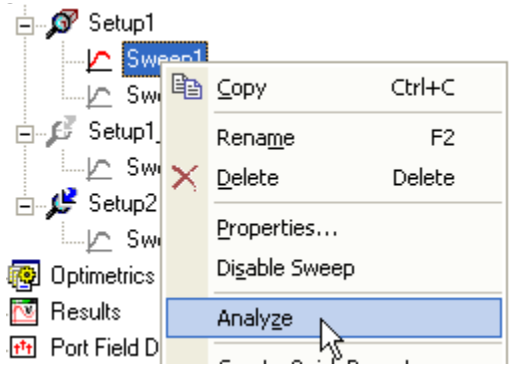
ANSYS Workbench Integration Overview

Running from a Command Line

Solving a Single Setup or Sweep

To run a single setup or sweep:

1. Select a solution setup or sweep in the project tree.
2. Right-click and select **Analyze** from the shortcut menu. The graphic shows a single sweep selected for analysis.



HFSS computes the 3D field solution inside the structure for a solution, and for a select sweep, does so for the sweep variables.

To run more than one analysis at a time, follow the same procedure while a simulation is running. If you have enabled queuing, the next solution setup will be solved when the previous solution is complete.

Note If a linked dependency in the setup is already simulating (for example, due to setup links to the same external source for a near or far field wave, or a magnetic bias), HFSS will not allow another dependent simulation to start until the first use of the source has completed.

Related Topics

Running an Optimetrics Analysis

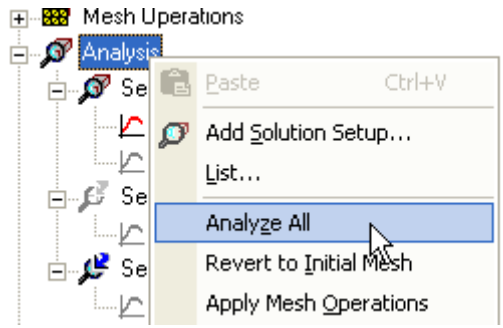
Running More Than One Simulation

To solve every enabled solution setup in a design:

1. In the project tree, under the design you want to solve, select **Analysis**.
2. Click **HFSS>Analyze All**.

Each enabled solution setup is solved in the order it appears in the project tree.

The example here show an analysis invoked from the Project tree popup menu with three setups, one disabled, two enabled. The first setup has one sweep enabled, and one disabled (grayed icon).The second setup is disabled, and the third is enabled, with a disabled sweep.

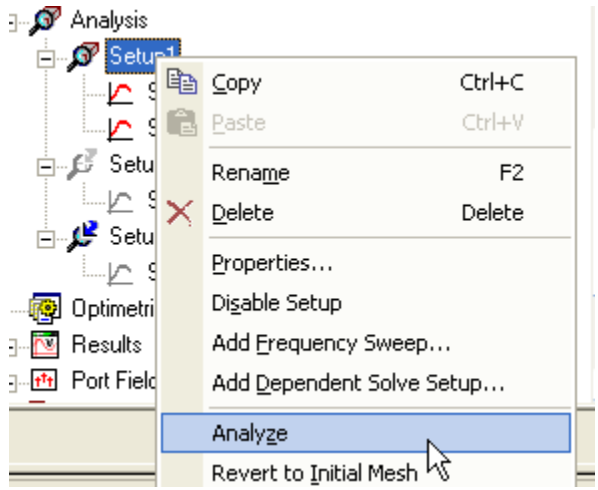


Note The **General** tab for the Setup includes an "Enabled" checkbox. By default, this is checked. Unchecking the **Enabled** checkbox excludes a setup from running

To solve two or more sweeps or two or more parametric analyses under a setup:

1. In the project tree, under the design you want to solve, right-click the setup icon that includes the sweeps of interest.
2. Click **Analyze** on the shortcut menu.

   Each solution sweep under that setup is solved in the order it appears in the project tree, using the available machines. The example below shows a setup with two enabled sweeps.



Related Topics

Technical Notes: The Solution Process

Technical Notes: Handling Complicated Models

Solving a Single Setup

HPC and Analysis Options.Remote Analysis

Monitoring the Solution Process

Aborting Analysis

Running an Optimetrics Analysis

Monitoring Queued Simulations

If you have multiple setups for a design, and have selected **Analyze All**, the simulations can be queued until there is a machine available. You enable queuing in the HPC and Analysis Options: Options tab. If queuing is enabled and you run multiple setups, they are solved in the order that they appear in the project tree. You can prioritize setups by changing the order in the queue.

1. To view the solution queue, click **Tools>Show Queued Simulations** or click the **Show Queue** icon on the toolbar.

   This displays a dialog that displays each simulation and its current status. You select and remove any simulation from the queue.

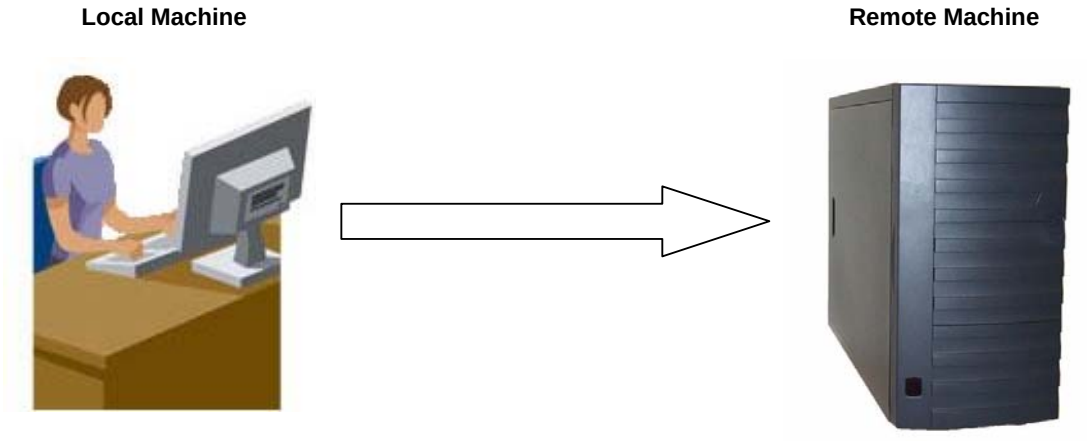
   You can also select any setup and use the **Move up** and **Move down** buttons to prioritize them.

2. To remove a simulation from the queue, select the simulation, and click **Remove from Queue**.

   This removes the selected simulation from the queue.

Remote Analysis

It is possible to solve a project on a different machine from the one on which you are running HFSS. This is particularly useful when you want to take advantage of a more powerful machine but it is not convenient to access that machine. This process involves configuring the machine that is to perform the solving (the remote machine), as well as the machine from which the simulation is to be launched (the local machine). This can also be extended into distributed analysis, where a specified analysis, if supported, is concurrently solved on multiple machines.



Note Communication between machines in remote analysis and distributed analysis can drastically affect performance. Use of a high-speed network system, like Gigabit or Infiniband, is recommended for optimal performance.

- Prerequisites for Remote and Distributed Analysis
- Configuring the Local Machine to Solve Remotely
- Remote Analysis Options
- Running Remote Analysis

The Tools>Options>Export Options Files command writes xml files containing the Options settings at all levels to the specified directory. The **Tools>Options>Export Options** feature is intended to make it easier for different users to use ANSYS EM tools installed on shared directories or network drives. The Example Uses for Export Options Features section outlines some use cases enabled by this feature.

Prerequisites for Remote and Distributed Analysis

1. You must have ANSYS EM's Remote Simulation Manager (RSM) or a supported High Performance Computing (HPC) management software program. (See High Performance Computing (HPC) Integration). The list of currently-supported HPC software includes
   - Platform's Load Sharing Facility or LSF

- Altair's PBS
- Sun GridEngine
- Microsoft® Windows® Compute Cluster Server 2003
- Microsoft® Windows® HPC Server 2008 R2 and HPC Server 2012

2. HFSS must be accessible from all remote machines as well as accessible on the local machine.
3. If you use RSM, it must be accessible from all remote machines. In addition, the HFSS engines must be registered with each initialization of RSM. To do this, on each remote machine:
   - On Windows on the local and remote machines, click **Start>Programs>ANSYS Electromagnetics>***product* **>Register with RSM**. You can also run **RegisterEnginesWith RSM.exe**, located in the product subdirectory (for example, C:\Program Files\AnsysEM\hfss15\Windows 64-bit\RegisterEnginesWithRSM.exe).

     In each case, you see a dialog confirming the registration. OK the dialog.
   - On Linux, run **RegisterEnginesWithRSM.pl**, located in the product installation directory. (for example, /apps/ansyselectromagnetics/hfss15/RegisterEnginesWithRSM.pl).

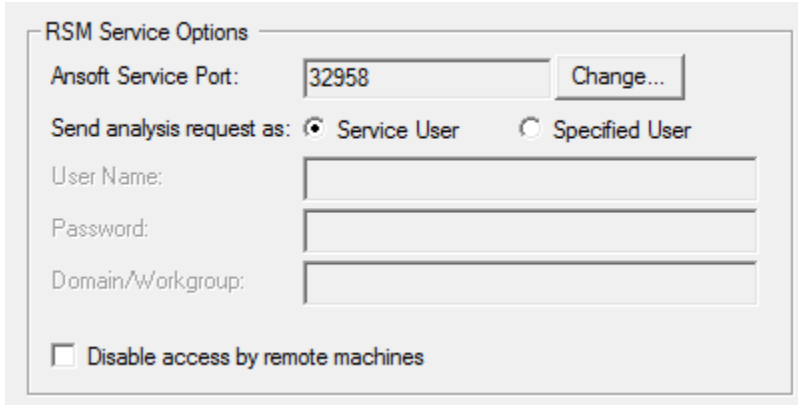
If the RSM service cannot run due to permission issues for the configuration file, it issues an error message and exits. If your product is not registered with RSM, the analysis will run locally.

Configuring the Local Machine to Solve Remotely

To set the Analysis options in HFSS see Configuring Distributed Analysis.

Remote Analysis Options

You also set the Remote Analysis Options in the **General options** dialog, **RSM Analysis Options** tab.



Select whether to run simulation processes as the user running RSM (Service User), or a Specified User. If you select Specified user, you must provide the User Name, Password, and any Domain/Workgroup on which this user is defined. If the name or password is incorrect, the Message window issues a warning message, and the solver attempts to perform the analysis as the Service User.

Running Remote Analysis

When you run a simulation remotely, you should see a message in the Progress window identifying the design name, and the specified remote machine. You will see Progress messages as the simulations continues. When the simulation is complete, you will see a message in the **Message** window.

Related Topics:

Distributed Analysis

Troubleshooting

Running from a Windows Remote Terminal

Troubleshooting

Problem: When you try to solve from local to remote machine, an HFSSCOMENGINE process starts on the remote machine, but the HFSS user interface hangs indefinitely.

This occurs if the remote solve option is enabled after the COM daemon is started, or when the option "Don't allow exceptions" is selected for the Windows firewall.

Resolution: Remote solve needs either firewall exceptions to be ON or firewall to be completely turned off.

Problem: When you try to solve from a local to a remote machine, you receive the following error message:

[error] Unable to locate or start COM engine on 'nomachine' : Unable to reach AnsoftRSMService. Check if the service is running and if the firewall allows communication. (10:57:13 PM Aug 13, 2009)

Resolution: This message can happen if the machine is not present, the network connection is down, if there are firewall issues or if the service is not running.

Remote Solve Node = Windows

Error: "Unable to locate or start COM engine on <remote node> : Unable to reach AnsoftRSMService. Check if the service is running and if the firewall allows communication."

1. Try disabling the firewall. If this fixes the problem, please contact at hfqa@ansoft.com
2. Confirm that you have not changed the Ansoft Service Port in **Tools > Options > General Options > Remote Analysis Options** from the default 32958. If you have, change it back to 32958, restart HFSS, and try to solve again.
3. Make sure that the local machine is able to contact the RSM port on the remote node. Open a command prompt on the local machine and type telnet <remote node name> 32958. If the terminal appears to be hanging then the connection was successful.
4. Check to make sure the Ansoft Communication Service is running. To do this, go to the **Windows Control Panel** and choose **Administrative Tools > Services**. Find the Ansoft RSM Service and make sure its status says Started. If it is not running, try to start it by right-clicking on the service and choosing Start. If it still does not start, then check the username/password com-

bination listed in the Log On tab of the service properties.

5. Make sure the user listed in the service is an administrator.
6. Make sure the COM engine is registered with the Ansoft RSM Service. From the Windows menu, choose **Start>All Programs >ANSYS Electromagnetics>ANSYS Electromagnetics Suite 15.0 >Windows 64-bit>Windows 64-bit>Register with RSM** to register the engines.

Error: "Unable to locate or start COM engine on <remote node>: Engine is not registered with the Ansoft RSM service which is running on this machine."

a. To register the engine, from the Windows menu, select **Start >All Programs >ANSYS Electromagnetics>ANSYS Electromagnetics Suite 15.0 >Windows 64-bit> Register with RSM**

Remote Solve Node = Linux

Error: "Unable to locate or start COM engine on <remote node>: Unable to reach AnsoftRSMService. Check if the service is running and if the firewall allows communication."

1. Try disabling the firewall. If this fixes the problem, please contact at hfqa@ansoft.com
2. Confirm that you have not changed the Ansoft Service Port in **Tools >Options >General Options >Remote Analysis Options** from the default 32958. If you have, change it back to 32958, restart the ANSYS Electromagnetics product, and try to solve again.
3. Make sure that the local machine is able to contact the RSM port on the remote node. Open a command prompt on the local machine and type telnet <remote node name> 32958. If the terminal appears to be hanging then the connection was successful.
4. Check to make sure Remote Simulation Manager is running. To do this:
   a. Go to the 'rsm' subdirectory of the Remote Simulation Manager installation directory, <RSM installdir>/rsm.
   b. Type ./ansoftrsmservice status.
   c. If the status query indicates that the service is stopped, type ./ansoftrsmservice start.
5. Make sure the COM engine is registered with RSM. Type ./RegisterEnginesWithRSM.pl status from within the HFSS installation directory. If the status query indicates "Not registered", type:

   ./RegisterEnginesWithRSM.pl add.

Error: "Unable to locate or start COM engine on <remote node>: Engine is not registered with the Ansoft RSM service which is running on this machine."

1. To register the engine, go to the ANSYS Electromagnetics product installation directory and type:

   ./RegisterEnginesWithRSM.pl add.

Distributed Analysis

Distributed analysis allows users to split certain types of analyses and solve each portion of an analysis simultaneously on multiple machines. Simulation times can be greatly decreased by using this feature.

HFSS and HFSS-IE support different forms of distributed analysis:

- Distributing rows of a parametric table, either as a regular DSO, or as Large Scale DSO performed through command line. Large Scale DSO generates a reduced set of outputs.
- Distributing array solves.
- Distributing domain solves.
- Distributing a single or discrete interpolating sweep.

Note Communication between machines in remote analysis and distributed analysis can drastically affect performance. Use of a high-speed network system, like Gigabit or Infiniband, is recommended for optimal performance.

Related Topics

Configuring Distributed Analysis

Editing Distributed Machine Configurations

Selecting an Optimal Configuration for Distributed Analysis

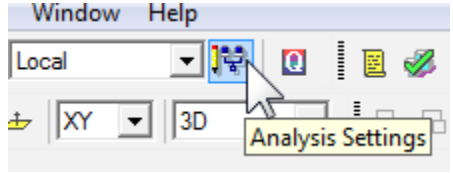
Large Scale DSO for Parametric Analysis

Configuring Distributed Analysis

To configure distributed analysis you select a distributed machine configuration. This is a list of machines to use for a simulation, based on considerations such as whether the simulation is more memory intensive or more CPU intensive, relative to the resources available on your network. (See Selecting an Optimal Configuration for Distributed Analysis for a discussion of issues.) To create an new distributed machine configuration. or to edit an existing one, see Editing Distributed Machine Configurations.

To select from an existing configuration:

1. Click the Analysis settings icon on the Toolbar or click **Tools>HPC and Analysis Options.**

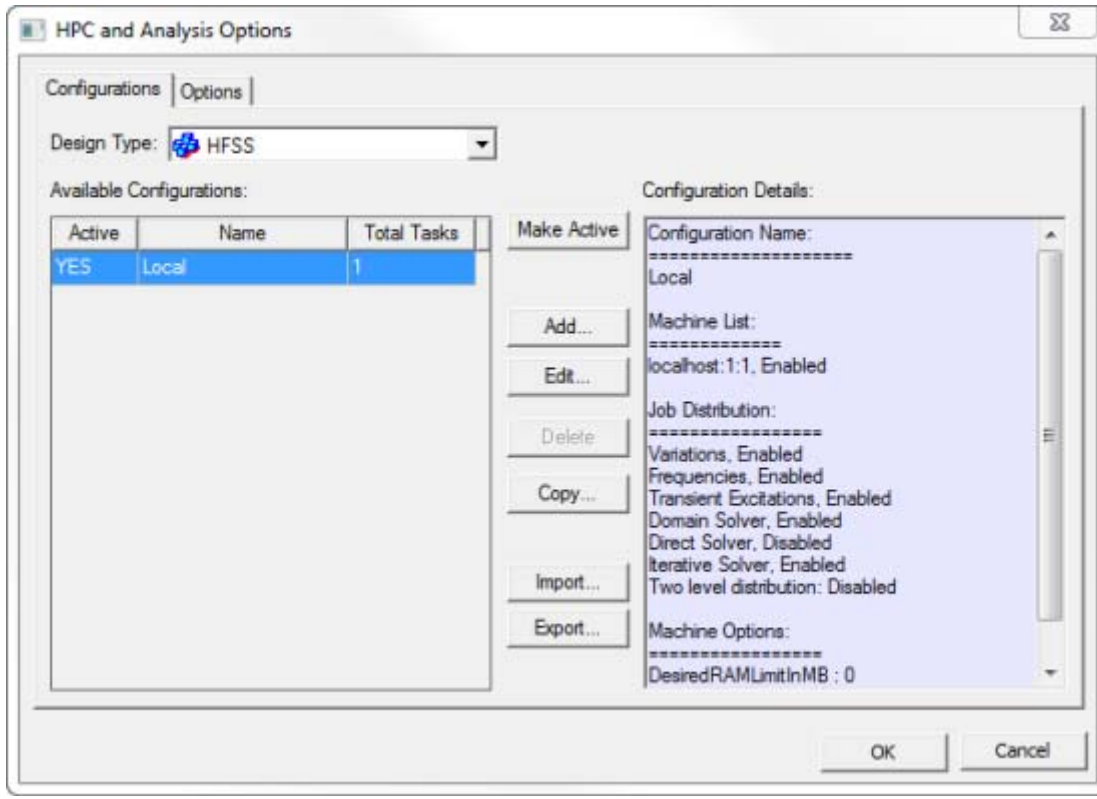


This opens the **HPC and Analysis Options** dialog. You can define from configurations defined for HFSS and HFSS-IE design types separately. You can view a list of Available configurations, and a report of the configuration details. From the current, list, you can select a configuration to **Make Active**. You can also **Add** a new configuration, **Edit** an existing one, or

Export as a ANSYS Configuration file (*.acf). You can also **Import** a configuration file. This lists existing configurations, and shows all machines in the selected configuration, enabled or not. You can **Copy** an existing configuration, typically to edit the name and contents for other purposes.

The **Analysis Options** tab lets you choose to Queue all simulations, You define configurations defined for HFSS and HFSS-IE design types separately. In the **Options** tab you can enable queuing, set the design type, specify the Distributed Memory vendor, set licensing options and enable GPU for Transient Solves. You can also specify default process priority .

For a more detailed discussion of this dialog, see Setting HPC and Analysis Options.



2. To define a new configuration, on the **Analysis Settings** dialog, click the **Add** button to open the **Analysis Configuration** dialog. See Editing Distributed Machine Configurations.

Related Topics

Editing Distributed Machine Configurations

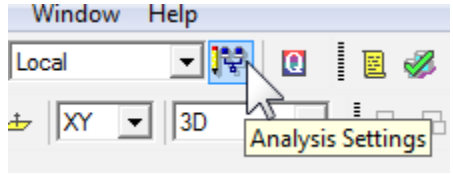
Selecting an Optimal Configuration for Distributed Analysis

The Tools>Export Options Files command writes xml files containing the Options settings at all levels to the specified directory (default, ~\Documents\Ansoft\). The **Tools>Options>Export Options** feature is intended to make it easier for different users to use ANSYS EM tools installed on shared directories or network drives. The Example Uses for Export Options Features section outlines some use cases enabled by this feature.

Editing Distributed Machine Configurations

To create a new distributed machine configuration, or edit an existing machine configuration.

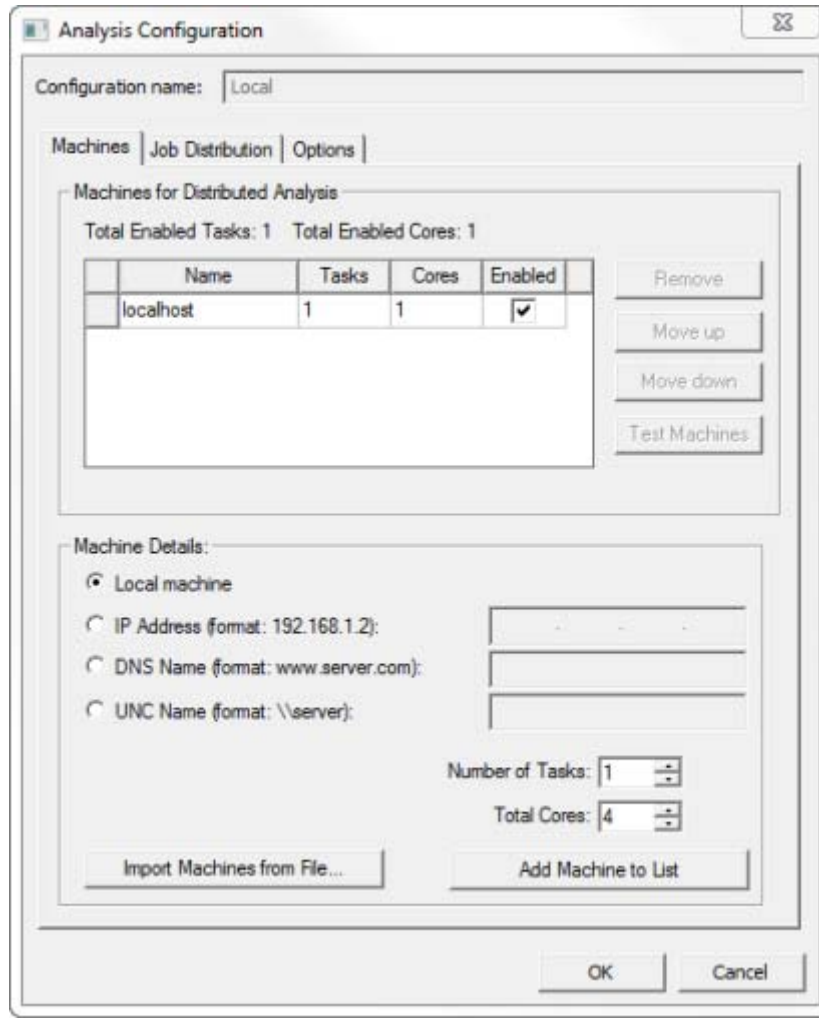
1. Click **Tools>Edit Active Analysis Configuration** to open the **Analysis Configuration** dialog directly or Click the Analysis settings icon on the Toolbar.



This opens the **Analysis Configuration** dialog. directly. You can also access this dialog from the **HPC and Analysis Options** dialog by clicking the **Add..,**, **Edit...**, or **Copy...** buttons.

If you have selected **Add...** from the **HPC and Analysis Options dialog** , the fields are empty. If you have selected **Edit...** or used **Tools>Edit Active Analysis Configuration** or clicked

Copy from the **HPC and Analysis Option**s dialog, the fields show the selected configuration.



2. Specify the name of the new or edited configuration. It cannot be empty and cannot be a previously used name or a reserved word.

Machines Tab

3. This tab contains the machine list for the analysis configuration.

   Here you can provide machine information, either by specifying Remote machine details, or by importing a list of machines from a file. You can then remove, order, test, and enable machines on the list. Control buttons let you **Add machine(s) to list** or **Remove** machines from the list.

Numbers of Tasks and Total Cores

Note that each machine has an associated number of tasks and number of cores. The number of

tasks specifies the total number of compute jobs that will be run on that machine simultaneously. Each separate solver or engine instance is one task. The Total Cores specifies the total number of cores that will be used on the given machine. This is how you specify multiprocessing. For instance, if you want to run two threads for each task, you specify Total Cores = 2 x Total Tasks. The Total Cores must always be greater than or equal to the Number of Tasks. If the number of cores is not an exact multiple of the number of tasks, some tasks will use more cores than others. For instance, if Number of Tasks is 4 and Total Cores is 10, 2 tasks will use 3 cores, and 2 tasks will use 2 cores.

The purpose behind specifying Total Cores at the machine level is to allow you the flexibility of assigning large amounts of multiprocessing to some machines and smaller amounts to others.

Import Machines from File...

You can import a machine list from a file, and an enhanced the file format handles the new flexibility. Each line of the file can contain a machine specifier of the form:

```
<MachineName>:<NumTasks>:<NumCores>.
```

Note: this same format is used with the "-machinelist file=" command line option.

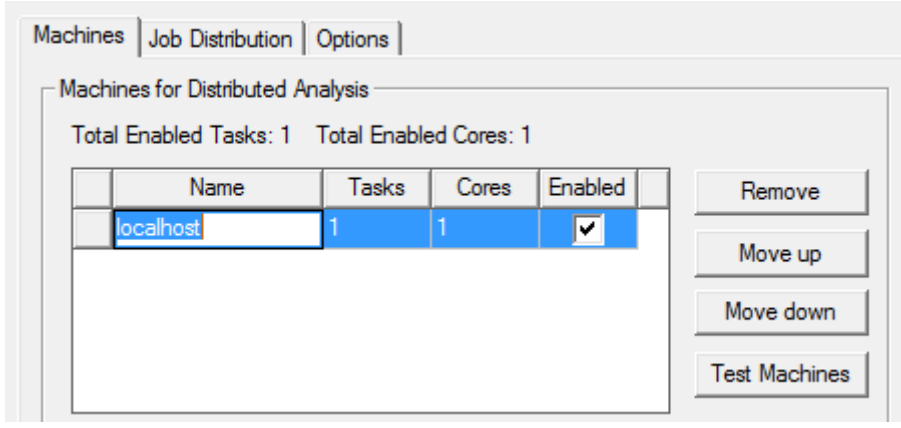
Local Machine Radio Button

To streamline the common case of running jobs on the current machine, use the dedicated radio button to specify the local machine.

4. For each machine to manually add to the list, under Remote Machine Details, specify an IP address, a DNS name, or a UNC name.

   The remote machines must have the same ANSYS Electromagnetics Suite version installed in the same OS and version, and have the RSM service active.

   Once you have specified the remote machine details, either directly or by **Importing Machine from a File**, you select a machine from the table to enable the buttons to **Remove** a machine, or to **Move** a machine up or down on the list.



   The displayed list always shows the order in which you entered them irrespective of the load on the machines. To control the list order, select one or more machines, and use the **Move up**

or **Move down** buttons. **Move up** and **Move down** are enabled when you select one or more adjacent machine names.

Enabled Machines

Each machine on the current list has an **Enabled** checkbox. Here you can enable or disable the listed machines according to circumstance. Above the table, the dialog gives a count of the total enabled tasks, and the total enabled cores.

For distributed tasks, the software will allocate the total cores on a given machine to that machine's tasks. If a machine with 8 cores is running 2 distributed tasks, the software will automatically allocate 4 cores to each task. If it is running 4 distributed tasks, each gets 2 cores. And if it is running 3 distributed tasks, the first two tasks get 3 cores and the last task gets 2 cores.

For a given variation (for example, frequency or geometry), you should make assignments so that each task has the same number of cores. This is because the solvers attempt to make each task computationally balanced. For example, with two machines, one with eight cores, and another with four, assuming that the memory is proportionally equivalent, you could assign two tasks for machine 1, and one task for machine 2, giving all tasks the same number of cores.

In general, HFSS and HFSS-IE use machines in the distributed analysis machines list in the order in which they appear. If you select a distributed configuration (rather than Local) from the Toolbar menu and you launch multiple analyses from the same UI, HFSS and HFSS-IE select the machines that are running the fewest number of engines in the order in which the machines appear in the list. For example, if the list contains 4 machines, and you launch a simulation that requires one machine, HFSS chooses the first machine in the list. If another simulation is launched while the previous one is running, and this simulation requires two machines, HFSS chooses machines 2 and 3 from the list. If the first simulation then terminates and we launch another simulation requiring three machines, HFSS chooses 1, 4, and 2 (in that order).

Test Machines

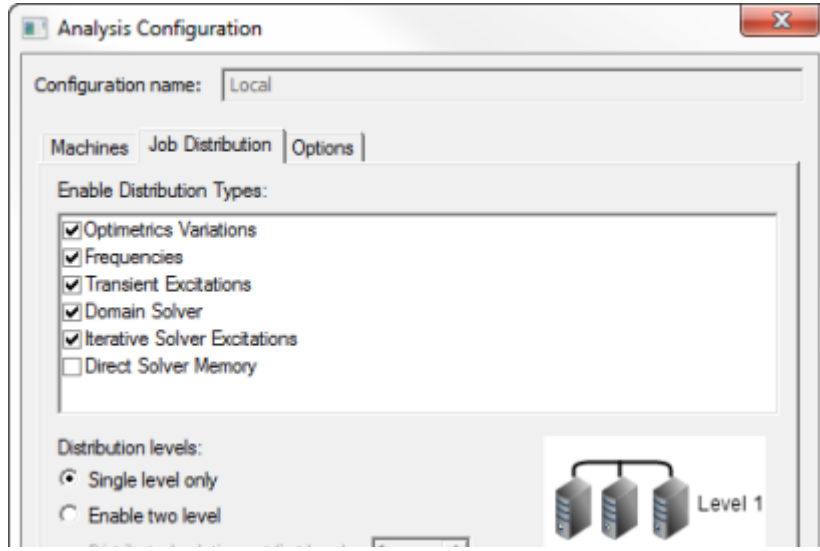
Test Machines- When multiple users on a network are using distributed solve or remote solve, they should check the status of their machines before launching a simulation to ensure no other ANSYS

EM processes are running on the machine. To do this, you can select one or more machines and click the **Test Machine**s button. A **Test Machines** dialog opens.

The test goes through the current machine list and gives a report on the status of each machine. A progress bar shows how far testing has gone. An Abort button lets you cancel a test. When the test is complete, you can OK to close the dialog. If you need to disable or Enable machines from the list based on the report, you can do so in the **Distributed Analysis Machines** dialog.

Job Distribution Tab

Use this tab to enable specific types of job distribution and to enable multi-level solves. The Job distribution types listed are design type specific and will differ between products.

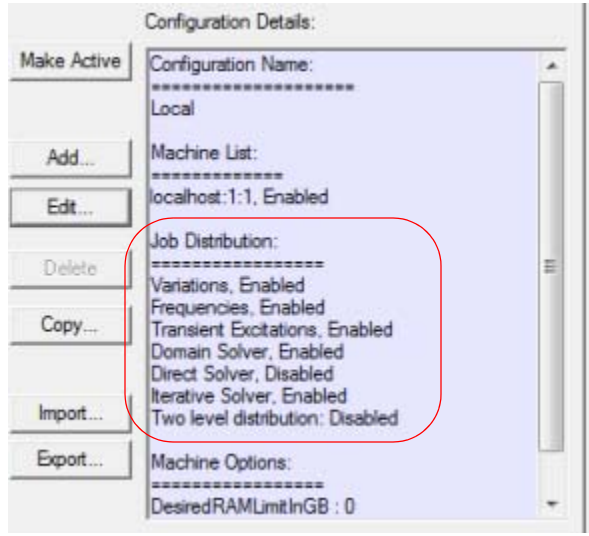


Use the check boxes enable/disable available distribution-types. The job distribution list box allows you to specify which job distribution types to allow for the current analysis configuration. At solve

time, the ANSYS Electromagnetics software automatically select the best distribution type from the enabled distribution types. By enabling/disabling distribution types, you can control the job distribution.

Just because you enable a distribution type does not mean it will be used. It must be also allowed by the solve setup. Note that the enabled distribution types will apply to all setups of the given design type, so it is possible for different setups in a design to be solved using different distribution types.

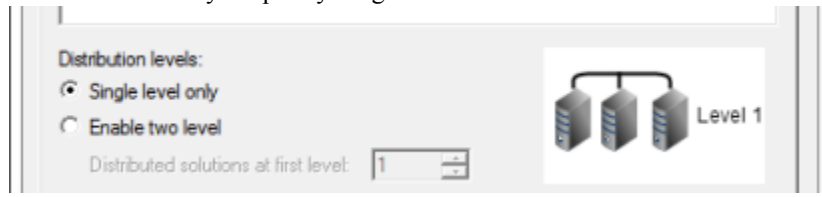
The distribution types that you enabled here are listed on the **Analysis Settings** dialog in the Configuration details pane when select that configuration from the list.



For products that support two-level distribution, when the design is appropriate, you can turn on two level distributed solves, and specify the number of engines to use for level 1. An example design that could use two level distribution would be an array with frequency sweeps.

Distribution Levels

The radio buttons let you specify Single level or Enable two level distribution.



If you select single level, one distribution type will be applied at each stage of the solution process. If multiple types are available, the higher level solution will generally be distributed. All machine tasks will be used by the single-level distribution.

Single Level Distributions

Supported distribution types are Optimetrics Variations, Frequencies, Transient Excitations, Domain Solver, Iterative Solver Excitations and Direct Solver Memory. Solver distributions require MPI.

One distribution type will be applied at each stage of the solution process. Common stages include "LastAdaptive," "Sweep," and "Parametric." All machine tasks will be used by the single-level distribution.

Parallel distribution types such as Optimetrics Variations, Frequencies, and Excitations are considered as not required - if these types are not able to distribute, the simulation can be run sequentially. Memory distribution types such as Direct Solver Memory and Domain Solver are considered as required - if these types are enabled the software will assume that distribution is necessary to extend the simulation scale or add fundamental solution capabilities.

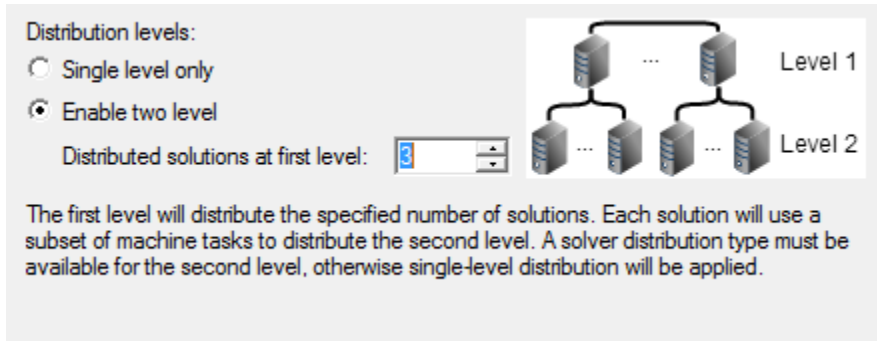
When multiple distribution types are available, the higher level solution will generally be distributed. For example, Optimetrics Variations will be distributed when both Optimetrics Variations and Iterative Solver Excitations are enabled. Domain solver and Direct Solver Memory are exceptions because they are required. Even though they are lower level, these types are distributed instead of parallel distribution types.

The following is an example of single-level distributions:

An analysis setup distributes Iterative Solver Excitations during adaptive, then distributes Frequencies during the sweep.

Two-level Distributions

Selecting Enable two-level enables the Distributed Solutions at first level selection box.



The first level will distribute the specified number of solutions. Each solution will use a subset of machine tasks to distribute the second level. A solver distribution type must be available for the second level; otherwise single-level distribution will be applied.

Domain solver requires a minimum of 3 to 4 tasks. When there are not enough tasks, the simulation will fall back to a single level distribution. For example, the total number of machine tasks is 8 an "Distributed solutions at first level" is 3. Since the 3rd solution at first level is only allowed 2 tasks, the simulation will fall back to a single level distribution of Domain Solver.

Direct Solver Memory requires a minimum of 2 machine tasks.

Types that can be distributed at Level 1 are Optimetrics Variations and Frequencies. All Solver distributions can be distributed as Level 2.

The following are examples of two-level distributions:

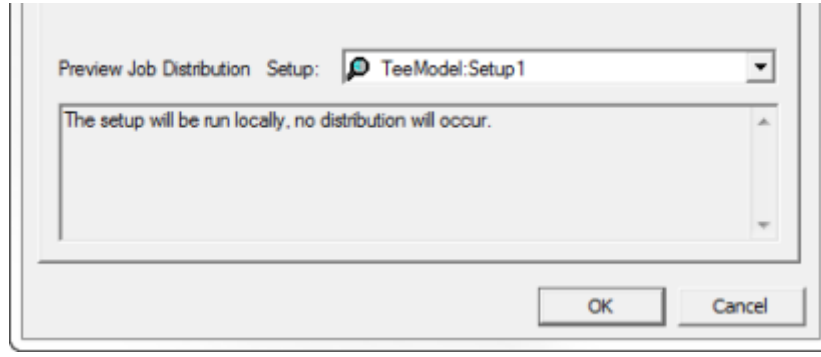
- A parametric setup distributing Optimetrics Variations as level 1 and Iterative Solver Excitations as level 2.
- A frequency sweep distributing Frequencies as level 1 and Direct Solver Memory as level 2

Machine Tasks Distribution

All machine tasks are used in a single-level distribution. For example, 10 frequencies will be solved in parallel when there are 10 machine tasks. In two-level distributions, machine tasks are generally distributed evenly between the first-level solutions.

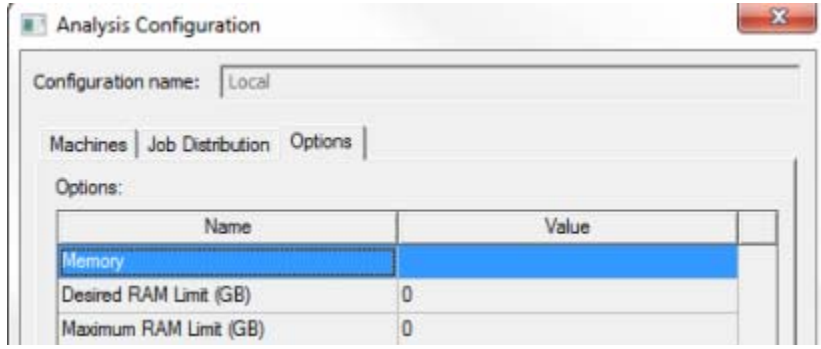
Preview Job Distribution

The Preview Job Distribution Setup menu and Field lets you set how the selected setup will be distributed.



Options Tab (Analysis Configuration Dialog)

Use the **Options** tab to specify options for the current analysis configuration. Different design types may have different options available in their analysis configurations.

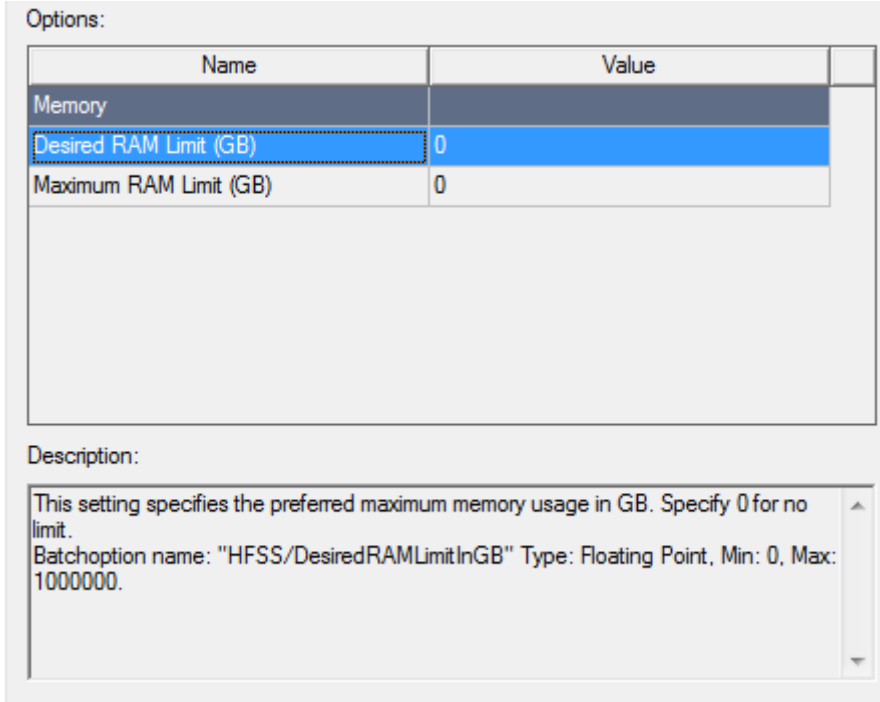


The Options grid contains the available option Names and editable values.

These options settings will be in effect only when all the following are true:

- A design is being solved whose design type matches this analysis configuration's
- This analysis configuration is the active configuration for its design type
- You have not specified corresponding batch options on the command line. Command line batchoptions can be used to override the options specified by the active configuration.

Selecting an available options causes a display of a description of the option.



Relation to Batchoptions

Analysis configuration option settings can be overwritten by specifying the option name and value inside a -batchoption string. See -batchoptionhelp for a list of batchoption names and possible values. You can also view all available and frequently used batchoptions in the Job Management **Submit Job To:** dialogue, by clicking the **Add...** button under Analysis Options. This opens the **Add Batchoption** dialog, which gives access to all batchoptions.

Adding Configurations or Accepting Edits

Click **OK** to accept the changes and close the **Analysis Configuration** dialog. Only machines checked as **Enabled** appear on the distributed machines Configuration Details Machine list.

Regardless of the machine(s) on which the analysis is actually run, the number of processors and Desired RAM Limit settings, and the default process priority settings are now read from the machine from which you launch the analysis. See Setting HPC and Analysis Options.

For more information, see distributed analysis.

Note The option is only active if there are multiple rows listed in the parametric table, there are multiple frequency sweeps listed under a given analysis setup, and the number of distributed analysis machines is two or greater.

Related Topics

Configuring Distributed Analysis

Editing Distributed Machine Configurations

Selecting an Optimal Configuration for Distributed Analysis

Selecting an Optimal Configuration for Distributed Analysis

With the distributed solve option (DSO) it is possible to assign more than one machine in the DSO setup panel to a single computer - assuming the computer has multiple cores. For example, a quad core computer can be listed 4 times in the DSO machine list and each CPU will run a separate simulation in the sweep. This raises the question of how to configure a distributed solution option (DSO) in combination with the multiprocessing option to take optimal advantage of the available hardware. The simplest answer is that for the overwhelming majority of situations the speed improvement will be greater if additional machines are added to a DSO than if the same CPU's are added as multiple processors. That simple answer assumes that there is enough memory for the DSO simulations.

For multiple DSO simulations on a single machine the total memory needed is the sum of the memory used by each simulation. For example, assume we are running discrete frequency sweep and each frequency point needs 3.5GB. Our computer has only 8GB of RAM but is a quad core system. To keep the computer from going heavily into swap, which is highly inefficient, we would only want to assign this computer twice to the list of machines in the DSO setup. To take advantage of the remaining 2 CPUs then it would make sense to set the number of distributed processors to 2. This would be the optimum for this setup. In addition one would need to be sure that the amount of disk space available is also sufficient to fit all the requested simulations - typically hard drive space is not the limiting factor.

Related Topics

Configuring Distributed Analysis

Editing Distributed Machine Configurations

Large Scale DSO for Parametric Analysis

Large Scale DSO for parametric analysis operates through a non-graphical batch application called desktopjob. You can run the desktopjob command-line to perform parametric analysis DSO. The command-line interface supported by this batch program is consistent with the command-line used for regular DSO jobs.

Large Scale DSO is used for 'large scale parallel' jobs, which either fail or scale poorly as Regular DSO jobs. A Large Scale DSO job does not support the output of full parametric results, but produces 'reduced' datasets corresponding to predefined Rectangular plots. The extracted columns of data are saved as .csv files. Typically, there is one .csv file per-trace, per-variation. These .csv outputs can be used directly in downstream applications (for example, Excel, or custom programs that parse .csv files). They can also be Imported as Dataset Solutions for post processing. Non-Rectangular plots of the design (such as statistical eye or digital plot) are not extracted). In order to produce a new output you must re-run the analysis.

The basic process involves:

1. Prepare the model for Large Scale DSO Analysis
2. Submit the Large Scale DSO Job through the Tools>Job Management menu, or via a command line.
3. Monitor the job's progress
4. Post process the results

For details, refer to the following sections:

- Prerequisites for Large Scale DSO
- Job Management Interface for Large Scale DSO
- Command Line Syntax
- Deployment/Configuration
- Tutorial Example for HFSS
- Results Database Organization
- Job outputs
- Job Monitoring
- Known Issues for Large Scale DSO
- Troubleshooting for Large Scale DSO

Related Topics

Running HFSS from a Command Line

Distributed Analysis

High Performance Computing (HPC) Integration

Technical Notes: Large Scale DSO Theory

Prerequisites for Large Scale DSO

General Prerequisites

- HFSS must be installed on the cluster which runs either a supported scheduler or ANSYS RSM.
- The cluster is compatible with Large Scale DSO Requirements
- There is a folder on a shared drive where the input projects are located.
- Every node of the cluster supports the disk space (in temp directory) and memory requirements of multiple engines that un in parallel.
- All the machines allocated to Large Scale DSO job must all come from the same platform, Windows or LINUX.

Ansoft RSM Environment

On the Windows platform, RSM is started as an 'admin' account, rather than as a 'system' account.

Note Large Scale DSO does not support RSM Service running with 'system login' credentials.

- On each machine of the cluster, 'desktopjob' application is registered with Ansoft RSM service using the command shown below:

  Windows: <installation-directory>/<platform> desktopjob.exe -regserver

  LINUX: <installation-directory>/<platform>desktopjob -regserver

Note Detailed cluster configuration instructions are here.

Scheduler Environment

- No extra configuration is needed.

Related Topics

Large Scale DSO for Parametric Analysis

Distributed Analysis

High Performance Computing (HPC) Integration

DSO Behavior in HFSS and HFSS-IE

In general two level DSO can only happen when 2 different distribution technologies (engine and MPI) are used. We never do two-level with the same technology (MPI and MPI for example)

HFSS and DSO

What's supported?

HFSS supports the following job distribution options

- Frequencies (engine)
- Variations (engine)

- Transient excitations (engine)
- Domain solver (MPI)
- Direct solver (MPI)
- Iterative solver (MPI)

HFSS supports two-level distribution.

What will we distribute?

HFSS will distribute Optimetrics unless:

- There is only 1 machine in the pool.
- Two-level distribution is disabled and MPI is required (see below).
- Two-level distribution is enabled and MPI is required, but dividing the pool by the 2 level count would not leave enough machines for each MPI solve.

If Optimetrics variations are not distributed, we might distribute frequencies. Discrete and interpolating sweeps are distributable. Fast sweeps are not. As with Optimetrics, if MPI is required it takes precedence over variations. So we need to be able to distribute enough MPI machines for each frequency.

Next, we look whether the setup is requesting the iterative solver. If we are not distributing Optimetrics, we will distribute iterative solvers via MPI. If we are distributing Optimetrics and 2 level is enabled, then we need enough MPI solvers for each variation in order to distribute iterative solvers.

Note MPI distribution of domain and direct solvers is of higher precedence than Optimetrics. But, Optimetrics is of higher precedence than iterative solvers.

Finally, if we are a transient network problem there is no MPI. So, we will distribute transient excitations unless we are already distributing variations.

When is MPI required?

If the current solve setup (for example. “Setup1”) has domains enabled, MPI is required if any of these are true:

- if there is an array
- if there are no FEBI or IE regions
- the DSO configuration enables domain distribution and we have a FEBI or an IE region

If the current solve setup does not enable domains, MPI is required if the Setup selected the direct solver and the configuration enables Direct solver distribution

How many MPI solvers are needed?

We can always use more, but the minimum requirement is

1. If we are using domains
    a. If it’s an array problem, 3
    b. If there’s a FEBI and an IE region, then 4
    c. For all other direct solves, 3

2. Otherwise 2

HFSS-IE

What's supported?

- Frequencies (engine)
- Variations (engine)
- IE Solvers (MPI)

HFSS-IE supports 2 level distribution.

What will we distribute?

As with HFSS, MPI solves take precedence over engine solves. We will always distribute IE solvers if enabled and there are enough machines to use.

We will distribute variations if

- IE solver distribution is disabled
- IE solver distribution is enabled and 2 level distribution is enabled and there are enough machines in each variation to distribute IE solvers

If we are not distributing variations, we will distribute frequencies if the above criteria are met.

What's supported?

HFSS-IE supports the following job distribution options

- Frequencies (engine)
- Variations (engine)
- Transient excitations (engine)
- Domain solver (MPI)
- Direct solver (MPI)
- Iterative solver (MPI)

HFSS-IE supports two level distribution.

What will we distribute?

HFSS-IE will distribute Optimetrics unless

- There is only 1 machine in the pool
- Two level distribution is disabled and MPI is required (see below)
- Two level distribution is enabled and MPI is required, but dividing the pool by the 2 level count would not leave enough machines for each MPI solve.

If Optimetrics variations are not distributed, we might distribute frequencies. Discrete and interpolating sweeps are distributable. Fast sweeps are not. As with Optimetrics, if MPI is required it takes precedence over variations. So we need to be able to distribute enough MPI machines for each frequency.

Next, we look whether the setup is requesting the iterative solver. If we are not distributing Optimetrics, we will distribute iterative solvers via MPI. If we are distributing Optimetrics and 2 level is enabled, then we need enough MPI solvers for each variation in order to distribute iterative solvers.

.

Note MPI distribution of domain and direct solvers is of higher precedence than Optimetrics. But, Optimetrics is of higher precedence than iterative solvers.

Finally, if we are a transient network problem there is no MPI. So, we will distribute transient excitations unless we are already distributing variations.

When is MPI required?

If the current solve setup (for example. “Setup1”) has domains enabled, MPI is required if any of these are true:

- if there is an array
- if there are no FEBI or IE regions
- the DSO configuration enables domain distribution and we have a FEBI or an IE region

If the current solve setup does not enable domains, MPI is required if the Setup selected the direct solver and the configuration enables Direct solver distribution

How many MPI solvers are needed?

We can always use more, but the minimum requirement is

1. If we are using domains
   a. If it’s an array problem, 3
   b. If there’s a FEBI and an IE region, then 4
   c. For all other direct solves, 3
2. Otherwise 2

General Notes

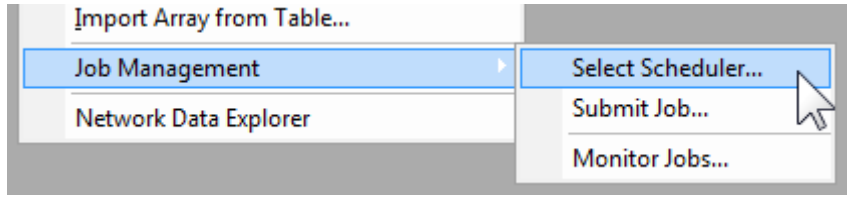
In the case of an uneven distribution, the smallest pool is used to determine capability. For example, if we have 8 machines and the user wants 2 level solve of a frequency sweep with 3 machines at level 1, that would divide the machines into (3,3,2). If this is an array problem, we would need 3 machines for each pool. So, we would disable 2 level distribution (rather than reconfigure to only have 2 pools).

The user’s selection of (say) frequency distribution or 2 level does not *require* that we do so. In the above example, if we are adapting we might choose to devote all 8 machines to the array solve at each adaptive pass as there is neither frequency nor 2 level possible.

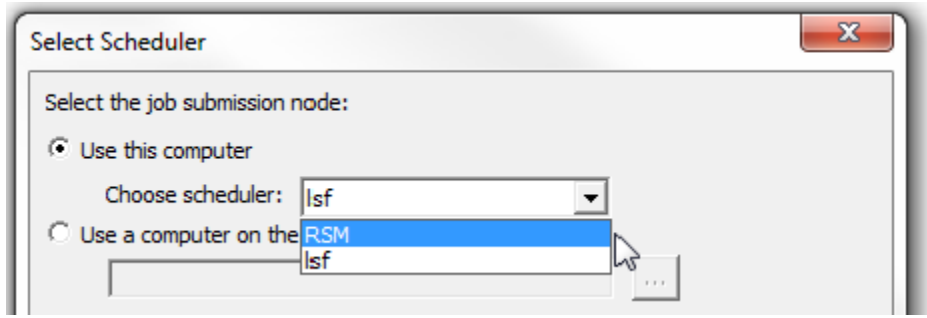
Job Management Interface for Large Scale DSO

Large Scale DSO jobs run only in non-graphical batch mode, irrespective of the scheduler environment. This is in contrast to a Regular DSO job, which, in RSM environment, can be launched from a graphical Desktop. This consideration implies that input project corresponding to Large Scale

DSO job must be saved and closed, prior to job-submission. Secondly, the command to submit Large Scale DSO job is only available through the **Tools>Job Management** menu or a command window, while a Regular DSO job can be run in RSM environment by right-clicking directly on the parametric setup. The Job Management UI is accessed by running ANSYS Electromagnetics product Desktop on the designated 'Postprocessing node' of the cluster. The Desktop provides UI commands for Scheduler selection, Job submission and Job monitoring/control. You access the Scheduler User Interface by clicking **Tools>Job Management>Select Scheduler...**

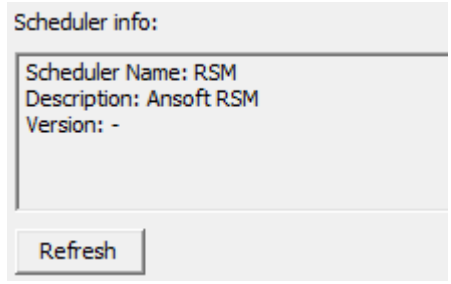


Click Select Scheduler to display the selection dialog. A drop down lists potential schedulers, (which can include RSM, lsf, or sge, depending on the environment).



If you select a scheduler that is not supported in your environment, you receive a warning message.

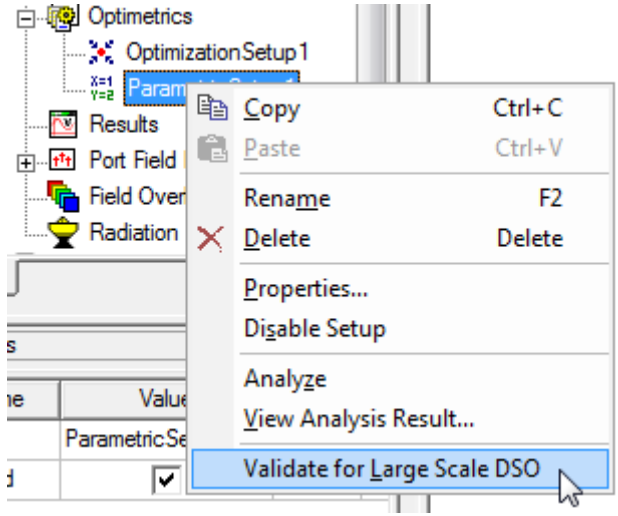
After selecting a scheduler, you can click Refresh to display information for that scheduler.



Once you have selected a scheduler supported in your environment, you can go through the following steps to submit a Large Scale DSO job.

1. Setup and prepare model on local workstation
   - Launch Desktop. Open project, for example, '/home/projects/spool/test.adsn'
   - Suppose the variations to solve come from ParametricSetup1 setup, Right-click on Para-

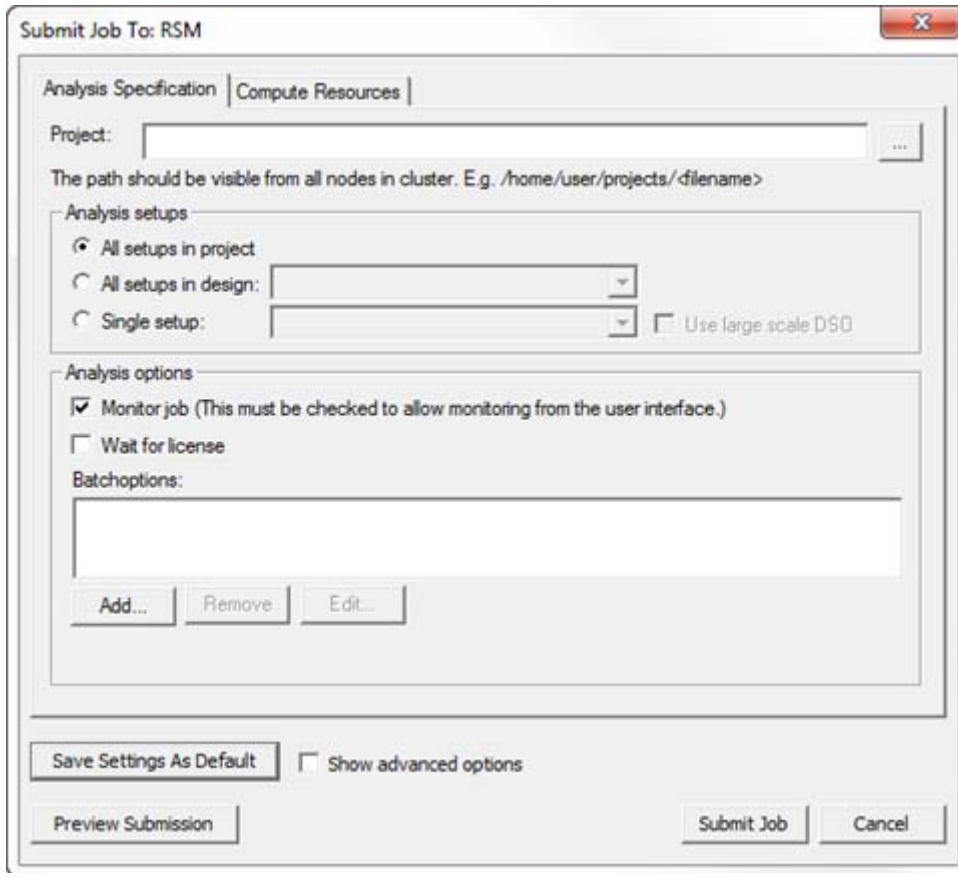
metricSetup1 and run 'Large Scale DSO/Validate'. Fix any validation errors.



- Save project, in case of edits.
- Close project.

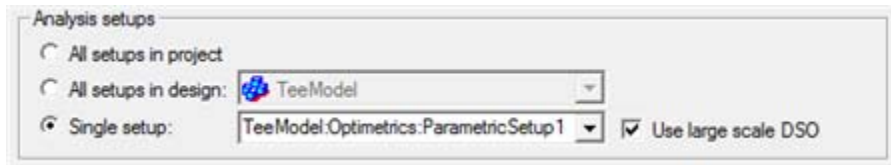
2. Copy the input project (or folder, if the project references external files) from a personal workstation to a shared-drive on cluster (say project is copied to /home/projects/spool/test.adsn).
   - In the RSM environment, you are required to specify a machine-list. (See the HPC and Analysis Options.) For example, say the machine-list is: 3 cores from 'm1' and 3 cores from 'm2', for a total of 6 engines. You select the list on the Compute Resources tab described below.
   - In a Linux scheduler environment, a cluster must have a designated 'postprocessing node'. Open a remote-desktop session (or equivalent such as vnc session) on the designated 'postprocessing node'. Say the name of this node is 'm1'. Launch Desktop graphically on 'm1'.
3. Open a remote-desktop session (or equivalent such as vnc session) on the node corresponding to the first machine of job's machine-list, 'm1' in this case. Launch Desktop graphically on 'm1'.
4. (Optional step) Double-check that the model is prepared correctly
   - Open project '/home/projects/spool/test.adsn'
   - Right-click on ParametricSetup1 and run 'Large Scale DSO/Validate'. Fix any validation errors
   - Save project, in case of edits. Close project

5. Run **Tools>Job Management>Submit Job...** The standard Job Submission panel pops up.



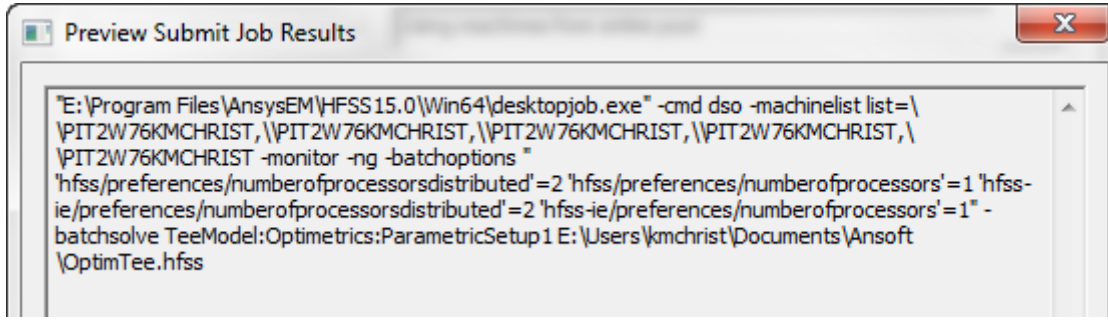
The panels for LSF and GE have some differences. See separate sections for Job Management User Interface for LSF and Job Management User Interface for SGE.

- Enter all fields. Make sure to select 'ParametricSetup*n*' for analysis and check the Use large scale DSO checkbox.



- To see the command-line submitted to the scheduler, click **Preview Submission.** This

opens a dialogue showing the command to be sent to the scheduler.

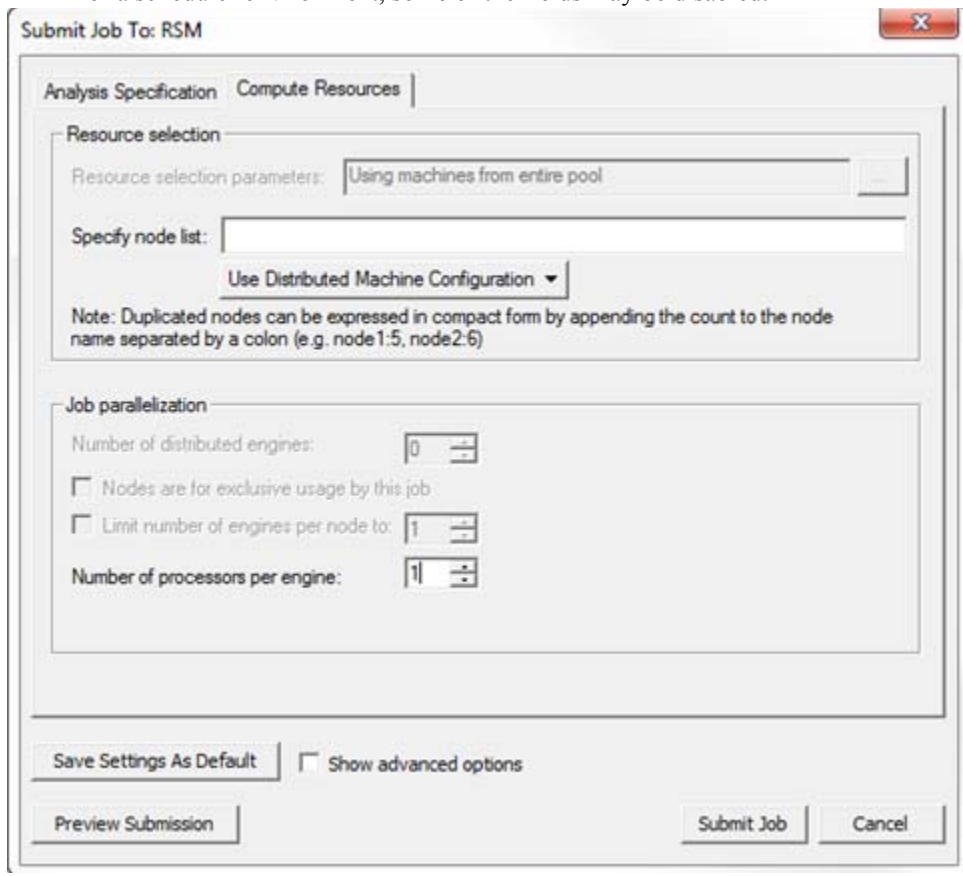


The text can be copied to the clipboard, if desired.

- The Batchoptions field allows you to add additional -batchoptions parameters.
- If you intend to Monitor the job through a user interface, you must check Monitor job. You can then monitor this job through the **Tools>Job Management>Monitor Jobs...** command or by checking the dialog that opens when you submit the job.

6. The Compute Resources tab displays other parameters. Depending on the resources available

for a scheduler environment, some of the fields may be disabled.



- Specify node list

  Checking Specify node list enables the field for specifying a node list, and the drop down for **Use Distributed Machine Configuration** menu, that lets you select from pre-defined lists. In a computing environment where the available cores are not uniform, you can use this to have control over which resources your job will use.

- Job parallelization:

  The values you specify here represent minimal requirements for each condition that can interact in leading to the total resources the Scheduler derives from them. If you specify a node list in the Resource selection area, that takes priority over any values specified in Job parallelization (which are then ignored). Depending on the resources available for a scheduler environment, some of the fields may be disabled.

- Number of distributed engines
- Whether Nodes are for exclusive usage by this job
- Whether to Limit number of engines per node to a value

This can be useful in situations where the amount of memory available for node is limited relative to the requirements for the project, and you want to ensure sufficient memory per process.

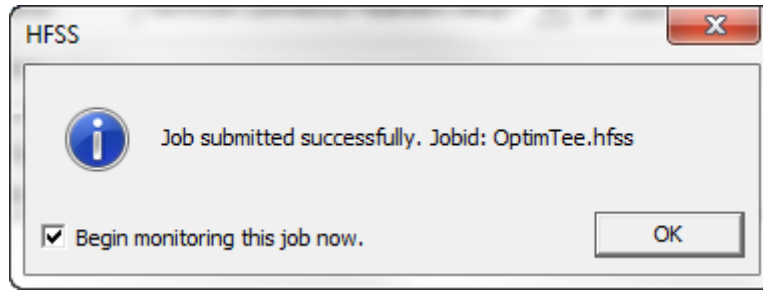
- Number of processors per engine
- Number of processors, non-distributed (visible only when **Show advanced options** is checked.)

  In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines, with two processors per engine, and also limit the number of engines per node to 4, the scheduler may increase the number of cores used in order to meet the limit specified for engines per node. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification

7. To submit the command with the specified parameters, click **Submit Job**.

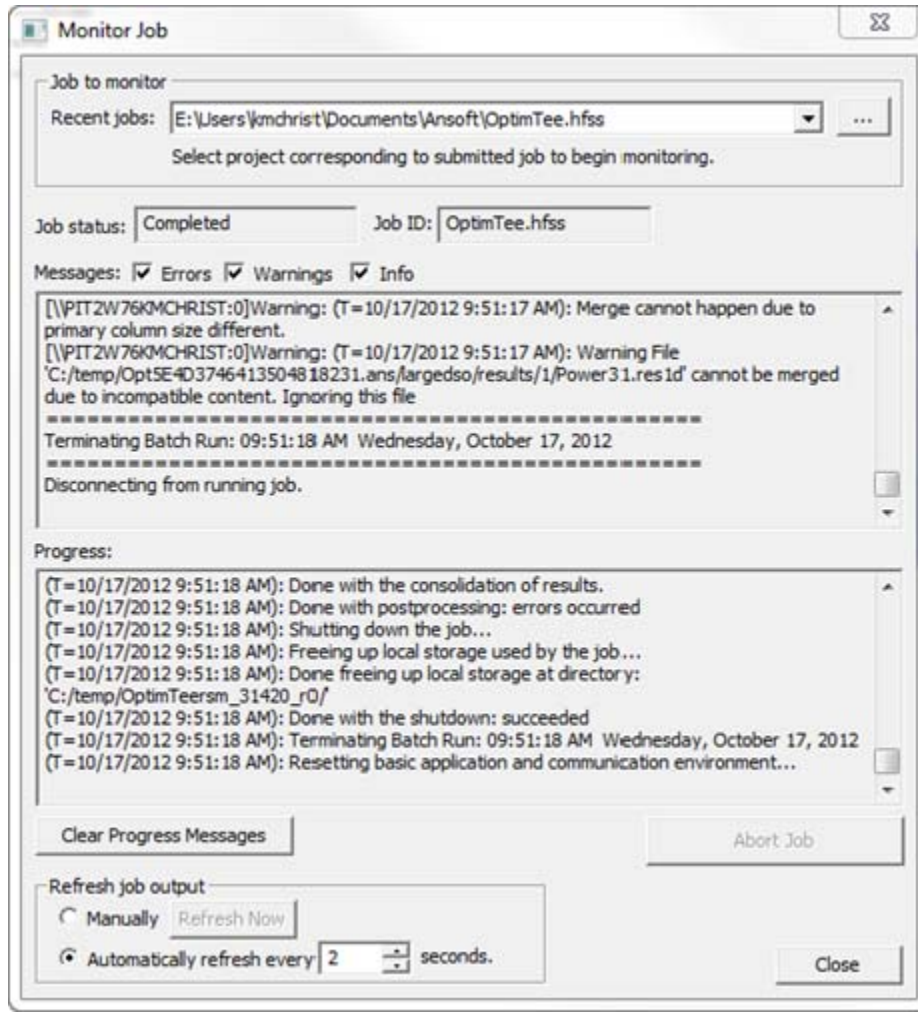
Note The RSM environment does not support for queuing, so 'Submit Job' will immediately start running the job.

A dialog displays in which you can check "Begin monitoring this job now."



8. You can monitor this job either automatically (by checking the option) or through the

Tools>Job Management>Monitor Jobs... command.



The dialog contains fields reporting the job status, job ID, messages issues, and progress. You can filter the messages for Errors, Warnings, and Info. By option you refresh the job manually or automatically at specified intervals.

The process for submitting and monitoring Large Scale DSO jobs in LSF/SGE environments. is generally similar.

1. Setup and prepare model on local workstation

   Launch Desktop. Open project '/home/projects/spool/test.adsn'

   Suppose the variations to solve come from ParametricSetup1 setup, Right-click on ParametricSetup1 and run 'Large Scale DSO/Validate'. Fix any validation errors

   Save project, in case of edits

Close project

2. Copy the input project (or folder, if the project references external files) from personal workstation to shared-drive on cluster (say project is copied to /home/projects/spool/test.adsn)

   In the scheduler environment, a cluster must have a designated 'postprocessing node'. Open a remote-desktop session (or equivalent such as vnc session) on the designated 'post-processing node'. Say the name of this node is 'm1'. Launch Desktop graphically on 'm1'.

- (Optional step) Double-check that the model is prepared correctly

  Open project '/home/projects/spool/test.adsn'

  Right-click on ParametricSetup1 and run **Large Scale DSO/Validate.** Fix any validation errors

  Save project, in case of edits. Close project

3. Run **Tools>Job Management>Submit Job...**. The standard Job Submission panel pops up, which is documented as part of the scheduler-gui-integration feature.
4. Enter all fields. Make sure to select 'ParametricSetup1' analysis and turn ON 'Use large scale DSO' checkbox
5. Click on 'Preview' to check (and/or copy to clipboard) job's command-line. Click on **Submit Job**.
6. Monitor this job through the **Tools>Job Management>Monitor Jobs...** command. See Scheduler GUI Integration for details regarding the job submission and monitoring panels

See separate sections for Job Management User Interface for LSF and Job Management User Interface for SGE.

Large Scale DSO Command Line Syntax

Large Scale DSO feature operates through a non-graphical batch application called 'desktopjob'. You can run the desktopjob command-line to perform parametric analysis DSO. The command-line interface supported by this batch program is consistent with the command-line used for current DSO jobs. "desktopjob -help" lists all available command-line options as shown below:

Command Line Syntax:

desktopjob.exe <options> <project-path-on-shared-drive>

Options:

-help: Print this help text

-cmd: Specify command to run. Available choices: dso

-ng: Run analysis in non-graphical mode

-monitor: Output progress and messages to standard output/error

-waitforlicense: Queue the job until the availability of licenses

-preserve: Preserve local storage space of the distributed job for investigation into job's run. If local storage directory (aka, temp directory) is provisioned by scheduler, ensure it is also con-

figured to preserve job's local storage. (Note: this storage should be deleted manually)

-batchoptions: Override the Tools/Option entries through either a batchoptions file or batchoptions string.

Example:

```
-batchoptions <config-file-on-shared-drive>
-batchoptions "'name1'='val1' 'n2'='v2'"
```

-machinelist:

- In the context of **Ansoft RSM**:

  Specify machines for distributed analysis. Machine list is specified either inline (as a comma separated machine names) or through a file. Multiple cores are specified by repeating the name of machine or by embedding number of cores in the machine name, using a colon separator.

  Example 1:

  ```
  -machinelist "list=m1,m1,m1,m2,m2,m3"
  ```

  Example 2:

  ```
  -machinelist "list=m1:3,m2:2,m3"
  ```

  Example 2:

  ```
  -machinelist "file=machines.txt"
  ```

- In the context of **a scheduler such as LSF**:

  Specify the portion of total machines for distributed analysis. Use remaining for overhead or shared memory multiprocessing.

  Example:

  ```
  -machinelist "Num=10"
  ```

-usefolderasinput: Choose this option if the job's input represents the entire folder rather than just the project file.

-maxfolderInMB: Specify the maximum size (MBytes) of input folder that is allowed for a valid job. By default, the maximum size allowed for input is 10MB. Specify a value of 0 to remove this size restriction and enable inputs of any size. Note: this option applies when '-usefolderasinput' is used.

-workdir: Specify the shared drive folder for status and result files generated by analysis. By default, the results folder of input project is used as the work directory.

-batchsolve: Solve the specified parametric setup. Syntax for the setup:

```
<design-name>:Optimetrics:<parametric-setup>
```

Related Topics

Large Scale DSO for Parametric Analysis

Running HFSS from a Command Line

Job Management Interface for Large Scale DSO

Large Scale DSO Job outputs

A large-scale-dso analysis does not support the output of full parametric results. Instead, it extracts 'subset' results using predefined Rectangular plots, which are created by user before job is run. The extracted columns of data are saved as CSV files. Typically, there is one CSV file per-trace, per-variation. (Note: Non-Rectangular plots of the design (for e.g. statistical eye, digital plot) are not extracted). The outputs can be either imported as datasets for post-processing in the desktop also as function of parametric variations, or used directly in downstream applications (for example, Excel, or custom programs that parse .csv files).

CSV File contents

The initial header rows of CSV file define the solved variation. For each such row, the first column has variable name and the second column has variable's value. The row following variation rows has the name of primary sweep and the name(s) of extracted quantities. Subsequent rows contain 'data' - quantity values as a function of primary sweep. Below examples provide a context for the CSV file contents:

- Traces of S-parameter report: The data portion of CSV file has 2 columns of data: first column has Freq values and the second column has values for trace's s-parameter component
- Trace of a far field report: Suppose there is a farfield report with a trace ('magrE'), whose primary sweep is 'phi' and secondary sweep is 'theta'. Further suppose that two values of 'theta' are chosen and 'all' values of 'phi' are chosen. For this trace, the data portion of CSV file contains three columns of data: the first column has phi values, the second column has magrE values for the first value of theta, the third column has magrE values for second value of theta. The magrE output columns are titled as 'magrE_crv1', 'magrE_crv2' respectively
- (Advanced) 'Special' sweeps: In the case of a trace with 'special' primary sweep (such as the trace of a time domain quantity), one CSV file is created per curve of trace, per variation. These CSV files always have two columns, irrespective of the number of values chosen for secondary/higher sweeps.

Related Topics

Large Scale DSO for Parametric Analysis

Import Large Scale DSO Dataset Solution

Large Scale DSO Tutorial Example

This section provides an example of the use of Large Scale DSO to distribute parametric variations of an HFSS model across the nodes of a cluster or to multiple cores of a single machine.

This example presumes that your configuration satisfies the Prerequisites for Large Scale DSO. For this example, pre-suppose that we have a Windows cluster. Further suppose that the shared drive folder, which contains the input projects and computed results is at the location

\\sjo7na1\hfssprojs

Major steps for Large Scale DSO Example

1. Prepare the model for Large Scale DSO Analysis
2. Submit the Large Scale DSO Job

3. Post process the results

Prepare the Model for Large Scale DSO Analysis

1. Launch the desktop.
2. Create the input project on the shared drive.

   For this example, start with the standard HFSSOptimTee.hfss example and copy it to the shared drive.

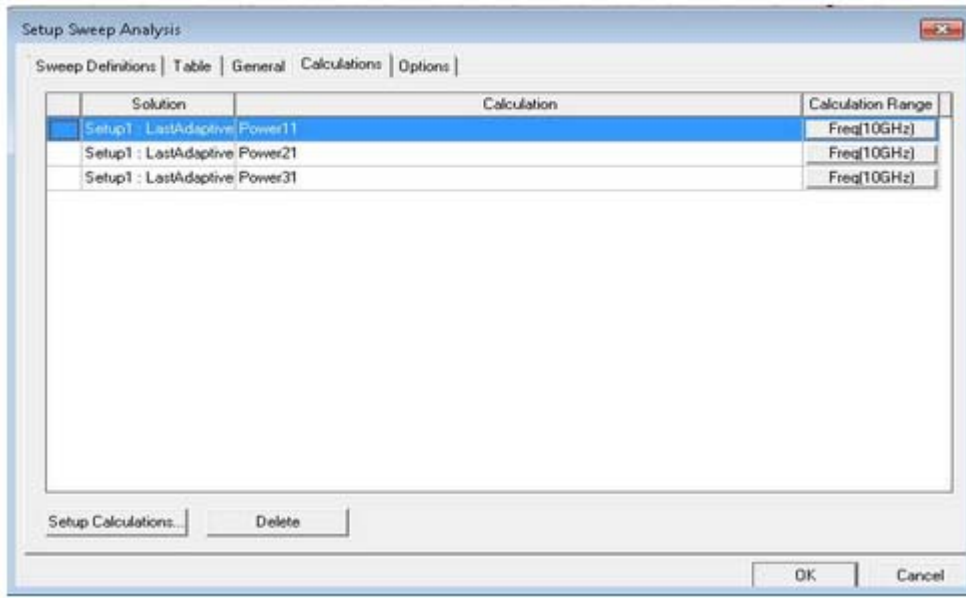
   Copy "<installation-directory>\<platform>Examples\RF_Microwave\OptimTee.hfss" to "\\sjo7na1\hfssprojs\OptimTee.hfss"
3. Setup the parametric table.

   For this example, use the existing 'ParametricSetup1' as the parametric setup to solve.
4. Prepare the outputs.

   Outputs from Large Scale DSO come from pre-defined rectangular plots that are created before the Analysis command is issued. Follow the steps below:

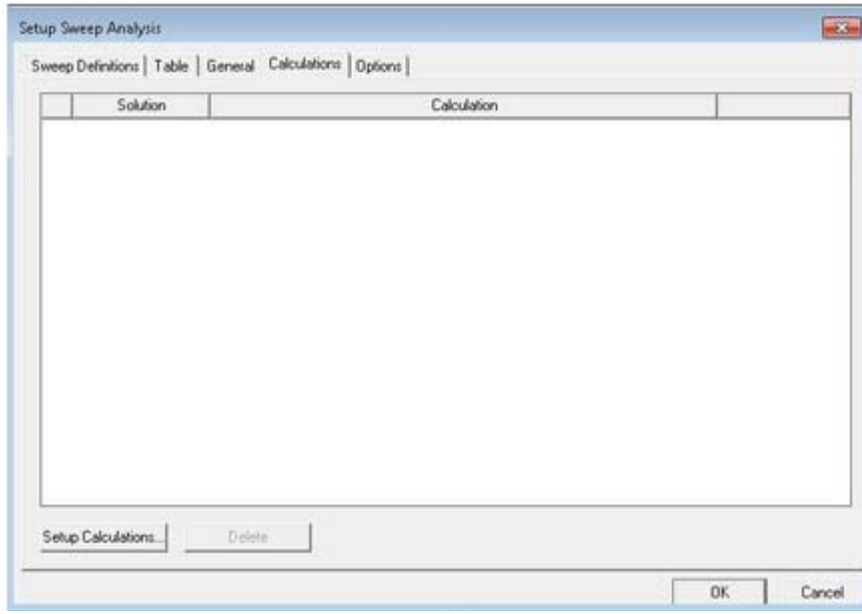
   a. Because these DSO outputs come solely from Rectangular Plots, delete all other postprocessing setups, and then turn off Save Fields And Mesh as shown below.

      Open ParametricSetup1 for editing. You will see that the OptimTee parametric setup contains three calculations, as shown below.

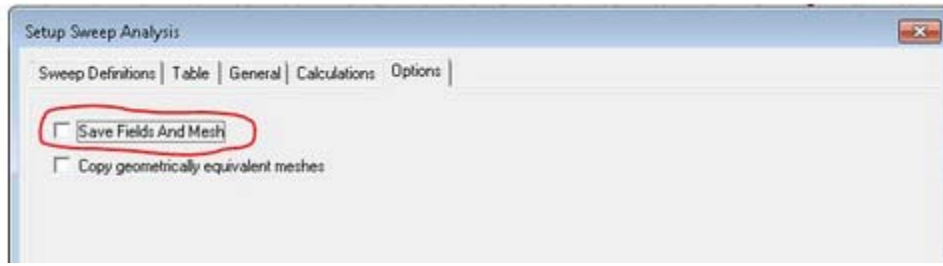


   b. Delete all three calculations. When you have done so, the Setup Sweep Analysis Calcula-

tions tab looks like this:



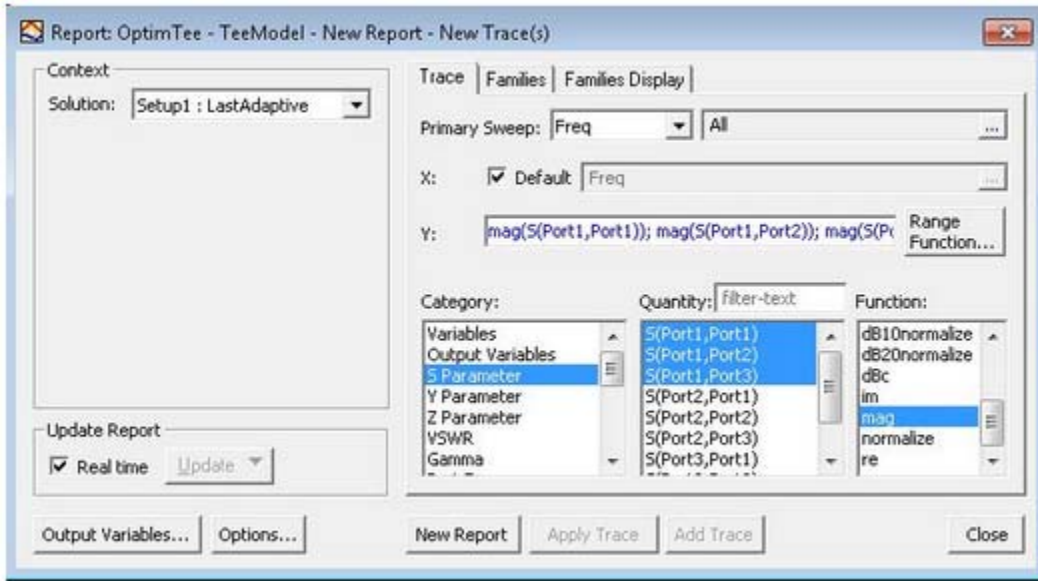
c. Click on the **Options** tab and uncheck Save Fields And Mesh, as shown below:



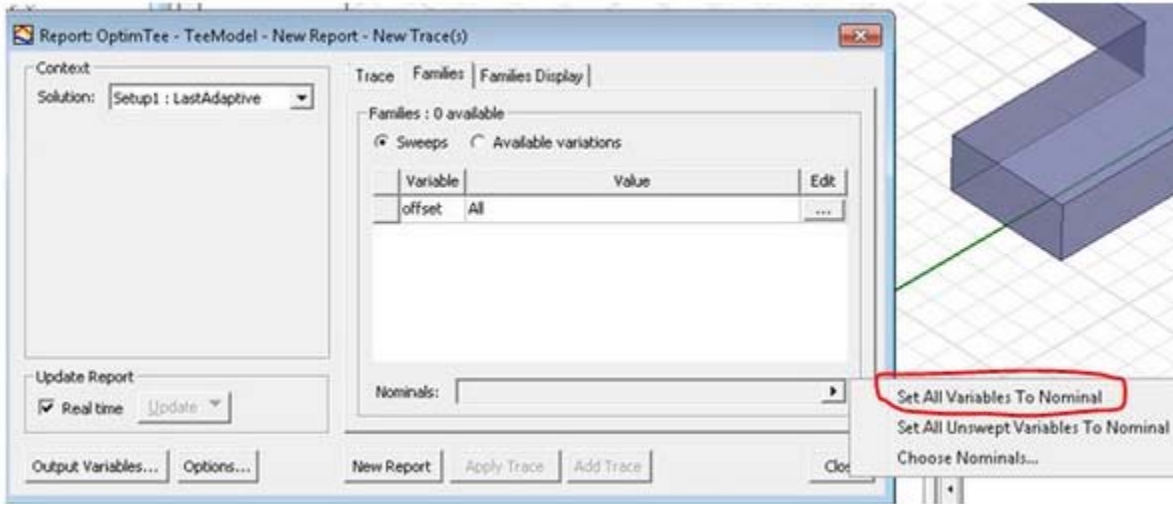
5. Use the Reporter to define outputs.

   For this example, you add six traces that correspond to six csv outputs of the Large Scale DSO job: mag(S11), mag(S12), mag(S13), Power11, Power12, and Power13.

a. As shown below, select three quantities: mag(S11), mag(S12), and mag(S13).



b. Click on the Families tab and ensure that all variables are set to Nominal, as shown below.



Note For Large Scale DSO, outputs are not extracted correctly unless all variables on the **Families** tab are set to nominal.

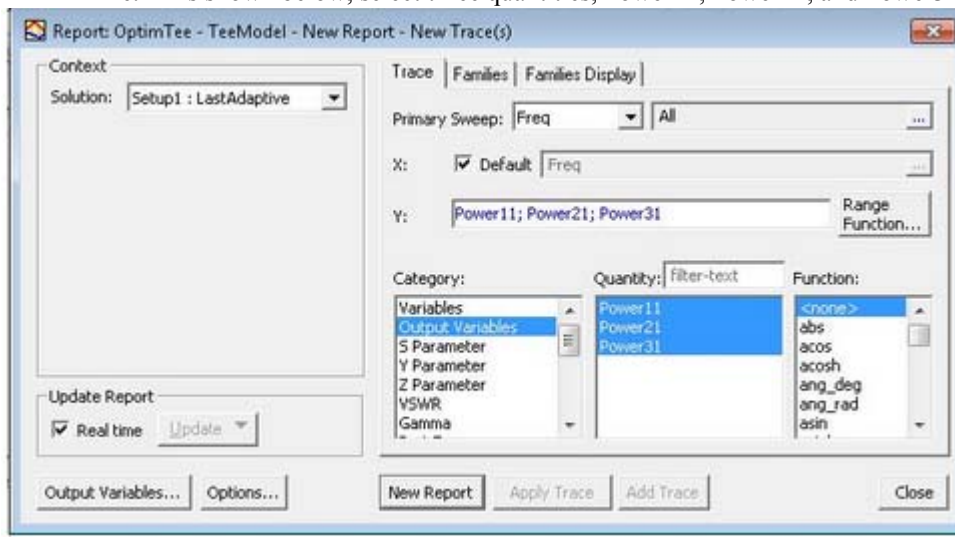
c. Click the **New Report** button.

Use the Report to create a Power distribution plot that has three power distribution traces.

d. To create an S-parameter plot that has three traces, click **Results>Create Modal Solution**

Data Report>Rectangular Plot.

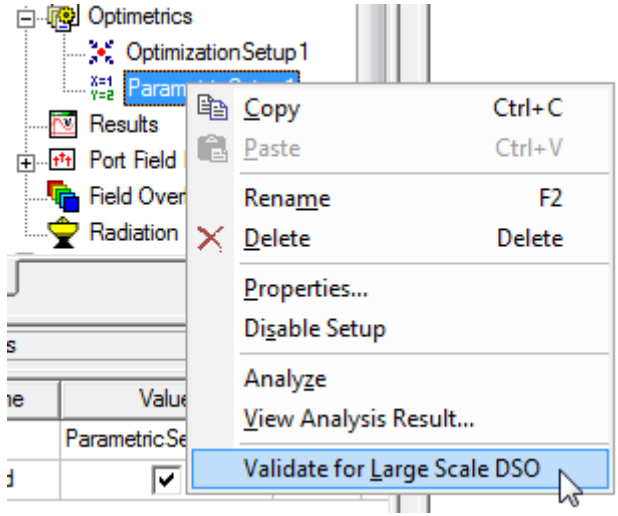
e. As shown below, select three quantities, Power11, Power21, and Power31.



f. Click on the Families tab and ensure that all variables are set to Nominal.

g. Click the **New Report** button.

6. Right-click on the parametric sweep, and select Validate for Large Scale DSO.



7. A dialog reports any errors. These may occur if the steps just taken are neglected:

```
[error] Please remove all the calculations in Parametric Setup/
Calculation page.
```

```
[error] Please turn off "Save Fields And Mesh" in Parametric
Setup/Options page.

[error] No rectangular plot exist in the design.  Please create
a rectangular report.
```

8. Correct any errors, if necessary to pass validation.
9. Save and Close the project.

Next

Submit the Large Scale DSO Job: Examples

Related Topics

Large Scale DSO for Parametric Analysis

Submit the Large Scale DSO Job: Examples

This section includes examples of submitting a Large Scale DSO job using Ansoft RSM and a Scheduler.

Using the Ansoft RSM Environment

1. Before submitting the job make sure that the input project is not open in any HFSS Desktop window.
2. For this example, suppose that there are two quad-core machines on your cluster with the names "m1" and "m2". Further suppose that there are two engines per machine, for a total of four parallel engines. Let the number of processors allocated to each engine be 1.
3. From a command prompt, issue the following command:

```
<installation-directory>/<platform>desktopjob.exe -cmd dso -
machinelist "list=m1:2,m2:2" -batchoptions
\\sjo7na\hfssprojs\hfssoptions.txt -batchsolve
"TeeModel"Optimetrics:ParametricSetup1"
\\sjo7na\hfssprojs\OptimTee.hfss
```

where the file \\sjo7na\hfssproj\hfssoptions.txt has the following contents:

```
$begin Config
   'Desktop/Settings/ProjectOptions/NumberOfProcessors'=1
   'HFSS/Preferences/NumberOfProcessorsDistributed'=1
   'HFSS/Preferences/NumberOfProcessorsPostProc'=1
   'HFSS/Preferences/UseHPCforMP'=0
   'HFSS/Preferences/SaveBeforeSolving'=0
   'HFSS/Preferences/MemLiimitHard'=0
   'HFSS/Preferences/MemLimitSoft'=0
   'HFSS/Preferences/HPCLicenceType'='pack'
#end 'Config'
```

4. Suppose the above job is assigned ID "jobID".

Using a Scheduler Environment (such as LSF)

1. Suppose you want to solve variations using four parallel engines, each engine being assigned a single core.
2. From a command prompt, run the following command:

   ```
   bsub -n 4 <installation-directory>/<platform>/desktopjob.exe -
   cmd dso -batchoptions \\sjo7na\hfssprojs\hfssoptions.txt -
   batchsolve
   "TeeModel:Optimetrics:ParametricSetup1"\\sjo7na\hfssprojs\Optim
   Tee.hfss
   ```

   where the file \\sjo7na\hfssproj\hfssoptions.txt has the same contents as the RSM example above.
3. Suppose the above job is assigned an ID "jobid"

Next

Large Scale DSO Example: Post Process the Results

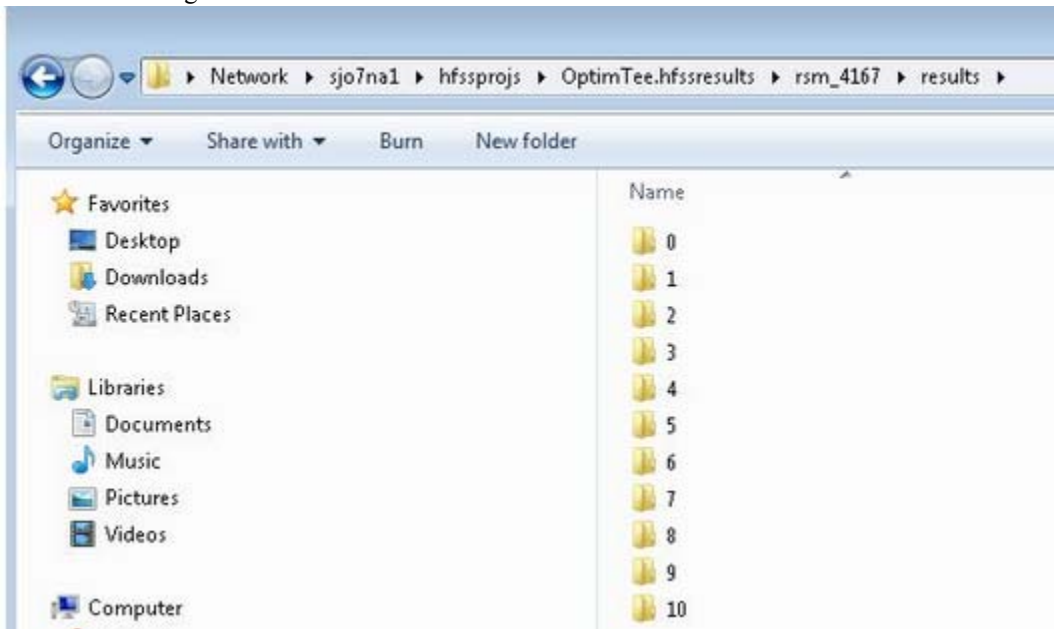
Related Topics

Large Scale DSO for Parametric Analysis

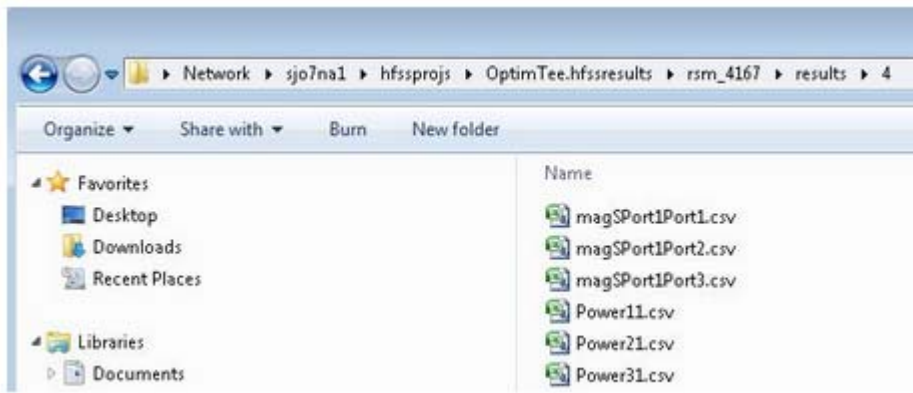
Large Scale DSO Example: Post Process the Results

Once the job is done, output is available in the ~\OptimTee.hfss\jobid\results folder. Each variation creates a subfolder, which in turn has one csv file per trace of each report. See the detailed information regarding job monitoring and the location of the analysis logs.

The figure below shows the results for 10 variations as located in 10 folders.



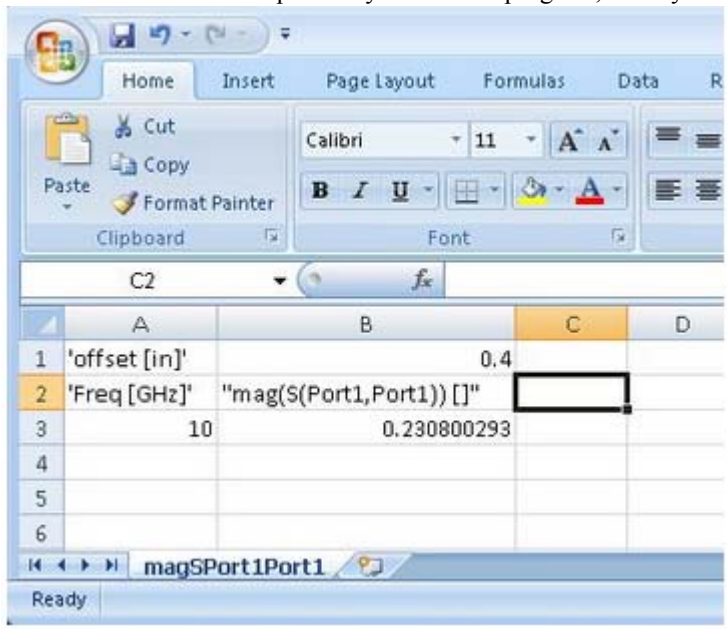
There are six csv files corresponding to three S-parameter traces and three power distribution traces, as shown below for the fourth variation.



You have three options for postprocessing csv files.

- Import Large Scale DSO Dataset Solution
- Use Microsoft Excel or any other application that has csv post processing functionality.

- Parse the csv output into your custom program, for any downstream flow.



Related Topics

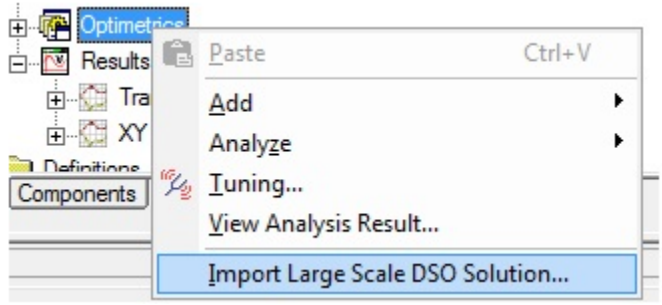
Large Scale DSO for Parametric Analysis

Import Large Scale DSO Dataset Solution

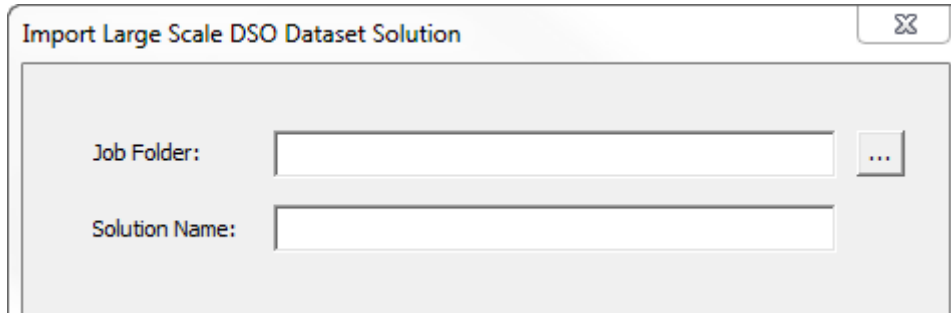
For post-processing Large Scale DSO dataset solutions in the desktop, you create a dataset solution through the **Import Large Scale DSO Solution** command and pointing to Large Scale DSO job's top level results folder.

1. Import solved large scale DSO solution. You can do this in two ways.

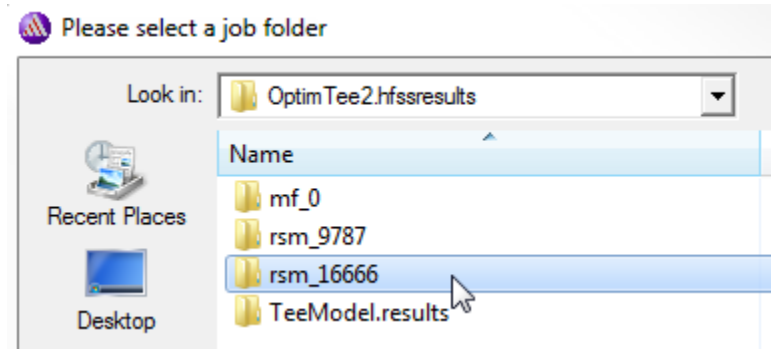
   In the Project tree. right click on Optimetrics and from the menu click **Import Large Scale DSO Solution**.



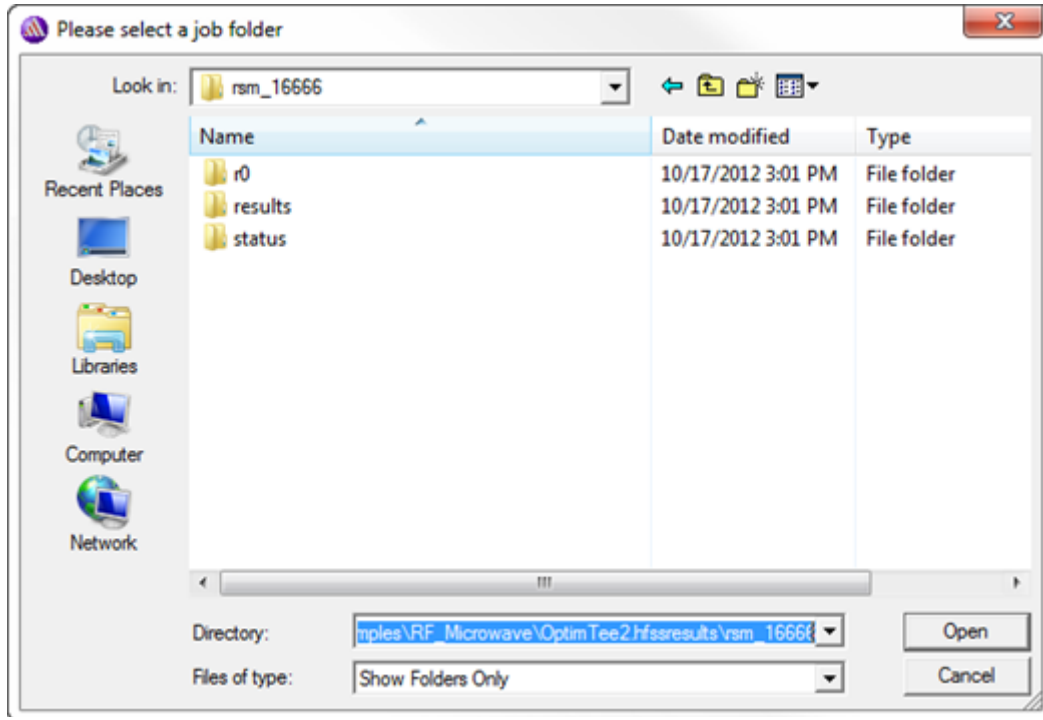
This opens the **Import Large Scale DSO Dataset Solution** dialog.



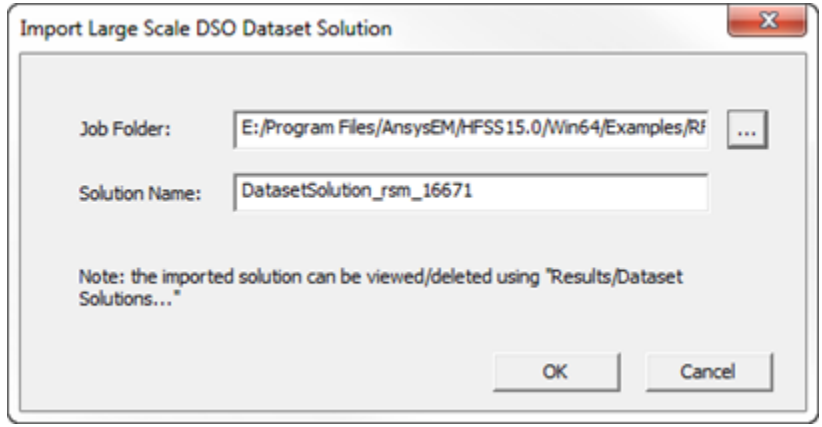
From here you can browse to the results of a Large Scale DSO job and select a job folder. Click the ellipsis button [...] to open the browser window Navigate to the results directory to see the results, organized by the scheduler prefix and job ID number. To select a results data-set, double-click on the results folder name.



This shows the selected Directory path, and lists the contents of the selected directory.



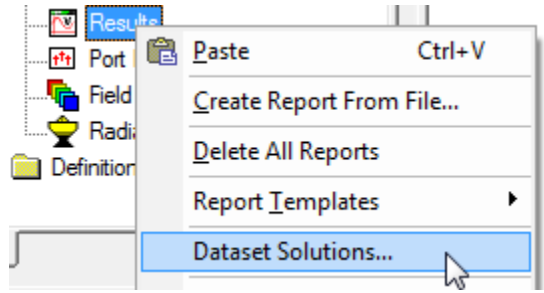
Click Open to show the **Import Large Scale DSO Dataset Solution**s dialog with the Job Folder path and the Solution name listed.



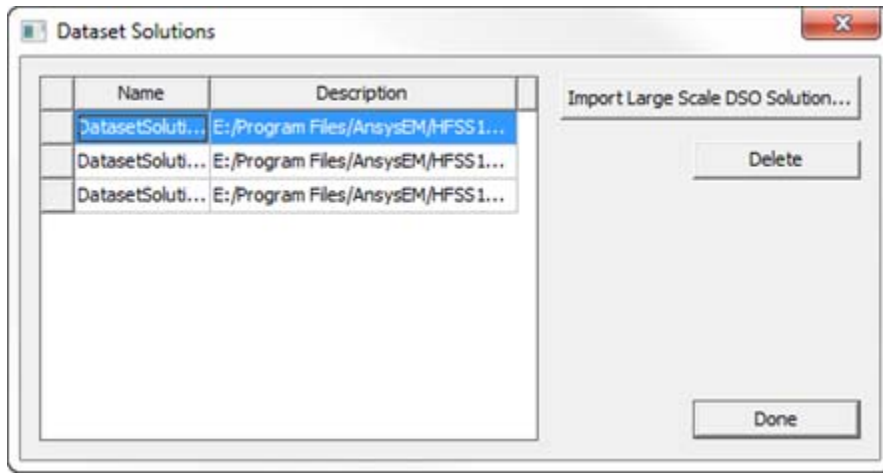
Click, OK to import the Dataset. If you have selected the solution folder correctly the dialog closes. If not, an error dialog opens.

2. To View or Delete the imported dataset solutions:

In the Project tree, right click on Results and from the menu click **Dataset Solutions....**



This displays the Dataset Solutions dialog, listing any existing datasets.

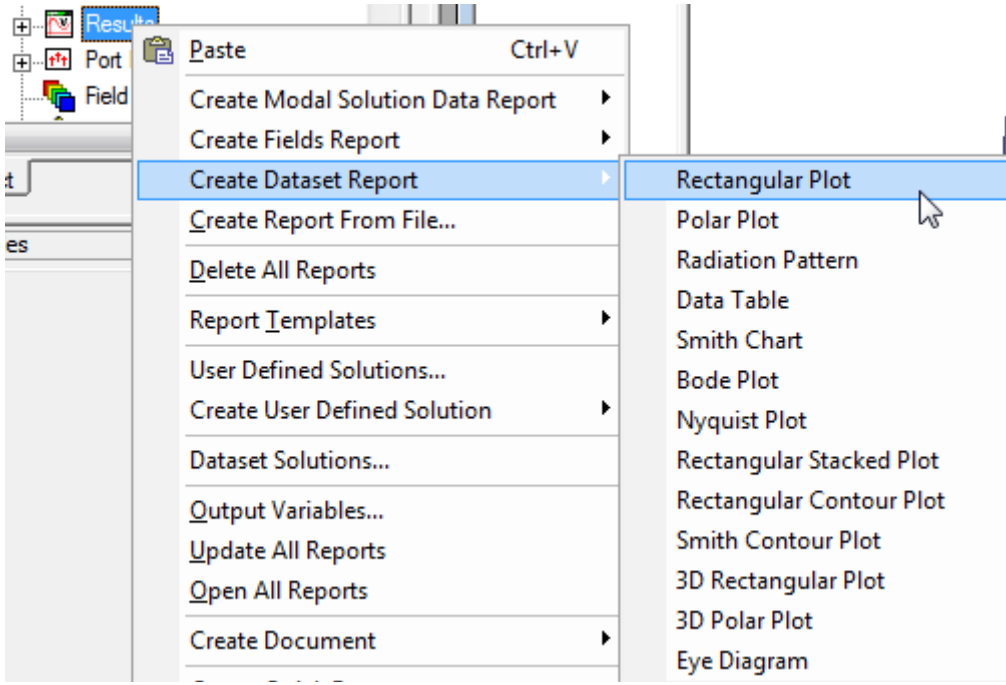


Selecting a listed dataset enables the Delete button. The **Import Large Scale DSO Solution** button also opens the dialog for importing a dataset solution.

3. To create a Dataset Report

   After you have imported one or more DSO solutions, you can create a Dataset Report. Right

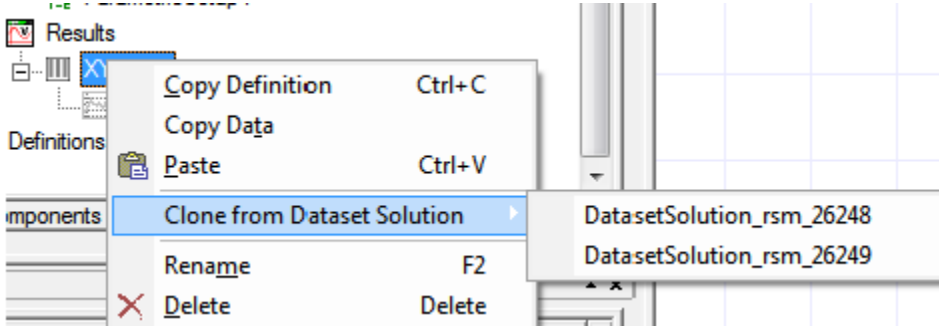
click on Results and from the menu select **Create Dataset Report**, and the type of report.



This opens a reporter window from which you can create a report. If you have previously created an eye diagram report and it is included in the DSO solution data extraction, you can use **Create Dataset Report > Eye Diagram** to recreate this report.

4. Cloning from a Dataset Solution

   If you re-open a project that was solved using large scale DSO, you can quickly clone a report for a solved large scale DSO solution by right-clicking in the Project tree on the report, and choosing Clone from Dataset Solution..., provided this report is qualified for Large Scale DSO data extraction. The provides a way that you can reuse the existing report definition and save the work of creating new report.



Related Topics

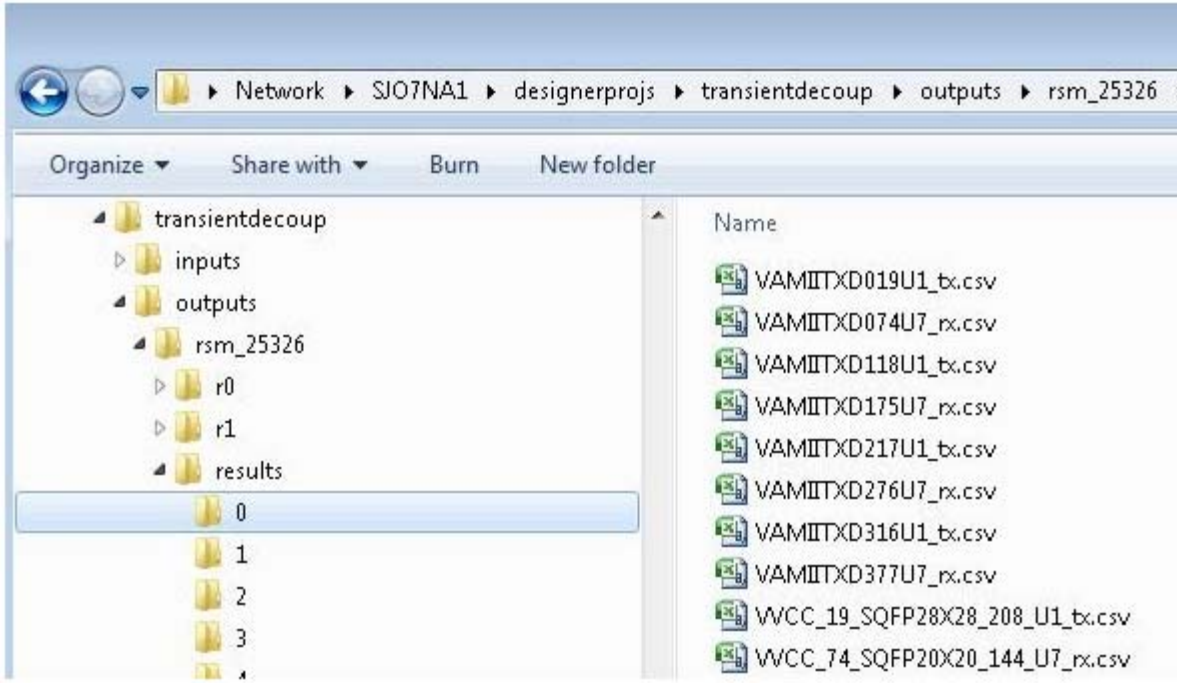
Large Scale DSO Results Database Organization.

Large Scale DSO Results Database Organization

A Large Scale DSO analysis does not support the output of full parametric results. Instead, it extracts 'subset' results using predefined Rectangular plots, which are created by user before job is run. The extracted columns of data are saved as CSV files. Typically, there is one CSV file per-trace, per-variation. (Note: Non-Rectangular plots of the design (for e.g. statistical eye, digital plot) are not extracted)

The results of a Large Scale DSO job are located in the '<workdir>/<jobid>/results' folder. If 'workdir' is not specified on the job command-line, it is same as the input project's results folder. For example the default 'workdir' corresponding to '\\shared\projects\tee.hfss' is '\\shared\projects\tee.hfssresults'. Within this results folder, there is one folder per variation. The name of the variation's folder is an integer number corresponding to variation's index in the parametric table. For example, a variation-folder named '4' has results for the fifth row of parametric table, while a variation-folder named '0' has results for the first row of the table.

Below is a sample results folder showing the contents corresponding to results of first variation. There are ten CSV files corresponding to ten predefined traces.



Related Topics

Large Scale DSO for Parametric Analysis

Large Scale DSO Job Monitoring

Large Scale DSO avoids detailed intra-variation monitoring is avoided as it increases network traffic for large-scale jobs. Large Scale DSO jobs are monitored as below:

- Cluster monitoring tools: The resource usage (CPU, Memory, Network) of Large Scale DSO jobs is monitored using standard cluster monitoring tools. Such job-neutral resource monitoring is ideal as it uses negligible network bandwidth, CPU/Memory.
- Detailed monitoring of analysis of a variation: For any detailed monitoring you must examine the information provided in the job's log files. Specifically, the large-scale-dso job writes detailed logs conveying information regarding the machines where engines are running and the local storage location of per-engine distributed database. With such information, you can login to individual machines for deeper probing of each distributed engine. Following logs are available:
- Per-node logs:

  There is one 'desktopjob.log' file per node assigned to the job. This log contains information regarding the node such as name, local storage folder, number of engines started on this node, etc. It is located in <workdir>/<jobid>/r<nodeIndex>. E.g. "<workdir>/<jobid>/r0" has desktopjob.log corresponding to the engines running on the first node of job, while "<workdir>/<jobid>/r2" has logs corresponding to engines running on third node
- Per-engine logs:

  There is one desktopjob.log file per distributed engine. It is located in <workdir>/<jobid>/r<nodeIndex>/r<coreIndex>. E.g. "<workdir>/<jobid>/r0/r0" has logs corresponding to first engine running on first node, while "<workdir>/<jobid>/r1/r2" has logs corresponding to third engine running on second node. Engine unique information (such as local storage of this engine) is logged here
- Parametric analysis log:

  This log file is located in '<workdir>/<jobid>/r<nodeIndex>/r<coreIndex>' folder and corresponds to Desktop's local-machine parametric 'batchsolve'. It is available only at the end of analysis and contains information regarding the variations solved by this engine and any info/warning/error messages.
- Root desktopjob.log:

  This is the top-level log that logs job distribution information such as hierarchical activation and the list of nodes assigned to this job
- GM specific extraction

  In R14, GM extraction is OFF, by default. It can be turned on through setting environment variable "ANSOFT_ENABLE_GM_EXTRACTION" to a value of 1

Related Topics

Large Scale DSO for Parametric Analysis

Large Scale DSO Deployment/Configuration

LINUX Cluster configuration

- Shared drive for projects: Cluster must provide a shared drive that hosts job inputs - the submitted project must be located on a shared drive (for e.g. a sub-folder of user's home directory). The shared-drive must accessible using the same path on every node of cluster
- 'Temp directory' configuration

  Temp directory is either on 'local storage' or on storage that has equivalent speed characteristics i.e. the I/O rates of the storage should be invariant to network traffic

  Temp directory on a host has sufficient space to hold results database for the variations that are solved on it. Note:

  This storage is freed at the end of the analysis

  The amount of required space depends on the number of engines per node and the cumulative variations solved on this node

  The amount of required space depends on the project's compression-options. For e.g. if 'Save Fields' of a parametric setup is OFF, the space requirement is smaller by the amount of space taken up by field solution data
- Ansoft RSM environment: In the case of supported scheduler environments, there is no extra configuration needed. In the case of Ansoft RSM environment, following additional steps are needed:

  Ansoft RSM must be running on all the nodes of cluster. The credentials of 'RSM service' allow read/write to shared drive. Reason: the remote engine processes are launched using the credentials of RSM service

  Registration of 'desktopjob.exe' with RSM service: 'desktopjob' program must be registered with Ansoft RSM using 'desktopjob -regserver". To ensure that the registration is successful, check that the 'desktopjob' entry in '<RSM-installation-folder>/AnsoftRSMService.cfg' file is valid.

Note LINUX specific critical note: Edit AnsoftRSMService.cfg and replace 'desktopjob.bin' with 'desktopjob'

Major limitation: In the Ansoft RSM environment, Large Scale DSO can only be enabled for one product.

Troubleshooting hints (Ansoft RSM environment only): "shared drive read/write" requirement is a new constraint introduced in Large Scale DSO. So if user runs into a situation where Regular DSO jobs run and Large Scale DSO jobs fail, one possible cause for the failure: RSM service does not have privileges to read and write to project folder located on shared-drive.

Windows Cluster configuration

All the above steps apply, except for steps that are stated as LINUX-specific. Additional instructions:

- Ansoft RSM and ANSYS Electromagnetics products are either installed locally on each node

of cluster (i.e. local installation) OR installed on a single shared-drive available to all nodes of cluster (i.e. network installation)

- Registration of 'desktopjob.exe' with RSM service:
- Network installation: desktopjob.exe is registered with RSM service once, on any of the nodes of cluster
- Local installation: Since each node has it's own RSM installation, desktopjob.exe must be registered with RSM on each node.

Note IMPORTANT! Ansoft RSM service must be started using the credentials of a non-system 'admin' account, which has read/write permissions to project's shared drive. If RSM service runs as 'system' user, large-scale-dso jobs will fail

Heterogeneous Cluster configuration

Limitation: Currently heterogeneous cluster (with both linux and windows nodes) is not supported. This is due to the shared drive requirement.

Related Topics

Large Scale DSO for Parametric Analysis

Known Issues for Large Scale DSO

Cluster configuration (shared drive requirement): The input files (project, etc.) must be present on a shared drive that is accessible from every node of the cluster.

Note: this item is actually listed as a requirement.

Related Topics

Large Scale DSO for Parametric Analysis

Troubleshooting for Large Scale DSO

(LINUX only) Deployment/Installation errors (such as mainsoft related or related to deployment configuration) are not captured. If there is such an issue, Large Scale DSO job will fail without useful messages in the logs

Report-based extraction fails mysteriously if traces and parametric-setup are not 'prepared' as per the Getting Started guides.

Job monitoring and control

Job Abort: When a Large Scale DSO job is aborted, all results are lost (this will be fixed in R14 patch)

Job re-start: There is no provision for stopping and re-starting a job. A new job does not reuse solved results - always solves all rows in the table. So an abort or failure of a job re-starts from the beginning, unless a new parametric table with the unsolved rows is created

Job outcome

Job status: The exit code of job doesn't indicate success or failure correctly. The error messages from multiple log files needs to be combined to determine the reason for failure. In many situations, the reason for a failure is apparent only after re-running the job after turning ON the 'debug logging'

In some LINUX scenarios, the analysis appears to finish successfully with valid results, except that the exit code is '134'. In this case, although the exit is abnormal, the failed exit code can be ignored

Load Balancing: For models with 'unbalanced variations table' (i.e. variations that take considerably different amount of time to solve are clustered in few regions of table), job will take longer time to solve than a Regular DSO as the job's overall completion time is determined by the slowest solving region.

Workaround: rearrange the rows in the parametric table so that each region takes a similar time to solve

GM Specifics: the model used for 'Report-based extractor' jobs is NOT compatible with the 'ANSYS-extractor-for-GM' jobs. A valid model for ANSYS-extractor-for-GM cannot contain any of: reports, overlay plots, Optimetrics calculations.

Related Topics

Large Scale DSO for Parametric Analysis

Monitoring the Solution Process

While a simulation is running, you can monitor the solution's progress in the **Progress** window. Above the red progress bar, messages describe the setup and step. The progress bar shows the relative progress of each step. Under the bar, messages note the part of the design being solved, and give memory estimates during the factoring process.

You can also view the following solution data at any time during or after the solution:

- The convergence data:
- The matrices computed for the S-parameters, impedances, and propagation constants.
- A profile of status of the adaptive analysis, including the number of valid passes completed.

To view the **Solutions** window:

1. Right-click the solution **Setup** in the project tree.
   A shortcut menu appears.
2. Select **Convergence**, **Matrix Data** or **Profile** from the shortcut menu.

   The **Solutions** window appears with the corresponding tab selected and the current data displayed.

For "out of core" problems, quite different amounts of memory may be used for factorization and for solution. So if the amount for factorization is displayed under the progress bar and the amount used is calculated for the profile at the end of the solution, they may be quite different numbers.

To view the status of the adaptive analysis:

- Click **HFSS>Results>Browse Solutions**.
  The **Solutions** dialog box appears with the **Browse** tab selected. It displays data about the number of valid passes completed. It contains a tree structure showing the solutions listed according to Setup, Solution, and Variation. A table lists the Setup, the solution, the sweep variable, and the state of the solution.
- You can use the **Properties** button to display a dialog that lets you change the way the Setup, Solution, and Variation are listed in the tree structure of the **Solutions** dialog.
- The **Statistics** tab of the **Solutions** dialog displays path information, as well as format, number of files, and size.
- You can delete one or more solutions by selecting from the table and clicking **Delete**. Click on a solution to select it, and use Ctrl-click to select multiple solutions, or Shift-click to select a range of solutions. You can also select all solutions using the **Select All** button.

Note If HFSS loses its license, it waits for the license to be regained, checking every 2 minutes or until you abort.

Related Topics

Aborting Analysis

Deleting Solution Data

Post Processing and Generating Reports

High Performance Computing (HPC) Integration

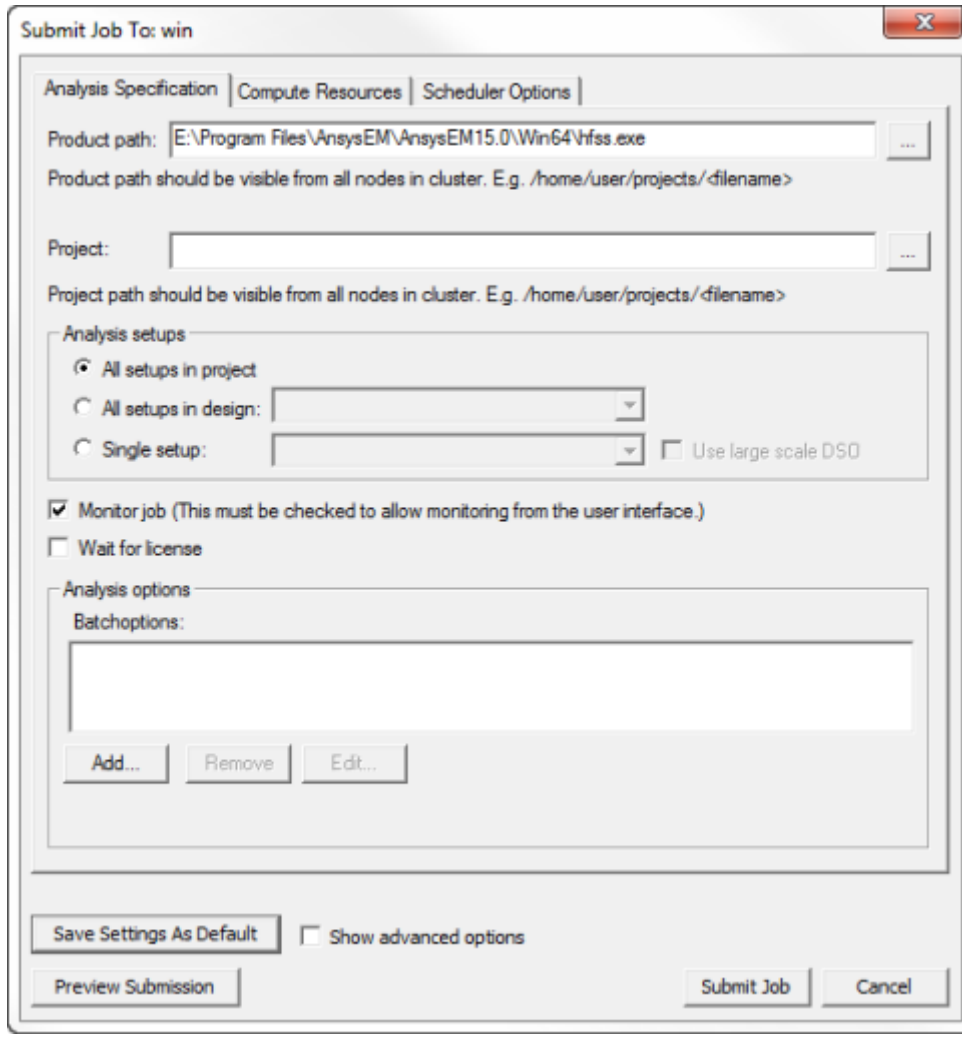
ANSYS Electromagnetics products offer a direct integration with a number of High Performance Computing (HPC) software programs. This direct integration does not require RSM Service. The list of currently-supported HPC software includes:

- Platform's Load Sharing Facility (LSF)
- Microsoft Windows® HPC Server 2008 R2 and HPC Server 2012
- PBS Professional from Altair Engineering
- Grid Engine (GE)

You can also do custom integration.

A job scheduler may also be described as a batch system, a Distributed Resource Management System (DRMS) or Distributed Resource Manager (DRM). The features supported on each scheduler are included in the documents for each. For each job scheduler, the versions or revisions that have been tested are included.

A user may submit jobs using the command line tools or other tools provided by the scheduler. The Desktop includes a GUI to help the user submit jobs to a job scheduler. This generic Job Submission GUI is shared across the ANSYS EM products.



The general procedure is to specify the scheduler and head node, describe and submit the job, and monitor the results.

The **Submit Job To** dialog contains three tabs:

- **Analysis Specification**--specify the Product path, Project name, the setups, and analysis options such as batchoptions, or, for advanced users, Environment variables. The project file pathname must be a UNC path that is accessible from each compute host used for ANSYS Electromagnetics jobs. The submission user must have permission to write to the directory containing the project file.

- **Compute Resources**--specify either a predefined Analysis Configuration (see Configuring Distributed Analysis) or specify parameters in the fields resource selection, for job parallelization and enabled forms of parallelization.
- **Scheduler Options**--specify for Job name and priority.

There are two ways that the GUI may be used to submit jobs. The first mode requires that the Desktop (UI) process run on a host which is also a submission host for the job scheduler. This mode is called local mode or working mode. The second mode is useful for cases in which the submission hosts are not able to run graphical processes, such as the Desktop.

The second mode is only supported on Linux in the ANSYS Electromagnetics Suite 15.0 release. In the second mode, an administrator configures the RSM Service to act as an interface to the job scheduler, and starts the RSM Service on a submission host for the cluster. The user runs the Desktop (UI) process on another host (which may be called the postprocessing host). To submit a job, the user specifies the host where the RSM Service is running, and the Desktop process connects to the RSM Service over the network to submit the job. In this mode, some configuration is required, and the RSM Service typically must run as a privileged user (e.g., root), so that it can launch processes as any user.

Related Topics

Scheduler Terminology

What a Scheduler Does

Installation of ANSYS Electromagnetics Tools

ANSYS Electromagnetics Jobs

Submitting and Monitoring ANSYS EM HPC Jobs

Large Scale DSO for Parametric Analysis

Scheduler Terminology

- Core: unit of processing
- Processor: consists of one or more cores
- Machine/Host/Node: consists of one or more processors, memory, disk, etc.
- Resource: Machines, licenses, etc. that are used by a Job
- Job: Application (also called: program, executable), with command line options, that uses resources to produce useful results. For example, hfss.exe -ng -BatchSolve.
- Serial Job: job that runs on a single core
- Parallel Job: job that runs on multiple cores (belonging to same or different machines)
- Compute Cluster: network of machines on which Jobs run. Typically, consists of head node(s) and many compute nodes
- Service: Program that runs in the background (e.g. RSM Service). 'Listens' on a 'port'. OS provides programming interface by which Applications communicate with service, once machine and port number are known. Launching an executable on remote machine, requires a service to

run on remote machine.

ANSYS Electronics Suite Terminology

- Desktop: The main application used to accomplish a task, such as hfss. The desktop may run as a GUI or it may run as a batch command.
- Engine: Application (aka: executable) that is launched during analysis commands, to generate analysis results
- Multi-processing: A single engine uses multiple cores on the **same** machine
- Distributed-processing: Multiple engines are launched simultaneously (on same machine **or** different machines). Uses ansoft_distrib (and related) license.

Related Topic

What a Scheduler Does

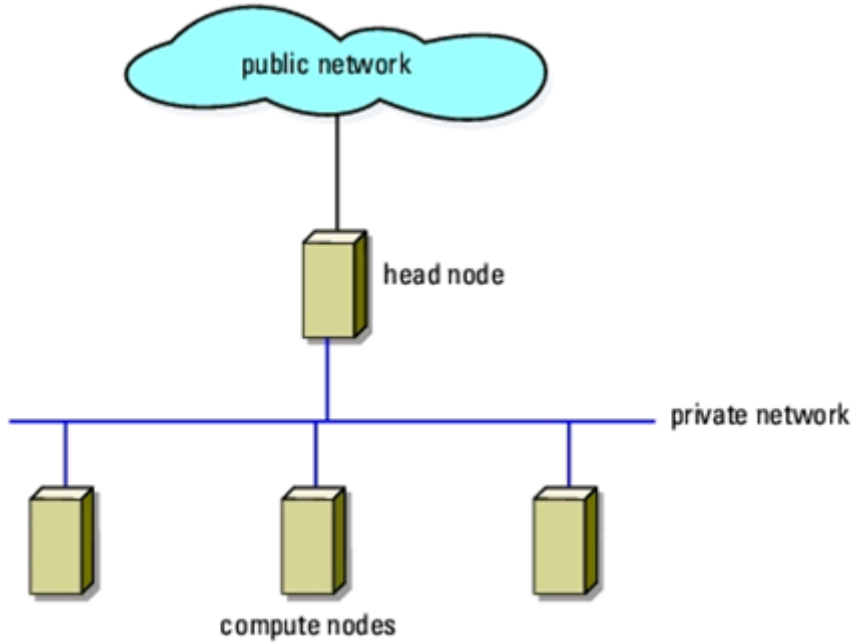
Command Line Enhancements for ANSYS Electronics Suite Desktop Products

High Performance Computing (HPC) Integration

What a Scheduler Does

- Enables effective/efficient utilization of cluster's resources consistent with organization's goals
- Maintains queue(s) of jobs
- Maximizes throughput of the jobs by processing all jobs as fast as possible
- Typically, one job per cpu policy
- Allows choice of various scheduling policies (e.g. First Come First Serve, Priority Based, Pre-emption)
- Provides a suite of tools or utilities (graphical or command line) for end user to submit jobs, monitor jobs, abort jobs, suspend jobs, …
- Manages a compute cluster by running various interacting 'services' on head nodes and compute nodes
- Provides a programming interface to access 'services'

Scheduler Managed Compute Cluster



Head node(s) typically maintains queues. Compute nodes are typically on a high speed network, to improve scalability of parallel jobs. Services running on nodes interact with each other to manage resources. End user tools communicate with services to submit/abort/suspend/etc. jobs.

Related Topic

High Performance Computing (HPC) Integration

Command Line Enhancements for ANSYS EM Desktop Products

Installation of ANSYS Electromagnetics Suite

ANSYS Electromagnetics Suite must be available on each cluster host where jobs may be run.

- On LINUX platform, ANSYS Electromagnetics Suite may be installed on a shared drive, that is accessible to all machines in the cluster.
- On Windows platform, ANSYS Electromagnetics Suite must be installed separately on each host of the cluster.

The ANSYS Electromagnetics Suite must be accessible using the same path on each host. All cluster users running ANSYS Electromagnetics jobs must have permission to read and execute the files in the installation directory and its subdirectories.

The Temp directory selected during installation must be readable and writable by all user accounts used to run the ANSYS Electromagnetics Suite. This temp directory path should be the same on all

machines of the cluster and should be local to every machine. For example, c:\temp on Windows, /tmp on LINUX

Because HPC is offered as a direct integration, you need only install the ANSYS Electromagnetic Suite software. No additional configuration is required.

Example

Install the ANSYS Electromagnetics Suite in directory C:\Program Files\AnsysEM\ on each node of the cluster. The same directory pathname must be used on all hosts.

Related Topics

High Performance Computing (HPC) Integration

Firewall Configuration

Installation Directory Examples

Firewall Configuration

If firewall is turned OFF between the machines of the cluster, there is no need for any configuration. If firewall is turned ON, you, or a system administrator, should perform the steps below.

- Windows cluster: Configure firewall by adding exceptions that allow ANSYS Electromagnetics Suite programs and services to communicate with each other. If you are using standard Windows Firewall, this is automatically done for you, by the Ansoft installation program. On the other hand, if you are using a 3rd-party firewall software, it needs to be configured in a similar manner.
- LINUX cluster: Open up the firewall for range of ports denoting ephemeral (or dynamic) ports. Check with your system administrator on how this can be done on each machine of cluster.

Related Topics

High Performance Computing (HPC) Integration

Installation Directory Examples

Installation Directory Examples

Microsoft Windows Example

Install the ANSYS Electromagnetics Suite in directory C:\Program Files\AnsysEM\AnsysEM15.0\win64 on each node of the cluster. The same directory pathname must be used on all hosts.

LINUX Example

Install the ANSYS Electromagnetics Suite in a common directory that is accessible using the path /opt/ansoft/hfss14 on each execution node of the cluster.

Related Topics

High Performance Computing (HPC) Integration

Firewall Configuration

ANSYS Electromagnetics Jobs

ANSYS EM Jobs

For most cluster environments, an ANSYS EM job will consist of an ANSYS EM Desktop running in non-graphical mode, performing a batch solve. The user will submit the job to the scheduler, specifying an ANSYS EM Desktop command line to be executed on the cluster. For some schedulers, the user may or must specify a script to run instead of specifying the ANSYS EM Desktop command line; in these cases, the script will contain the corresponding ANSYS Electromagnetics desktop command line. When the resources requested for the job are available to the job, the scheduler will start the job. In many cases, the user submitting the job will not know which host or hosts are allocated to the job. With direct integration, if the ANSYS EM job is a distributed job, the ANSYS Desktop will query the scheduler for the hosts allocated to the job, and it will use the scheduler facilities to launch the distributed engines.

Related Topics

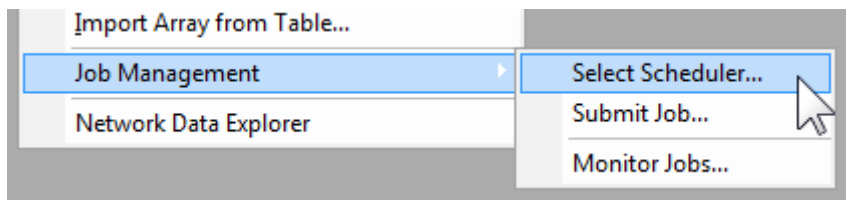
High Performance Computing (HPC) Integration

Running HFSS from a Command Line

Integration with Microsoft Windows® HPC Scheduler

The Windows HPC scheduler is only supported on Windows. Jobs may be submitted in any of the following ways:

- Using the Windows HPC GUIs from Microsoft: Job Manager or Cluster Manager
- Using the Windows HPC command line tools (job, etc.)
- Using the Desktop UI commands for Scheduler selection, Job submission and Job monitoring/control. You specify the Windows HPC Scheduler User Interface for Submit Job by clicking **Tools>Job Management>Select Scheduler...**



Once you select a scheduler, you can access the interface for job submission, monitoring and control.

The ANSYS Electromagnetics Suite 15.0 release has been tested with the following versions of Windows HPC:

- Windows HPC Server 2008 R2
- Windows HPC Server 2012

General Guidelines for Submitting ANSYS EM Jobs

A Job submitted to Windows HPC Cluster is defined by Job properties, Task List and Task properties. Priority, resource requirements, node preferences, etc. come from Job properties. In the case of ANSYS Electromagnetics jobs, Task List consists of a single task. Properties of this task specify

the command line that runs ANSYS Electromagnetics desktop in non-graphical mode to perform analysis of a project.

Specifying the Number of Compute Resource Units for HPC Jobs

In the old setup you had to list machines multiple times, and figure out how many cores per task on each machine in order to set Number of Processors Distributed to the smallest of these. In the new setup, you just enter the number of tasks and total cores per machine.

ANSYS EM Project File and Project Directory for use with Windows HPC Scheduler

ANSYS Electromagnetics Suite 15.0 tools write their results to a subdirectory of the directory containing the ANSYS EM project file. The Project Directory (the directory containing the project file) must be accessible to all of the cluster hosts that may run ANSYS EM jobs. The user account for the job must have permission to read the project directory, and to create and modify files and subdirectories of this directory. The pathname of the project file must be accessible to all cluster hosts using the same path name, which is generally expressed as a UNC pathname.

Example:

The project file is on the user's workstation (with hostname user1_PC) in directory C:\user1\projects\new\project1.hfss, and the directory C:\user1\projects is shared with sharename projects.

Correct

When submitting the job, you should use the following pathname to specify the project file:

```
\\user1_PC\projects\new\project1.hfss
```

Incorrect

If a local pathname is used, the cluster hosts will not be able to find the user's project on the workstation

```
user1_PC: ' C:\user1\projects\new\project1.hfss '
```

Related Topics

Windows® HPC Job Templates

Selecting Computation Resource Units (Job Unit Type)

Windows® HPC Job Credentials

Submitting and Monitoring ANSYS EM HPC Jobs

Jobs may be submitted to the Windows HPC Scheduler using any of the following methods:

- Using the **Submit HPC Job** dialog
- Using the Windows HPC Job Manager GUI
- Using the Windows HPC Command Line Tools
- Using the Windows PowerShell

Client Utilities from the Microsoft HPC Pack, must be installed on the submit host to use any of these methods to submit a job to a cluster. The **Submit HPC Job** dialog will be unable to contact the cluster head node if the client utilities are not installed.

This document covers the first method. See the Microsoft documentation for information on the other three methods.

- Submitting and Monitoring Jobs for Windows HPC
- Specifying the Number of Compute Resource Units for HPC Jobs

Jobs may be submitted from any Microsoft Windows host meeting the following requirements:

- For submitting jobs to the Windows HPC scheduler, the Desktop process must run on a node that is configured for submission of jobs to the Windows HPC cluster. That is, the Windows HPC Client Utilities must be installed on the node, and network communication from the Desktop node to the head node of the cluster must be allowed. For ANSYS Electromagnetics Suite 15.0, Windows HPC Server 2008 R2 (or later) client utilities are required. Using a computer on the network is not supported for submission of jobs to the Windows HPC cluster.
- When submitting jobs to a Windows HPC cluster, the user must also specify the head node of the cluster to which the jobs will be submitted. When the user selects the "Windows HPC" scheduler in the "Choose scheduler" list, the Head Node edit control is enabled. The user may enter the Windows HPC cluster head node name into the edit box. Alternatively, the head node may be selected using a "Browse for Computer" browser by pressing the ellipsis [...] button.
- The Windows HPC Pack client utilities are installed on the submission host
- Network communication between the submission host and the Windows HPC Cluster head node is permitted; there is a network connection between these hosts that is not blocked by any firewall or the like
- The submission user is permitted to submit jobs to the Windows HPC Cluster

Job Monitoring

Windows HPC Jobs may be monitored using the **Monitor Job** dialog which is brought up by the **Tools > Job Management > Monitor Jobs...** command. This dialog may also be brought up by checking the **Begin monitoring this job now** checkbox when a job is successfully submitting using the job submission dialog. In addition to the above requirements to allow job monitoring the following is also necessary:

- Network communication between the submission host and all Windows HPC Cluster nodes where the job may run is permitted; there is a network connection between these hosts that is not blocked by any firewall or the like

Cluster Configuration

Any job running on a Windows HPC Cluster that is distributed over multiple compute hosts requires network communication between processes running on these hosts. The cluster must be configured to allow this communication. Any firewall or other security software must be disabled or configured to allow communication between any of the compute hosts were a job could run.

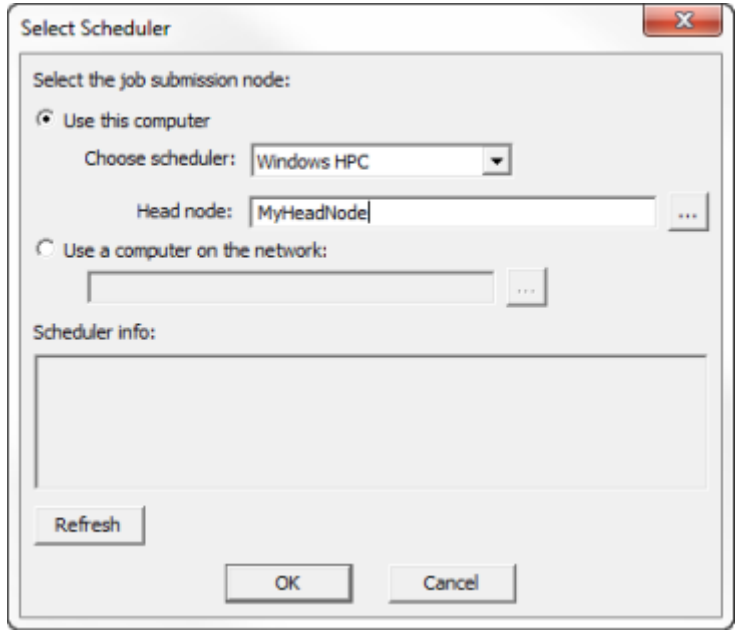
Job Submission User Profile on Cluster Compute Nodes

In order for a job to run correctly, the submission user's profile must be accessible and properly initialized on the cluster compute nodes where the job runs. If the Ansoft/temp subdirectory of the user's "My Documents" directory does not exist or is not accessible on the compute cluster nodes

where a job runs, the batchoptions for the job will not be processed correctly, resulting in job failure. One way to ensure that this directory is created on each compute host is for the submission user to login to each compute host and run the product GUI one time.

Submitting and Monitoring Jobs for Windows HPC

In order to submit jobs using the **Windows HPC**, you must click **Tools>Job Management>Select Scheduler** to open the **Select Scheduler** dialog and specify Windows HPC as the Scheduler.



For Windows HPC, the user must select the "Use this computer" radio button and then specify the head node of the cluster.

After specifying the job submission node, you can click **Refresh**. This verifies that the head node may be contacted, and displays the scheduler name, a brief description (including the head node name), and the version of the Windows HPC head node.

Pressing **Cancel** discards changes made in this dialog. Pressing **OK** verifies that the head node can be contacted before accepting the changes. If no problem occurs, the dialog will be closed. If there is a problem contacting the head node, the dialog will not be closed and the changes are not accepted.

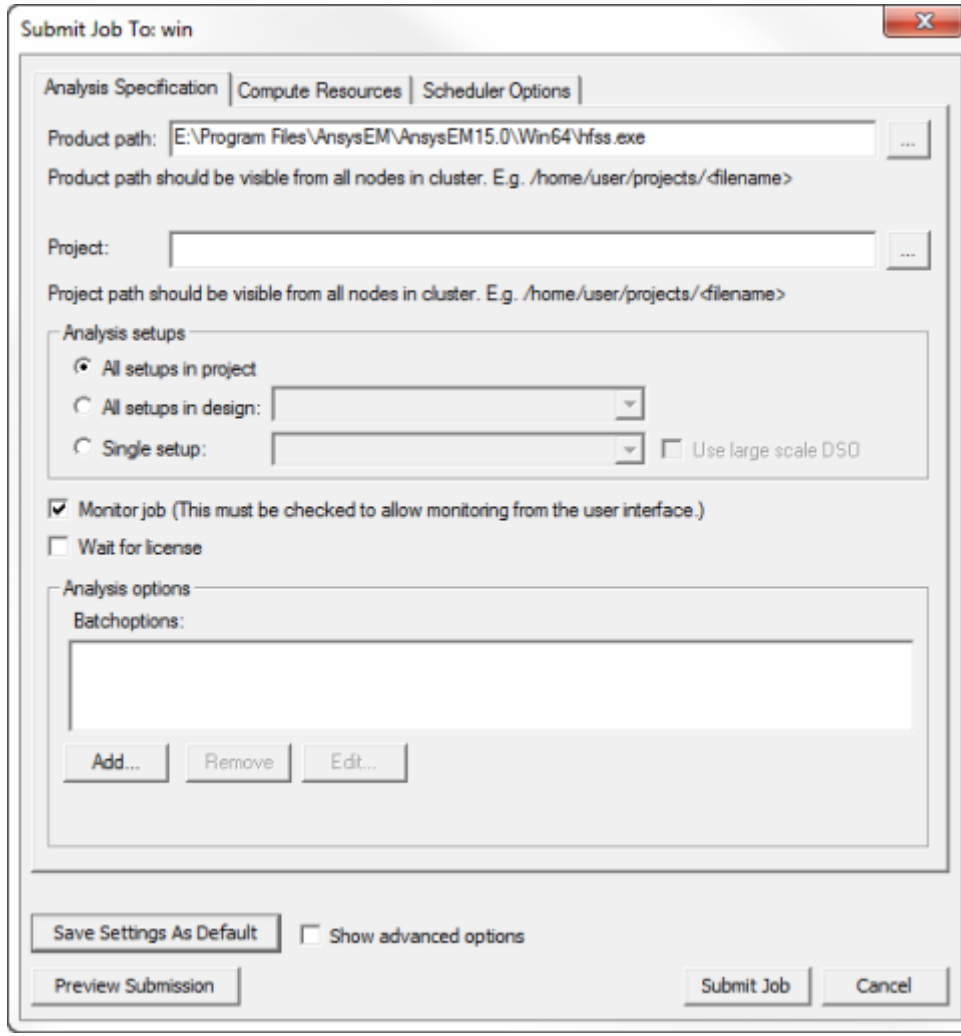
After setting the job submission node, select **Tools>Job Management>Submit Job...** to open the **Submit Job To:** dialog. This contains three tabs:

- **Analysis Specification**--specify the Product path, Project name, the setups, and analysis options such as batchoptions, or, for advanced users, Environment variables.
- **Compute Resources**--this tab can be populated either by predefined Analysis Configuration,

or specifying parameters in the fields resource selection, for job parallelization and enabled forms of parallelization.

- **Scheduler Options**--contains fields for Job name and priority. The customization options shown by checking advanced are not used for Windows HPC.

In the **Analysis Specification** tab, enter the pathnames of the product path and of the project file in the "Project" edit box. These must be UNC paths that are accessible from each compute host used for ANSYS Electromagnetics jobs. The submission user must have permission to write to the directory containing the project file.



You can select which setups are analyzed in the Analyze Setups section of this dialog. There are radio buttons to select:

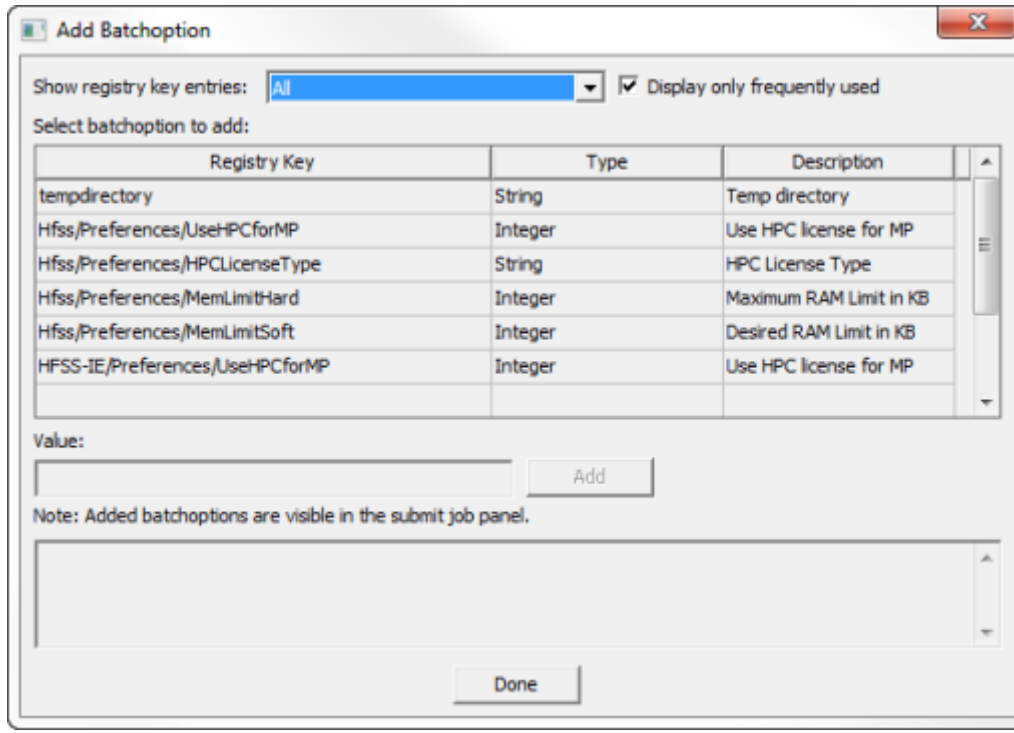
- All setups in the project
- All setups in a specified design: you selects the design from the dropdown list
- Single setup:

If you specify multiple setups, they will be processed sequentially in the order displayed in the edit box.

The Analysis options include:

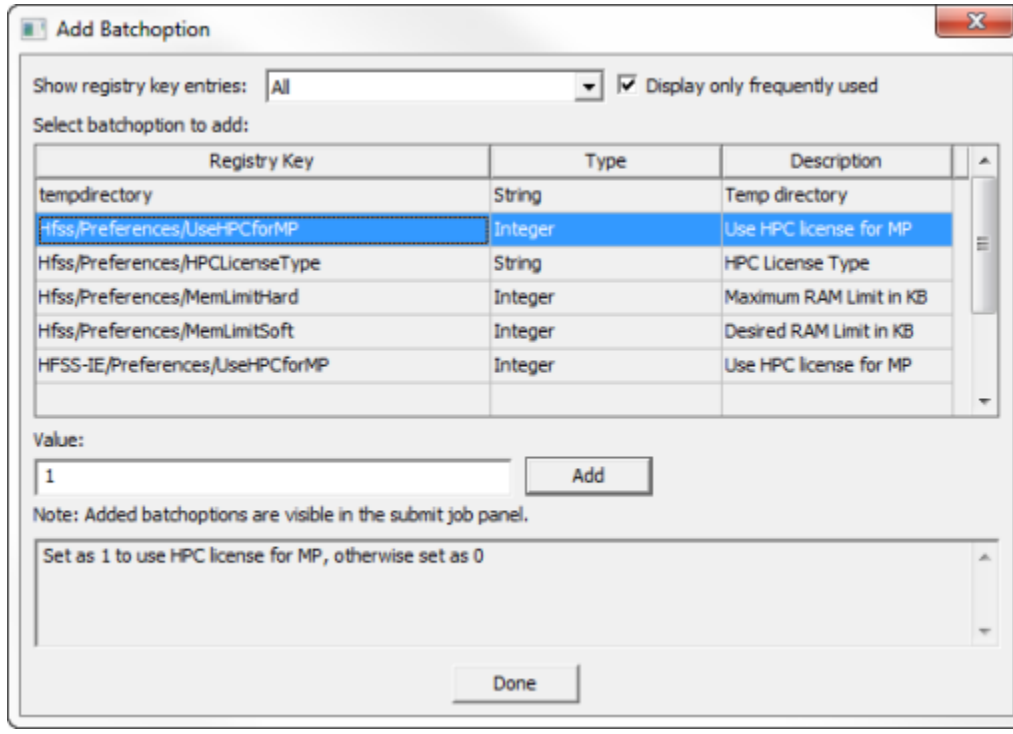
- Monitor job. You must enable this option to monitor the job from the user interface.
- Wait for license- whether to wait until a license if available before starting a simulation.
- Batch options. You can optionally specify -Batchoptions in the text field. See detailed discussion of -Batch Options beginning under Running HFSS from a Command Line.

The **Add...** button opens the **Add Batchoption** dialog.

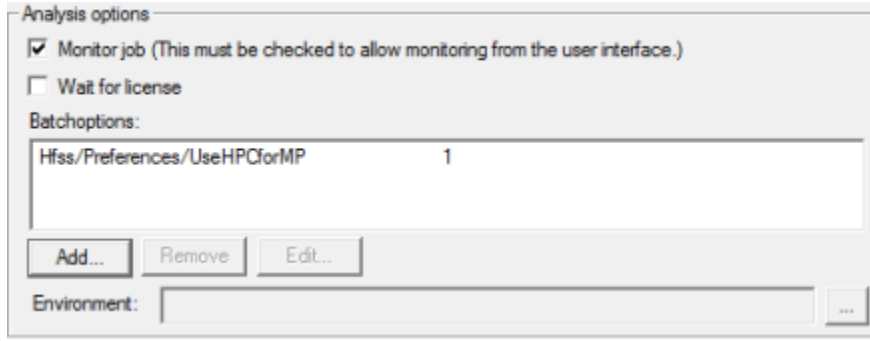


This dialog provides access to all -Batchoption commands. The drop down menu lets you select specific categories, and you can choose to display only frequently used commands. You can edit and remove any batch options you specify.

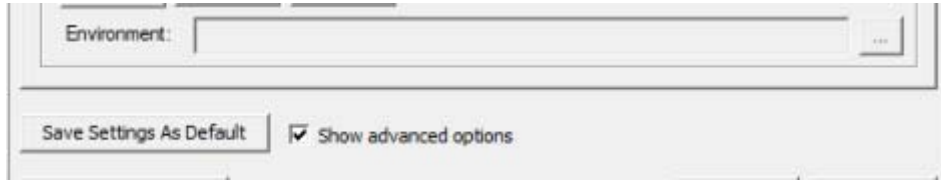
Select a Registry Key. in order to show the current Value for the type. The lower field explains the meaning of the Type Value.



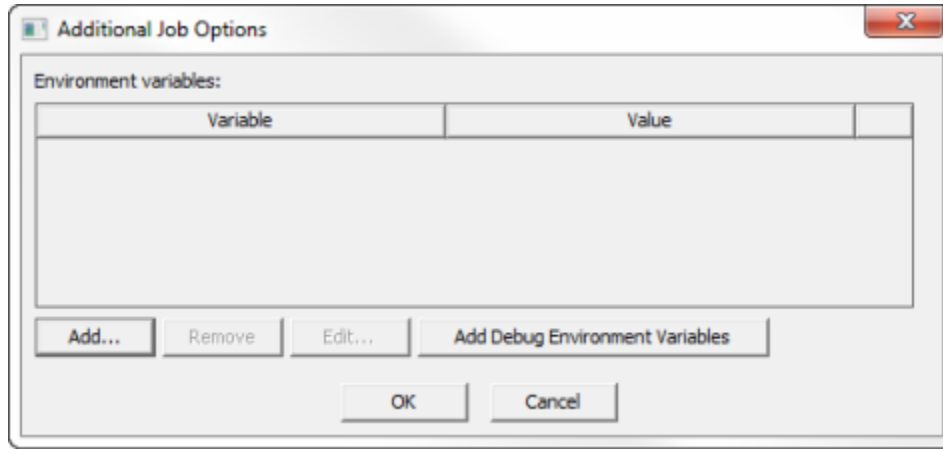
Any batchoptions for which you select **Add** will be visible in the **Submit Job** dialog.



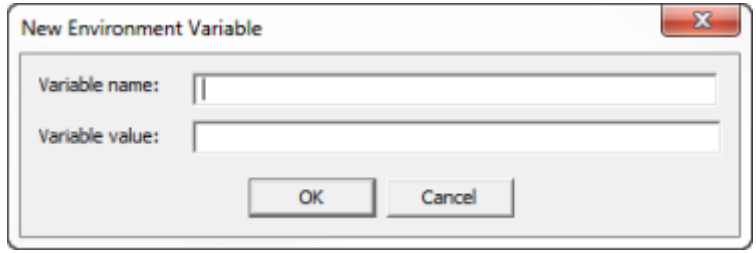
If you have the Show advanced options box checked in the Submit Job dialog, the Environment field displays.



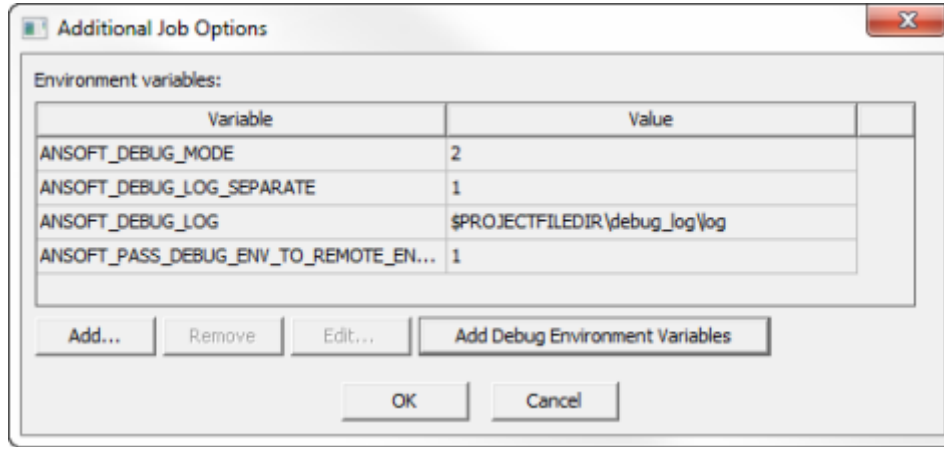
The Environment field lets you specify any Environment variables. Click the ellipsis button [...] to display the **Additional Job Options** dialog.



Click the **Add...** button to open the **New Environment Variable** dialog.



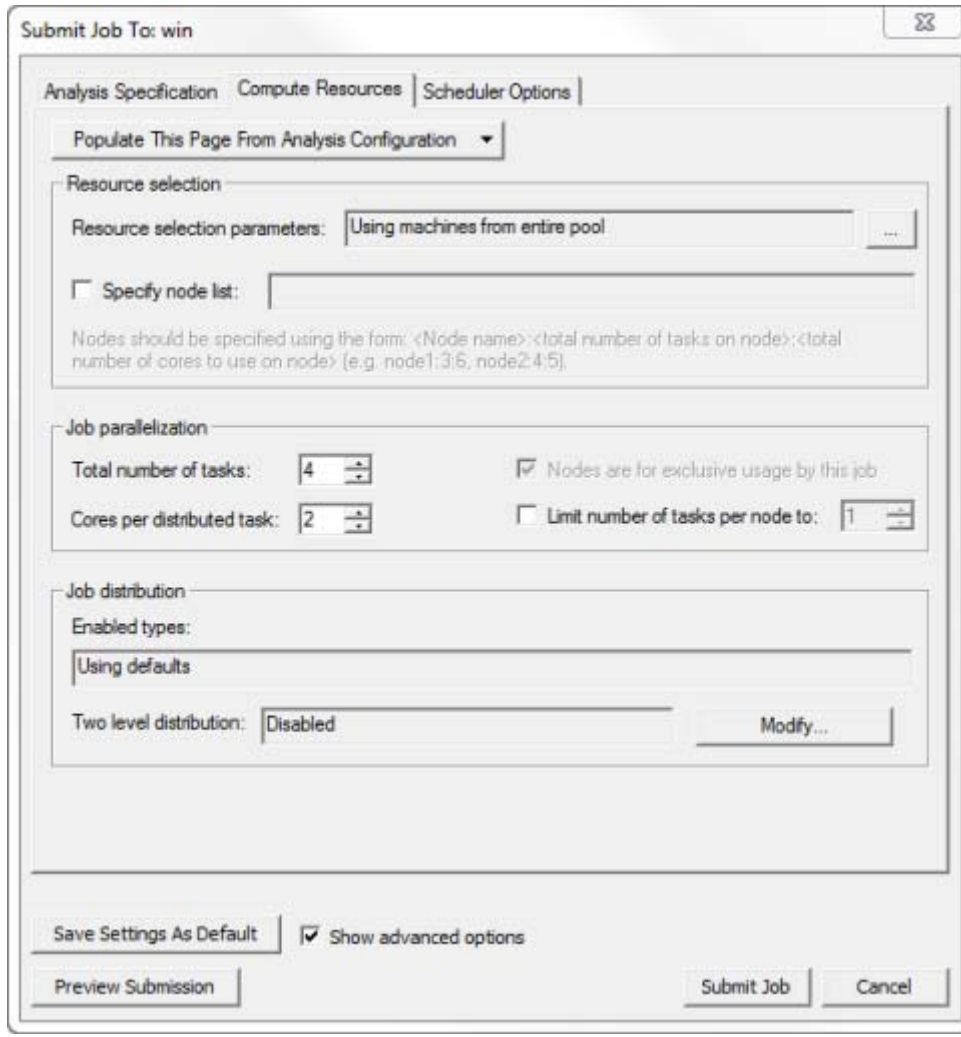
Here you can provide a Variable name and Variable value. Click OK to display the Variable in the Additional Job Options dialog. Select a Variable to enables the **Remove** and **Edit...** buttons. You can also click **Add Debug Environment Variables**.



Any Variables that you add will be displayed in the Environment field of the **Submit Job** dialog, if you have also enabled Show Advanced options.

The **Preview Submission** button opens a window that shows the text commands that will be sent to the scheduler.

The following figure shows the **Compute Resources** tab of the **Submit Job To** dialog.



The **Populate This Page From Analysis Configuration** drop down menu lists all Analysis configurations that you have previously defined. See Editing Distributed Machine Configurations.

Selecting an item from this menu will populate the node list and the controls in the "Job distribution" group box. You must enter the other settings, including the "Resource selection parameters" and the controls in the "Job parallelization" group box.

You can also populate this page yourself by specifying Resource selection, Job Parallelization parameters, and Job distribution parameters.

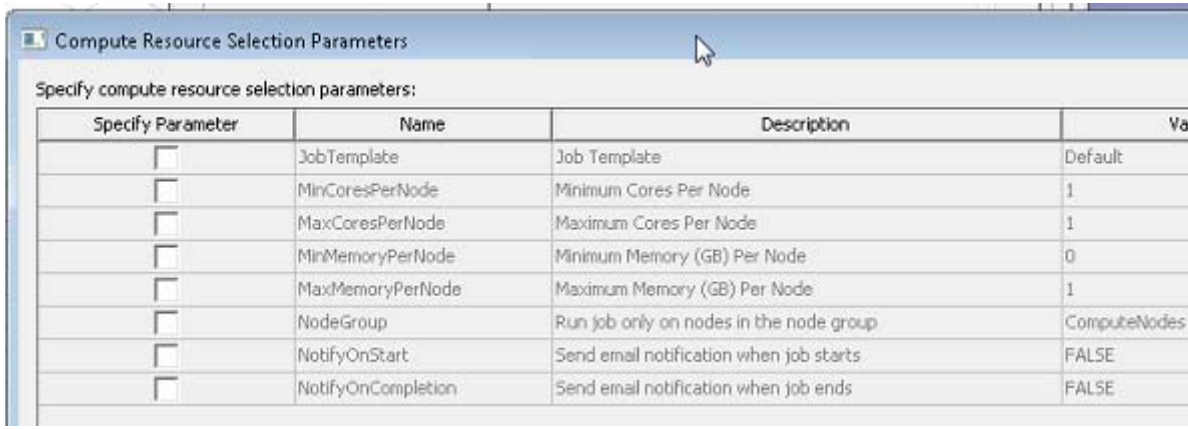
Node List

For Windows HPC jobs, you may either specify a node list, or specify the job parallelization parameters, but not both.

If the node list checkbox is checked, then you may specify a node list, and the Job parallelization controls are disabled, but the Total number of tasks shows the total number of tasks specified by the node list. In this case, the node list should only include cluster nodes that are valid for the job. If the specify node list checkbox is not checked, then the node list is ignored, and you specify the total number of tasks, the cores per distributed task and, optionally, the limit on the number of tasks per node. All Windows HPC jobs are submitted exclusively, so the exclusive checkbox is disabled.

To use a the node list, check the checkbox, and enter the node list in the associated edit control. For each node, you enter the node name, the total number of tasks for the node, and the total number of cores for the node, with the three fields separated by colons. The node names should be cluster nodes that meet all of the criteria specified by the compute resource selection parameters. The entries must be separated by a commas. Different numbers of cores or tasks may be specified for different nodes; this feature is useful for clusters that contain heterogeneous cluster nodes. When using a node list, the settings of the "Total number of tasks", "Cores per distributed task", and "Limit number of tasks per node" controls are not used for job submission, but the "Total number of tasks" control displays the total number of tasks specified by the node list, if the node list is valid. For this case the job unit type is Node, and it runs only on the specified nodes.

By default, you can draw from the entire pool. You can also click the ellipsis button [...] to open a **Compute Resource Selection** dialog.



The resource selection parameters for Windows HPC jobs are:

- JobTemplate: Job Template - The JobTemplate may limit the job parameters or specify defaults values for job parameters
- MinCoresPerNode: Minimum Cores Per Node
- MaxCoresPerNode: Maximum Cores Per Node
- MinMemoryPerNode: Minimum Memory (GB) Per Node
- MaxMemoryPerNode: Minimum Memory (GB) Per Node
- NodeGroup: Run job only on nodes in the node group

- NotifyOnStart: If True, send email notification when job starts. Email notifications must be configured and enabled for the cluster by the administrator. (The cluster head node must run Windows HPC Server 2008 or above.)
- NotifyOnCompletion: If True, send email notification when job ends. Email notifications must be configured and enabled for the cluster by the administrator. (The cluster head node must run Windows HPC Server 2008 or above.)

You can also use the text field to specify a node list.

Job Parallelization

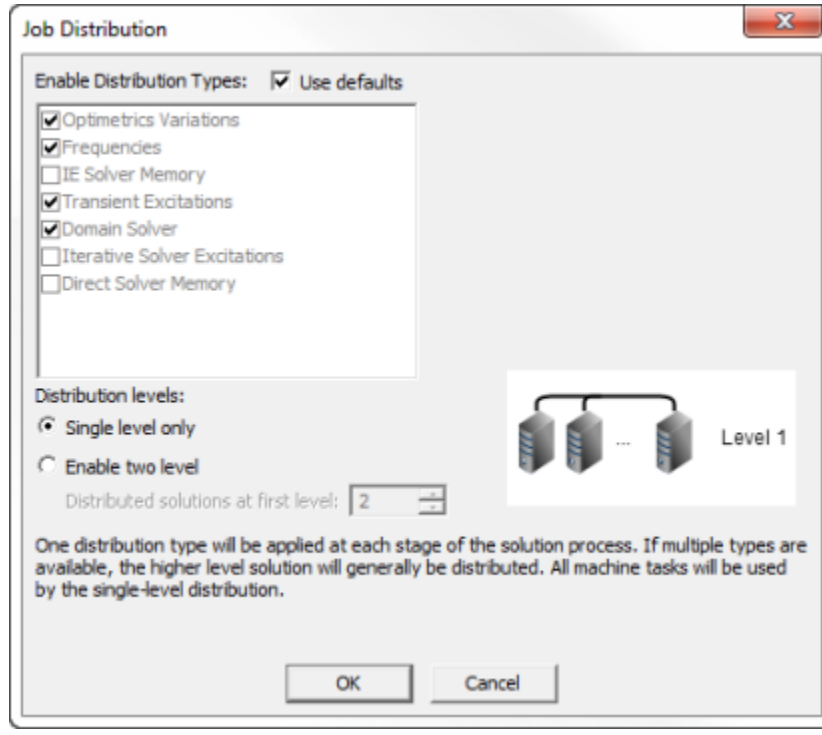
For Windows HPC jobs, you may either specify a node list, or specify the job parallelization parameters, but not both.The Job parallelization fields let you specify

- Total number of tasks: The number of nodes requested for the job is the total number of tasks divided by limit on the number of tasks per node, rounded up if it is not an integer.
- Cores per distributed task. This determines the amount of multiprocessing per task.
- Whether nodes are for exclusive usage by this job
- Whether to limit the number of tasks per node to a value. If the "Limit number of tasks per node" checkbox is not checked, then the job is submitted with a job unit type of "Core".

Job Distribution

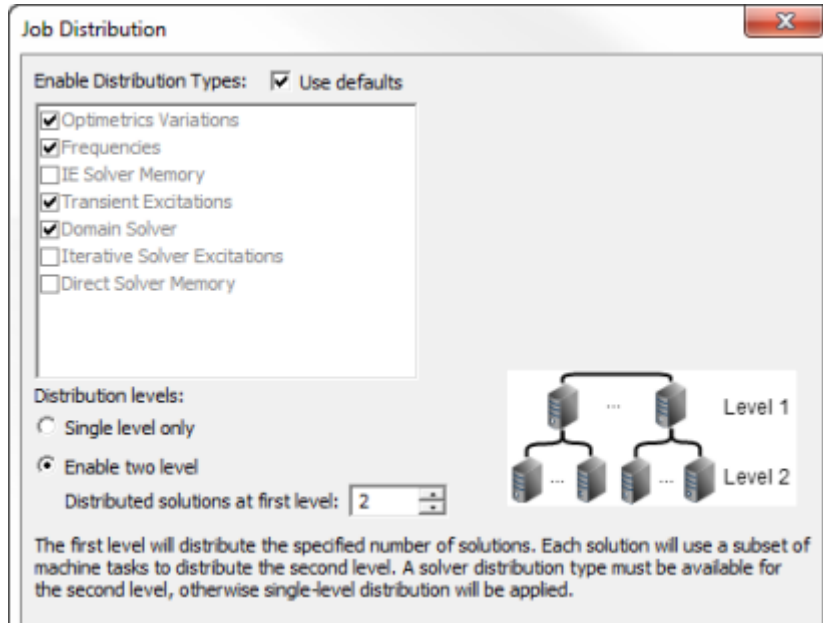
- Enabled types, such as Variations, Frequencies, Transient Excitations, Solver Domains, Direct Solver and Iterative Solver.
- Two level distribution, which may be disabled. Click the **Modify** button to display the **Job**

Distribution dialog.



Enabled Distribution types can modified here.

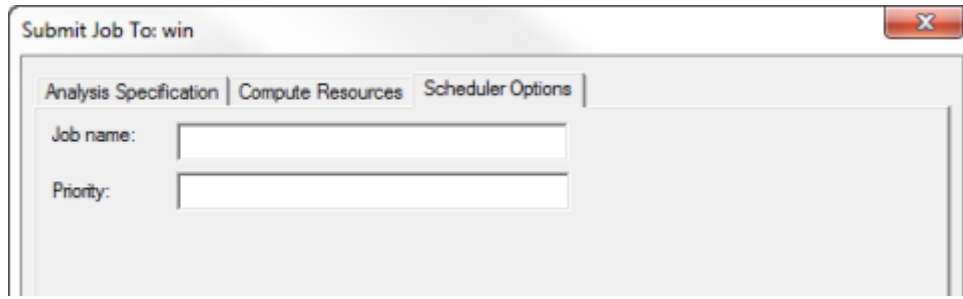
Second level distribution operates within DSO. If available and enabled you can specify a number of engines for level 1.



In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines, with two processors per engine, and also limit the number of engines per node to 4, the scheduler may increase the number of cores used in order to meet the limit specified for engines per node. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.

Scheduler Options

In the **Scheduler Options** tab provides for specifying the job name and/or the job priority. While the Show advanced options checkbox enables the display of Job submission options, no job submission options should be specified for Windows HPC.



Preview Submission

The Preview Submission button opens a window that shows a text description of the job to be submitted and the task used to start the product on one of the nodes.

The JOB PARAMETERS section contains information on parameter that apply to the job as a whole.

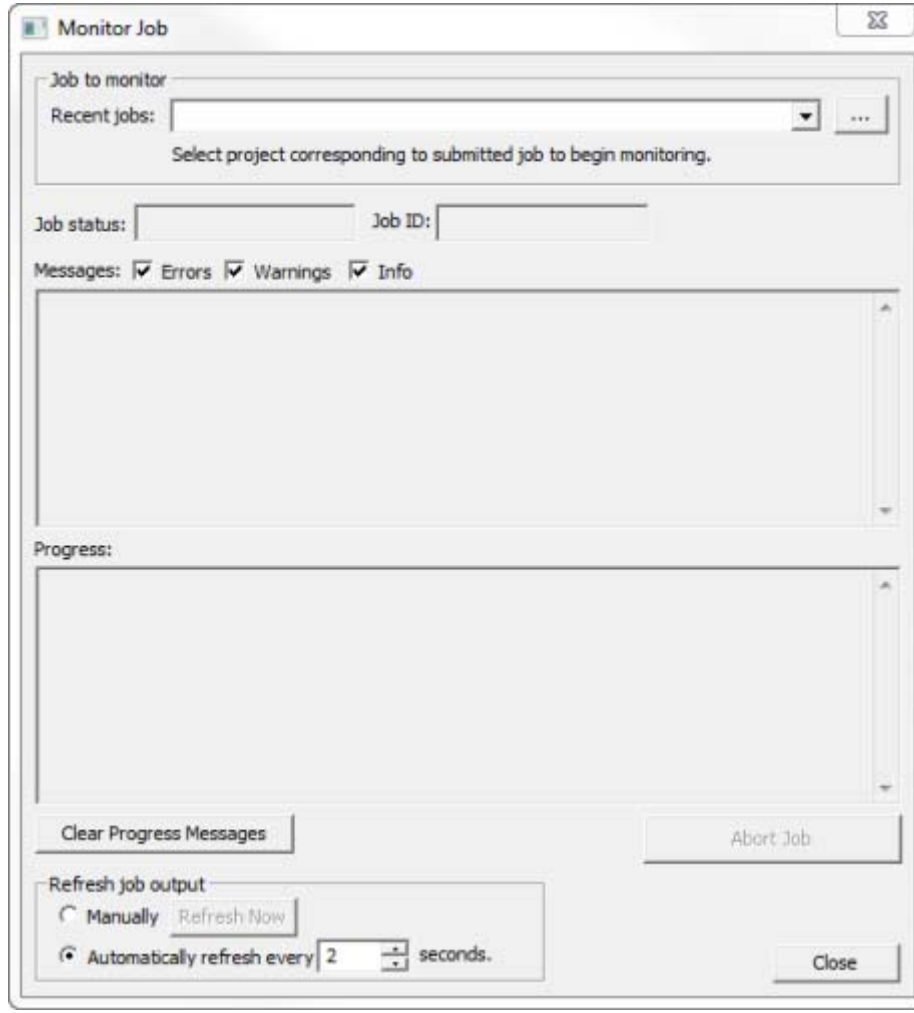
- The "Job resource parameters" section indicates whether the job has exclusive use of nodes, the job unit type, and the minimum and maximum number of units requested for the job, node group, and email notifications.
- The "Job attributes" section displays the job name and job priority.
- The "User Specified Compute Resource Attributes" displays the Resource selection settings.

The TASK PARAMETERS section contains information on parameters that apply to the Desktop task, which is the main task of the job.

- The "Desktop task resource parameters" section indicates the job unit type (which is the same as in the JOB PARAMETERS), and the minimum and maximum number of units requested for the Desktop task.
- The "Command Line section" displays the desktop task command line, including all arguments.
- The "Environment variables" section displays the environment variables that are set for the Desktop task; the same environment variables will also apply to all other tasks of the job.
- The "Working directory" section indicates the working directory in which the Desktop task will run.

Monitor Job

If you have checked the Monitor Job option on the **Submit Job To** dialog, **Analysis Specification** tab, you can invoke the **Monitor Job** window by clicking **Tools>Job Management>Monitor Jobs...** This dialog may also be brought up by checking the **Begin monitoring this job now** checkbox when a job is successfully submitting using the job submission dialog.



Related Topics

Submitting and Monitoring Ansoft HPC Jobs

Specifying the Number of Compute Resource Units for HPC Jobs

Windows® HPC Job Templates

The job templates are managed by the Windows HPC cluster administrator. Every cluster has at least one job template, the "Default" job template. Every job has an associated job template. If no job template is specified, then the "Default" job template is used. The job template controls two related aspects of the job submission process. When a job is submitted, there are a number of job parameters which may be specified. Each parameter has a set of valid values. For example, the Pri-

ority parameter has five valid values, Highest, AboveNormal, Normal, BelowNormal, and Lowest. The job template controls the default value of each parameter; this is the value that the parameter has if it is not specifically overridden by the submitter. For example, in the Default job template, the default value of the Priority parameter is Normal. The job template may also limit the allowed values of each parameter to a subset of the valid values. For example, a job template for privileged users could allow all five Priority values, which a job template for unprivileged users could limit the allowed Priority values to Normal, BelowNormal and Lowest.

Each job template is a Windows object with access controlled by an ACL (access control list). Instead of the usual "Read" or "Read & Execute" permissions, there is a "Submit Job" permission which corresponds to the right to submit a job with this job template. The cluster administrator may create job templates to limit or control access to cluster resources. For example, a job template with limited allowed job run times, or access to a limited set of compute nodes could be created by the cluster administrator. Specific users or user groups could be forced to use this limited job template by omitting access to the other job templates or by adding a deny access entry for the specified user or group to the other job templates. See the *Windows HPC Server 2008 Job Templates* white paper from Microsoft for additional details:

http://www.microsoft.com/en-us/download/confirmation.aspx?id=5659

Job templates may also be created to allow users to run jobs with limited knowledge of the appropriate job parameters. The cluster administrator creates a job template which has reasonable default values for the type of job to be run, and informs users which job template to use for each type of job. The template could also limit some parameters to only the subset of all values that are useful for the type of job associated with the template.

Related Topics

Integration With Microsoft Windows® HPC Scheduler

Windows® HPC Job Templates

Selecting Computation Resource Units (Job Unit Type)

Windows® HPC Job Credentials

High Performance Computing (HPC) Integration

Selecting Computation Resource Units (Job Unit Type)

The Job Unit Type is the smallest unit of processing resources used to schedule the job. This is one of the most important job properties. There are three options for the Job Unit Type: cores, nodes or sockets.

- **Cores**: Jobs are scheduled in units of cores, which may be also described as a CPU cores, logical processors, or CPUs. This is the smallest unit of granularity available. This selection allows the scheduler to start multiple tasks on a processor, if the total number of cores needed by the tasks is less than or equal to the number of cores on the processor. This selection may also allow the scheduler to distribute more of the computational load to processors with more cores than to processors with fewer cores.
- **Nodes**: Jobs are scheduled in units of nodes, hosts or machines. This is the coarsest level of

granularity that may be selected. When this option is selected, only one task will run on any give node at any given time. This is useful in cases where it is not desirable to run multiple tasks on a single host. For example, if each task is multi-threaded, running multiple tasks on the same node may not be needed to fully utilize the computing resources on the node. This may also be preferred if the tasks are memory intensive, and multiple tasks would be competing for the limited memory resources.

- **Sockets**: A socket (which may also be called a NUMA node) is a collection of cores sharing a direct connection to memory. A socket will contain at least one core, and it may contain several cores. The socket concept may not necessarily correspond to a physical socket. Scheduling at the socket level may be useful in cases in which each task requires extensive use of the memory bus, and scheduling multiple tasks on the same socket would result in excessive bus contention.

Related Topics

Integration With Microsoft Windows® HPC Scheduler

Windows® HPC Job Templates

Selecting Computation Resource Units (Job Unit Type)

Windows® HPC Job Credentials

Windows® HPC Job Credentials

Normally, a user will be prompted for the credentials used to submit a job. One way to simplify this process is to use the "cluscfg setcreds" command to set the user's credentials in the credentials cache. If this is done, then no password needs to be supplied for a job submitted for the specified user. Here is a cluscfg command that may be used to set the user credentials in the credentials cache:

```
cluscfg setcreds /password:* /scheduler:cluster_name
/user:domain\user_name
```

Here:

- cluster_name = the name of the cluster (hostname of the head node)
- domain = optional domain name; if omitted, the following \ should also be omitted
- user = user name

When this form of the command is used, the user is prompted for the password and also asked if the password should be remembered (cached).

See the following web page for more information on the cluscfg setcreds command:

http://technet.microsoft.com/en-us/library/cc947669(WS.10).aspx

Related Topics

Integration With Microsoft Windows® HPC Scheduler

Windows® HPC Job Templates

Selecting Computation Resource Units (Job Unit Type)

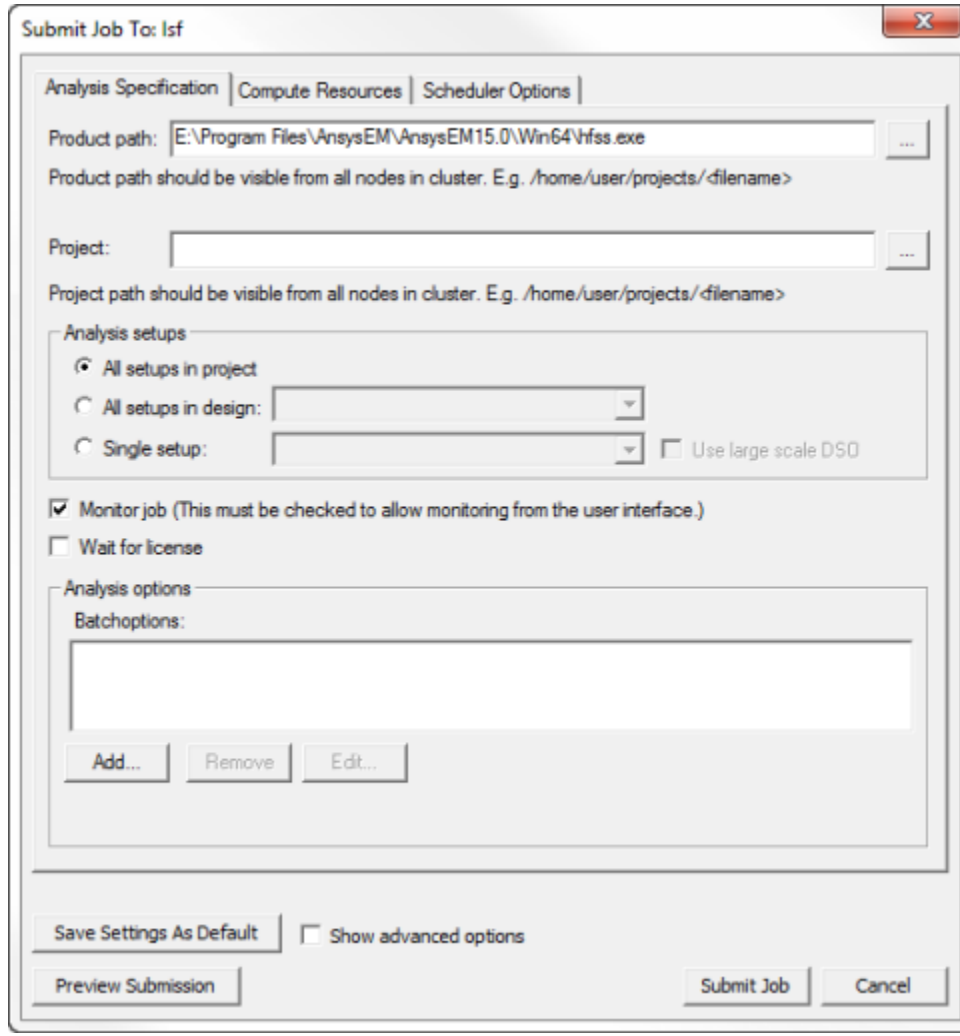
Windows® HPC Job Credentials

High Performance Computing (HPC) Integration

Integration with Platform's Load Sharing Facility (LSF)

ANSYS Electromagnetics products support Windows LSF cluster as well as LINUX LSF cluster. Serial analysis, Multi Processing and Distributed Analysis is done on compute resource units assigned by LSF. Models with parametric sweeps can use Large Scale DSO. With LSF, the ANSYS EM job doesn't require graphics. ANSYS EM job's progress can be monitored through LSF commands or through the dialog opened through **Tools>Job Management>Monitor Jobs...**. If a temp directory is setup by the LSF cluster administrator, analysis engines use this temp directory, overriding the setting in the ANSYS EM product.

Besides the command line interface, you can also use a Job Management user interface to submit jobs



Related Topic

General Terminology for LSF

Installation of ANSYS Electromagnetics Suite 15.0 on LSF Cluster

Integration of ANSYS EM Products with LSF

LSF Job Submission Guidelines

Job Management User Interface for LSF

LSF Command Used to Launch Remote Engine Processes

Installation of ANSYS EM Tools on LSF Cluster

The LSF scheduler is supported on both Linux and Windows.

On Linux, jobs may be submitted in any of the following ways:

- Using LSF commands (bsub, etc.)
- Using the generic scheduler GUI in local mode
- Using the generic scheduler GUI in service mode

On Windows, jobs may be submitted in any of the following ways:

- Using LSF commands (bsub, etc.)
- Using the generic scheduler GUI in local mode

The ANSYS Electromagnetics Suite 15.0 release has been tested with the following versions of LSF:

- LSF 7.0.5 on Windows
- LSF 7.0.6 on Linux

Windows:

Install on every node of cluster

Setup 'temp directory' to a path that is same on all nodes. For example, c:\temp

LINUX:

Install on a single node, on a shared drive.

Setup 'temp directory' to a path that is same on all nodes. For example, /tmp

Ensure that the product is available using the same path on all nodes

Permissions:

All users of the cluster should have read/write permissions to temp directory

All users should have read/execute permissions to installation directory

When a desktop scheduler GUI is run the same node as the job submission node, no other configuration is necessary: installation is sufficient. You select the scheduler through the desktop GUI. You need to ensure that scheduler commands are available in the path before you launch desktop.

Note There is no need to install RSM unless the you are using the scheduler GUI on a post processing node that is different than the than the job submission node. In this case, RSM must be configured with the scheduler type and path.

A post processing nodes is a node in the cluster that can run the ANSYS Electromagnetics desktop in graphical mode. A job submission node is a node in the cluster in which job submission commands are available.

Turn OFF firewall between cluster nodes.

Scenario 1: The post-processing node and job-submission node roles are served by distinct machines.

In this case, perform the following configuration:

The job-submission node should be configured to run RSM service, which serves as a proxy to scheduler. The RSM Service should be running as 'root' in order to facilitate jobs running using the credentials of the job's owner. A **configuration file in the RSM installation folder should be edited** to specify information regarding the scheduler that manages jobs on this cluster. A block labeled 'Scheduler' must be included within the 'AnsoftCOMDaemon' block. This block contains two string entries:

- SchedulerName: this contains the unique part of the scheduler proxy library name
- ConfigString: this contains a scheduler specific configuration string

The case of the SchedulerName string is significant on Linux because Linux file names are case sensitive. The case of the SchedulerName string is not significant on Microsoft Windows. In ANSYS Electromagnetics Suite 15.0, the possible scheduler names are: lsf and sge. The ConfigString entry is a scheduler specific configuration string, described below.

In addition, the AnsoftRSMService must be started with appropriate environment variables set. Generally, the environment variables must be set the same as they would be set for using the scheduler via command lines.

LSF Details

For the LSF scheduler proxy library, the ConfigString entry in the ansoftrsmservice.cfg configuration file is ignored. It may be empty or omitted entirely.

The AnsoftRSMService must be started with the environment set as it would be set for submitting jobs to the LSF cluster.

- For Linux, the cshrc.lsf or the profile.lsf file may be sourced to set up the environment, depending on the shell.
- For Microsoft Windows, the PATH environment variable should be set to include the directory containing the LSF commands; this directory should be before any other directory containing files with the same name as any LSF commands.

Example ansoftrsmservice.cfg configuration file:

```
$begin 'AnsoftCOMDaemon'
   $begin 'Managed COM Servers'
   $end 'Managed COM Servers'
   $begin 'Scheduler'
   'SchedulerName'='lsf'
   'ConfigString'=''
   $end 'Scheduler'
$end 'AnsoftCOMDaemon'
```

Scenario 2: The post-processing node and job-submission node roles are served by the same machine.

The **Select Scheduler...** command (as described in the Job Management User Interface for LSF section) is used to gather details about the scheduler. In this case, the Desktop process should be started in an environment suitable for submitting jobs to the scheduler. See below for details.

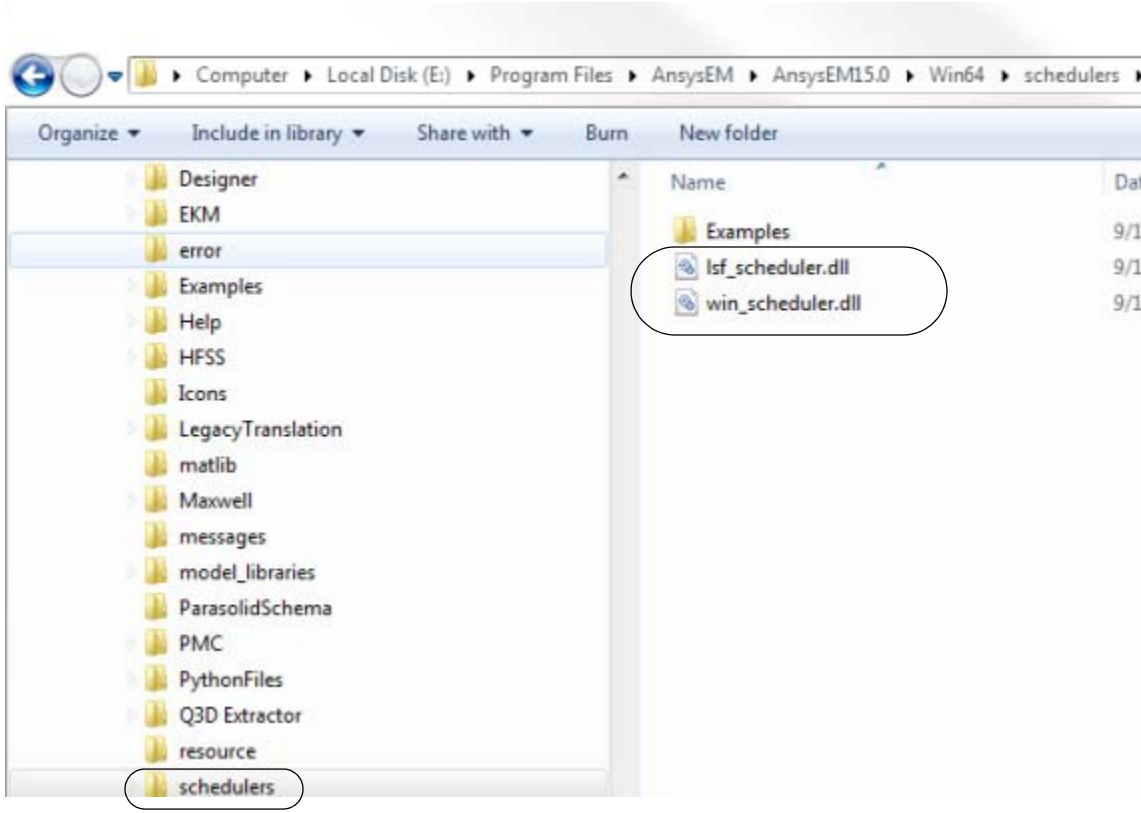
LSF on Linux

The environment should be configured so that the following LSF environment variables are set appropriately for the LSF cluster in use: LSF_BINDIR, LSF_SERVERDIR, LSF_LIBDIR, and LSF_ENVDIR. In addition, the following LSF commands should be found in the LSF_BINDIR directory: "bsub", "bjobs", "bkill", "lsid", "lsrun", "lshosts", "bmgroup", "bparams" and "bqueues".

LSF on Microsoft Windows

The environment should be configured so that all LSF commands are found using the PATH environment variable. In particular, search for the following commands in the PATH should result in the LSF command being found: "bsub", "bjobs", "bkill", "lsid", "lsrun", "lshosts", "bmgroup", "bparams" and "bqueues". No other command with the same name should appear before the LSF command in the PATH.

ANSYS Electromagnetics product Installation Changes



Scheduler Proxy DLLs are installed in the 'schedulers' directory of installation

Desktop talks to Scheduler Proxy which in turn uses Scheduler commands. It is possible to add a proxy dll to support new schedulers without changing Desktop.

Related Topic

Integration with Platform's Load Sharing Facility (LSF)

General Terminology for LSF

What a Scheduler Does

Integration of ANSYS EM Products with LSF

Job Management User Interface for LSF

LSF Job Submission Guidelines

Known Issues for LSF

Troubleshooting for LSF

Workarounds for LSF

Command Line Enhancements for ANSYS EM Desktop Products

Integration of ANSYS EM Products with LSF

With LSF you do not need to setup 'Distributed Analysis Machines' or 'Remote Machine' in the **Tools>General Options>Analysis Options** tab. Instead, you submit an HFSS job to LSF, requesting appropriate resources for this job (number of processors, memory per processor, etc.). For example:

```
bsub -n 1 hfss.exe -Batchsolve -ng -local -machinelist num=1
OptimTee.hfss
```

```
bsub -n 4 hfss.exe -Batchsolve -ng -Distributed -machinelist
num=4 OptimTee.adsn
```

The HFSS job is queued by LSF until the requested resources are available. Upon resource availability LSF starts HFSS.exe with the specified command line on one of the allocated machines. During analysis, HFSS dynamically obtains the allocated 'Distributed Analysis Machines' from LSF. HFSS interfaces with LSF to launch engines on remote machines without going through Ansoft RSM.

Related Topics

Integration with Platform's Load Sharing Facility (LSF)

General Terminology for LSF

What a Scheduler Does

Job Management User Interface for LSF

Installation of ANSYS EM Tools on LSF Cluster

LSF Job Submission Guidelines

Known Issues for LSF

Troubleshooting for LSF

Command Line Enhancements for ANSYS EM Desktop Products

LSF Job Submission Guidelines

- ANSYS EM project should be available in a shared drive that is accessible to all machines in the cluster
- ANSYS EM project should be available using the same path on all machines of cluster
- Ensure sufficient space in project directory and temp directories
- Ensure sufficient memory per engine
- Choose the number of compute resources (Distributed Analysis machines and Multi Processing cores) so as to achieve desired scale factor and effective resource utilization

Stop an HFSS job cleanly - ensures that the results obtained until now are preserved

```
bkill -s TERM <jobid>.
```

Stop an HFSS job abruptly - results are most likely lost. You have to manually remove the project lock file

```
bkill <jobid>
```

Related Topics

Integration with Platform's Load Sharing Facility (LSF)

Job Management User Interface for LSF

Installation of ANSYS EM Tools on LSF Cluster

Integration of ANSYS EM Products with LSF

Known Issues for LSF

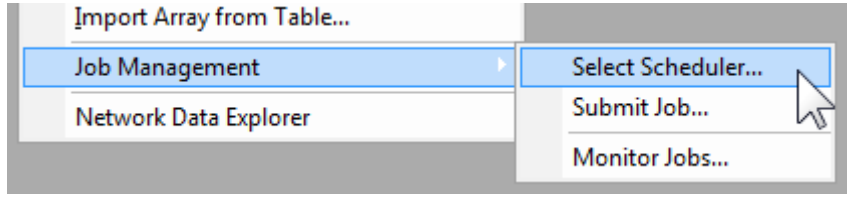
Troubleshooting for LSF

Command Line Enhancements for ANSYS EM Desktop Products

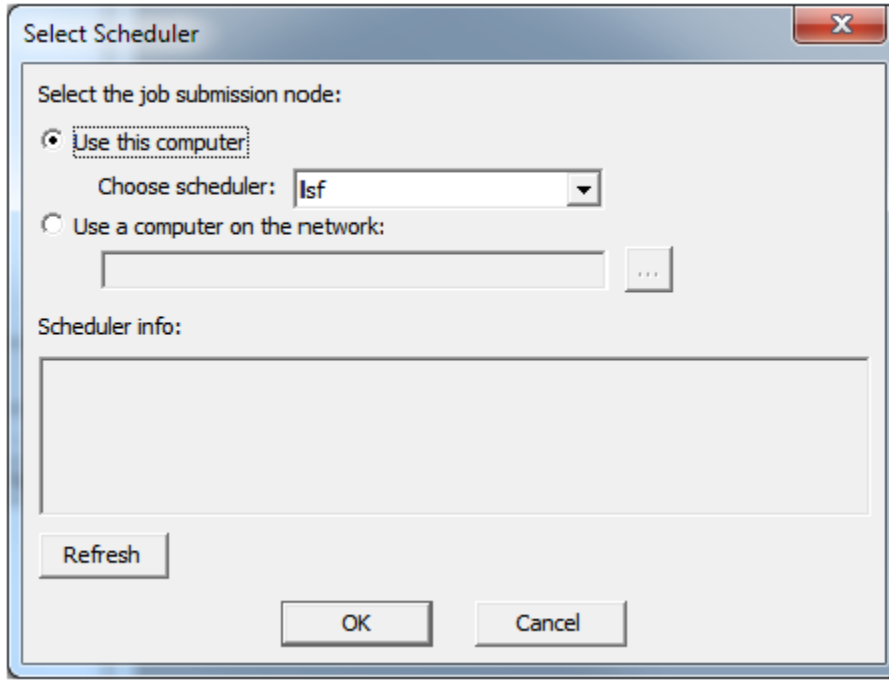
Aborting an Analysis

Job Management User Interface for LSF

The Job Management UI is accessed by running ANSYS Electromagnetics product Desktop on the designated 'Postprocessing node' of the cluster. The Desktop provides UI commands for Scheduler selection, Job submission and Job monitoring/control. You access the Scheduler User Interface by clicking **Tools>Job Management>Select Scheduler...**



Before you can use **Submit Job**, you must click **Select Scheduler** as the one-time initial step. This opens the Select Scheduler dialog.



Specify the following parameters:

- Job submission node: this is the node on the cluster where scheduler commands (such as LSF's bsub) are allowed to run.

  Choose **Use this computer** if scheduler commands are enabled on the post-processing node.

  Choose **Use a computer on network** if the cluster is configured in a manner as to disallow job-submission from the post-processing node. Specify node name appropriately.

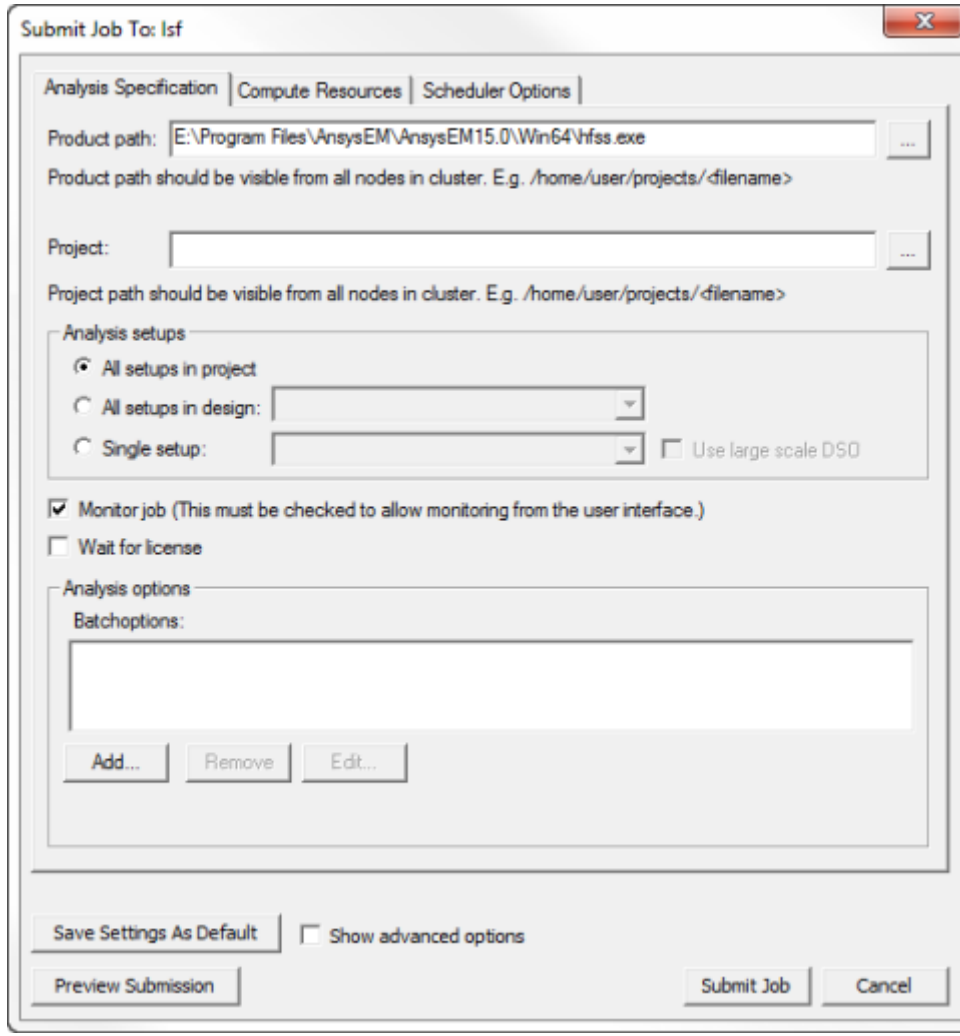
Pre-requisites: For this choice to work, the job-submission node must already be configured with a running Ansoft RSM service, as documented in 'Installation of ANSYS Electromagnetics Suite 15.0 on LSF Cluster.

- Scheduler: Available choices vary, depending on your configuration. It is possible for you to integrate their custom scheduler into this UI, through a scheduler proxy. When this is done, more choices will be available in the combo-box, one per custom scheduler proxy that is deployed in the installation

After specifying the job submission node, you can click **Refresh**. The scheduler information is then listed in the Scheduler info text field.

Once you select a scheduler, you can access the interface for job submission, monitoring and control. Click OK to close the dialog.

You access the Job submission UI by clicking **Tools>Job Management>Submit Job...**. This command launches a multi-tab dialog.



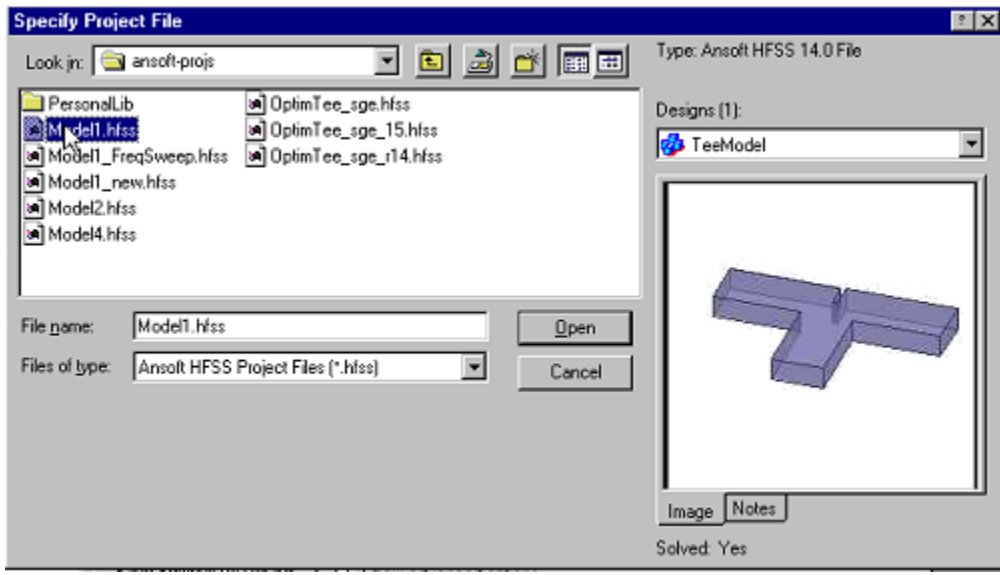
- The **Analysis Specification** tab has parameters to specify the product path, input project model, the analysis setup and analysis options (including batchoptions) that affect analysis algorithms.
- The **Compute Resources** tab specifies the amount of compute resources and how to select specific resources from the available pool (for example, <queue_name> is an LSF parameter that restricts job's candidate nodes to nodes in this queue).
- The **Scheduler Options** tab has analysis-independent parameters specific to the job, such as name, priority.

Analysis Specification tab for LSF

This tab lets you specify the following:

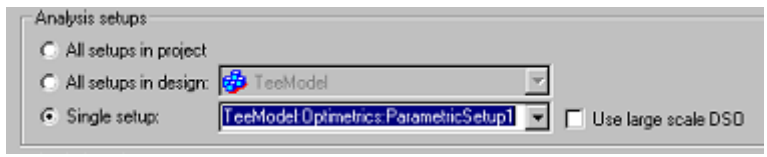
Project:

You can use the ellipsis button [...] to use a navigation window to browse. The path should be visible to all nodes in the cluster.



Analysis Setups:

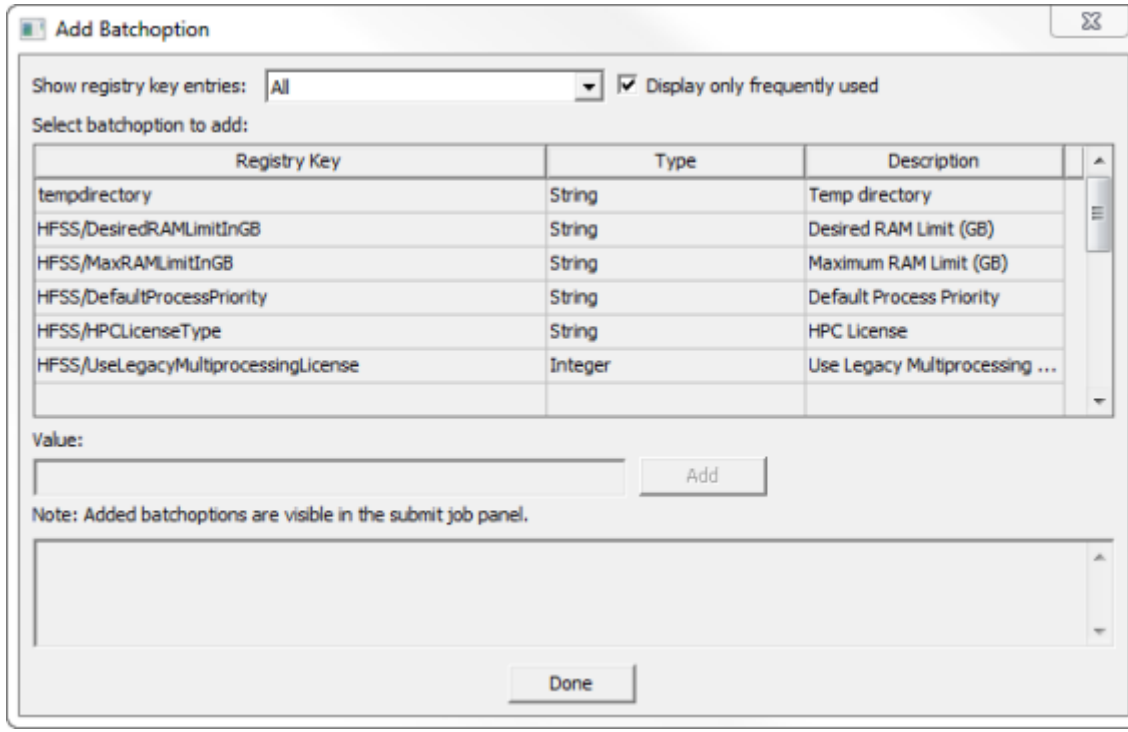
Specify All setups in the project, all in a design, or a single setup. If the setup includes a parametric sweep, the **Use large scale DSO** checkbox is enabled.



Analysis Options:

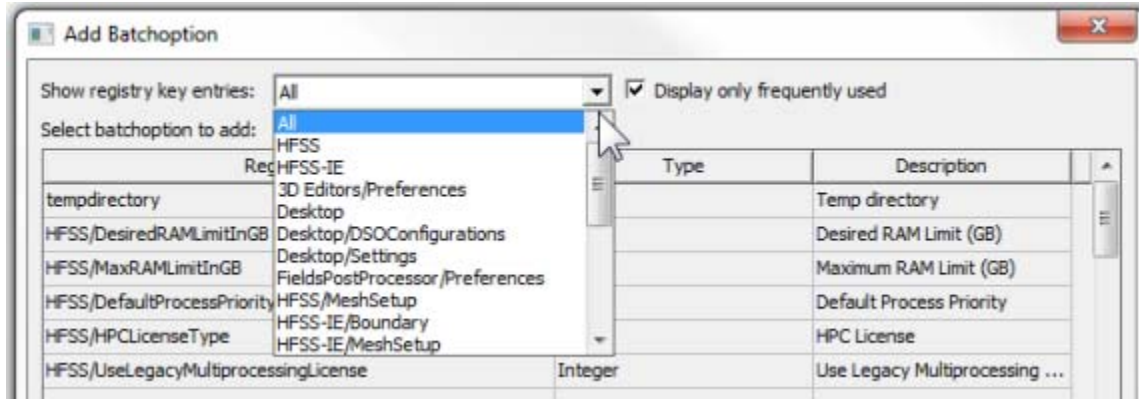
- You have checkboxes for whether to Monitor Job through the GUI, and whether to Wait for license.
- You can also specify Batchoptions. Click the **Add...** button to open a dialog for selecting the

Batchoptions.



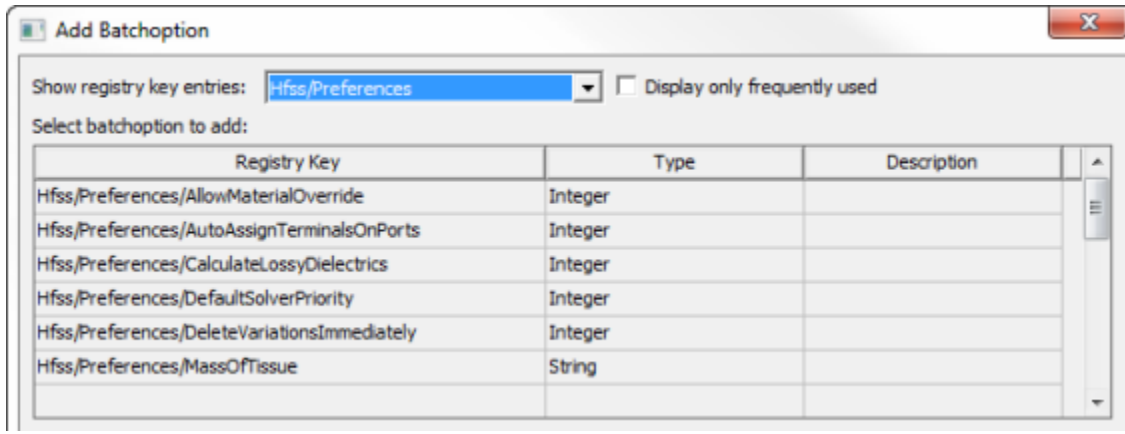
The lower Value field shows the legal values for the selected registry key. You can type the desired value into the upper text field under value. Click the **Add** button to accept the selection of the registry key with the specified value. Click **Done** close the **Add Batchoption** dialog.

At the upper left, a drop down menu lets you specify which registry key categories to display, whether All, or selected category.

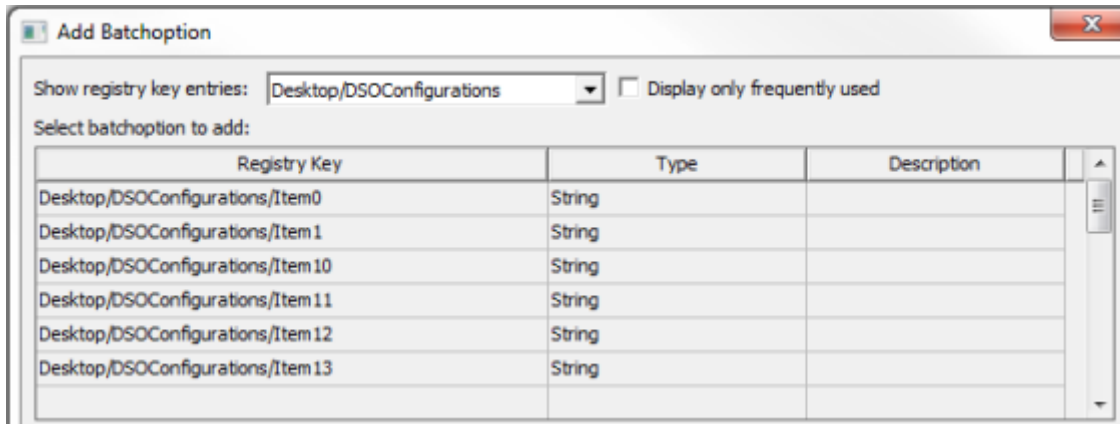


A checkbox lets you choose between displaying only frequently used entries (the default), or

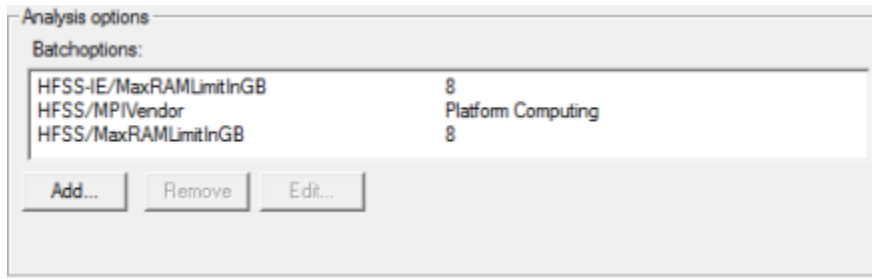
by unchecking, all options available for the selected group.



Note that for Large Scale DSO problems, you also have the option to specify predefined configurations.

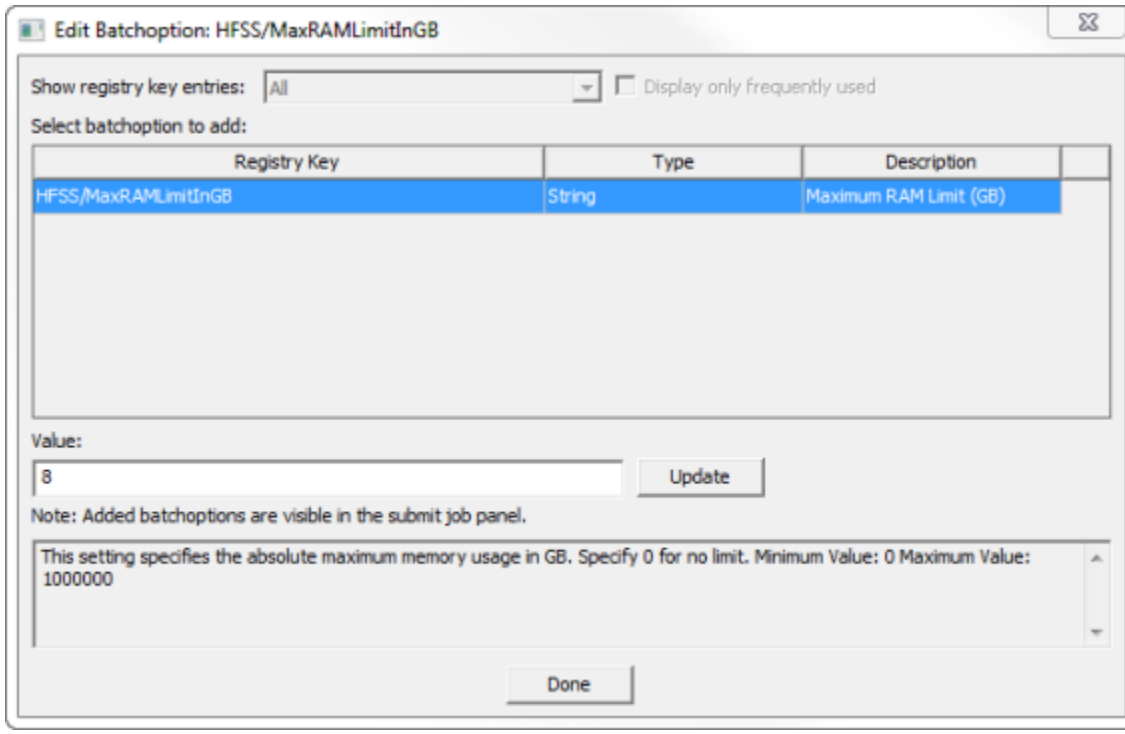


The added registry keys and values are listed in the Batchoptions field of the **Submit Job** dialog.



Selecting from the list enables buttons for removing or editing registry key values. Selecting a

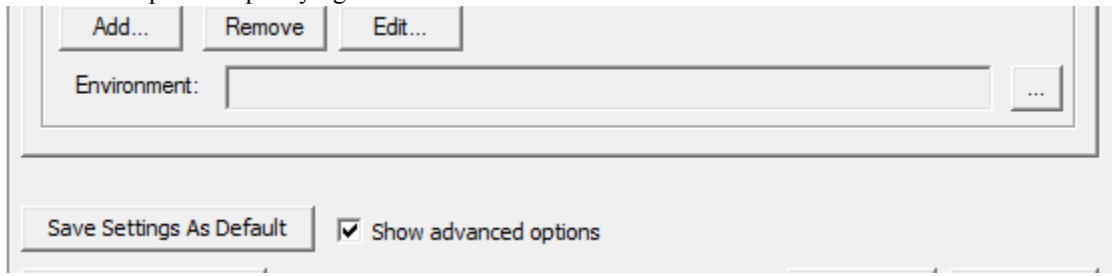
registry key and clicking **Edit...** opens the **Edit Batchoptions** dialog.



Here you can edit the Value field and update the value for the selected registry key.

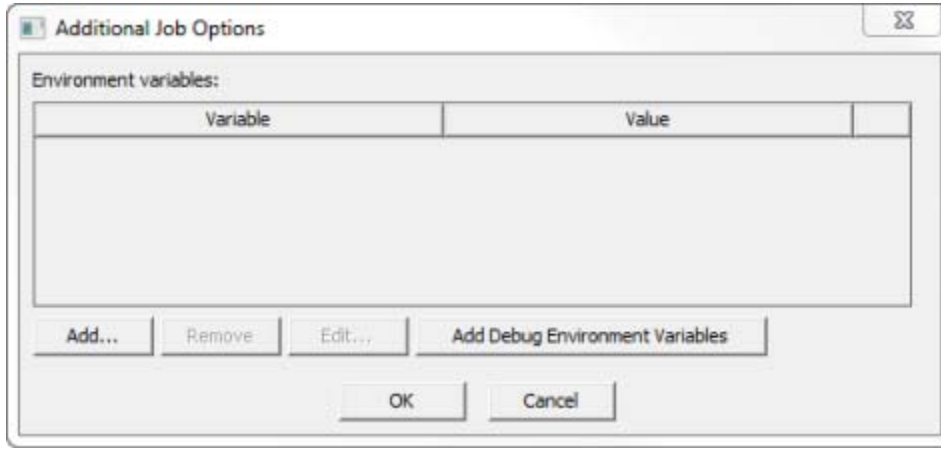
Environment:

If you check the Show advanced options check box, you can see the Environment field. This permits specifying Environment Variables.

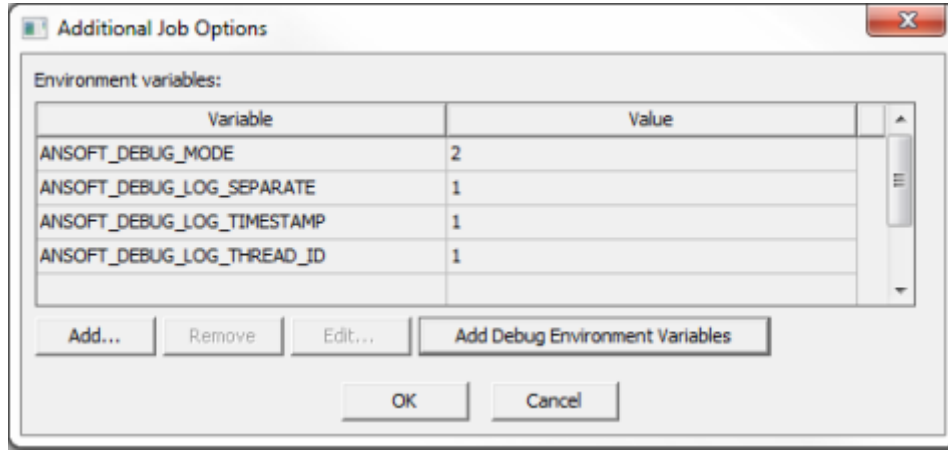


Click the ellipsis button [...] by the Environment field to open the **Additional Job Options**

dialog.



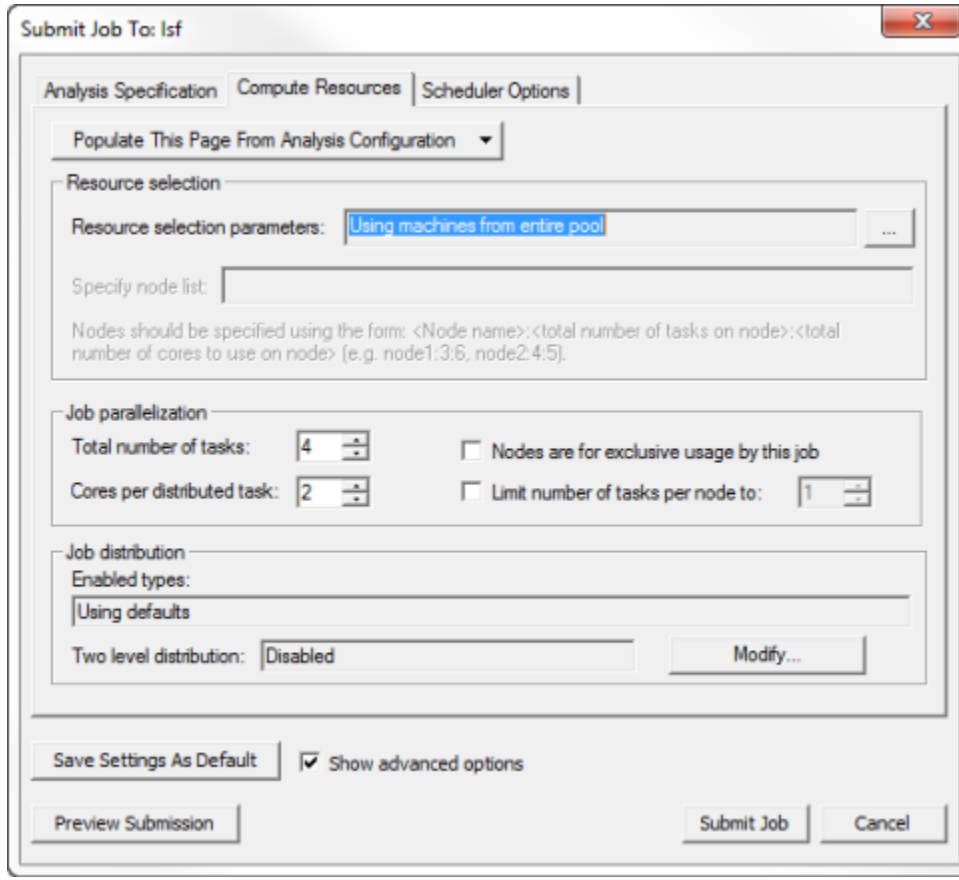
The **Add** button permits you to specify your environment variables. Once, added, you can select **Remove** or **Edit**. The **Add Debug Environment Variables** button adds variables of use in working with support.



- The **Save Settings as Default** button lets you save a current set of values as defaults the next time you invoke the Scheduler GUI. This can simply subsequent job submissions.

Compute Resources Specification Tab

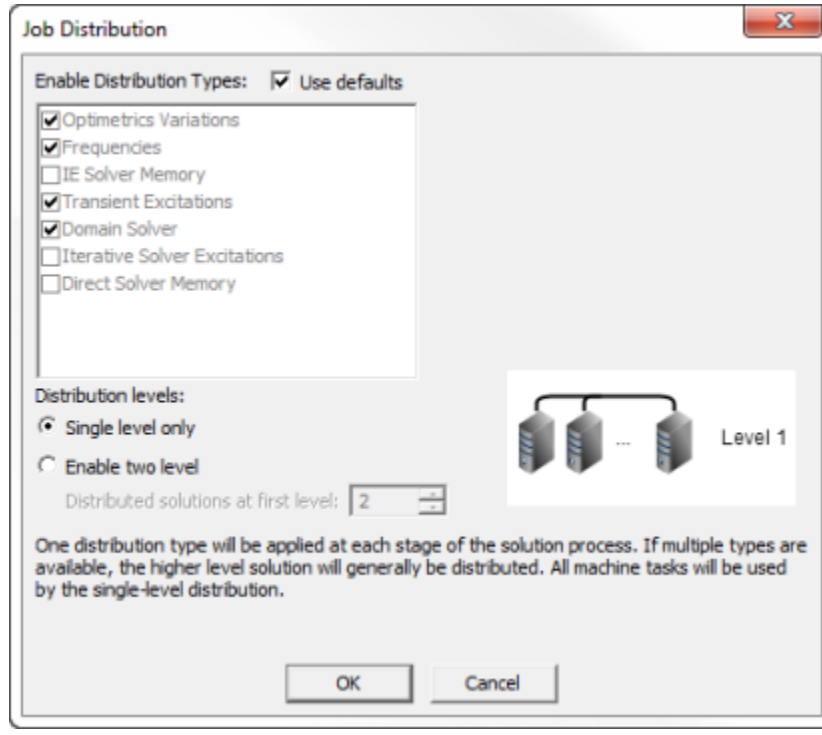
This tab lets you provide Resource selection and Job parallelization parameters. If you have created one or more Analysis configurations, you can select one of these to populate the fields on this page. Once you have specified parameters either from a configuration or by entering the information, you can Save Settings as Defaults, Preview a Submission, and Submit a job.



Job Distribution

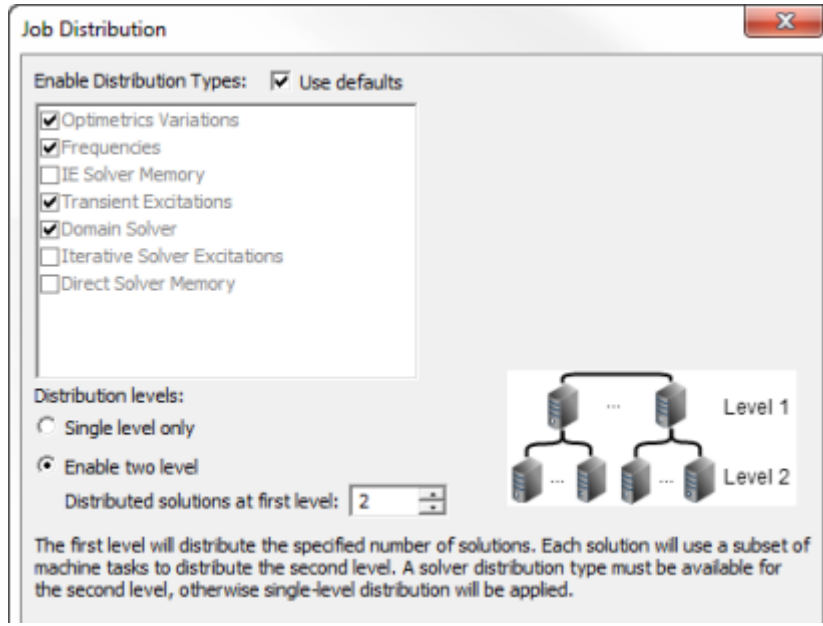
- Enabled types, such as Variations, Frequencies, Transient Excitations, Solver Domains, Direct Solver and Iterative Solver.
- Two level distribution, which may be disabled. Click the **Modify** button to display the **Job**

Distribution dialog.



Enabled Distribution types can modified here.

Second level distribution operates within DSO. If available and enabled you can specify a number of engines for level 1.



In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines, with two processors per engine, and also limit the number of engines per node to 4, the scheduler may increase the number of cores used in order to meet the limit specified for engines per node. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.

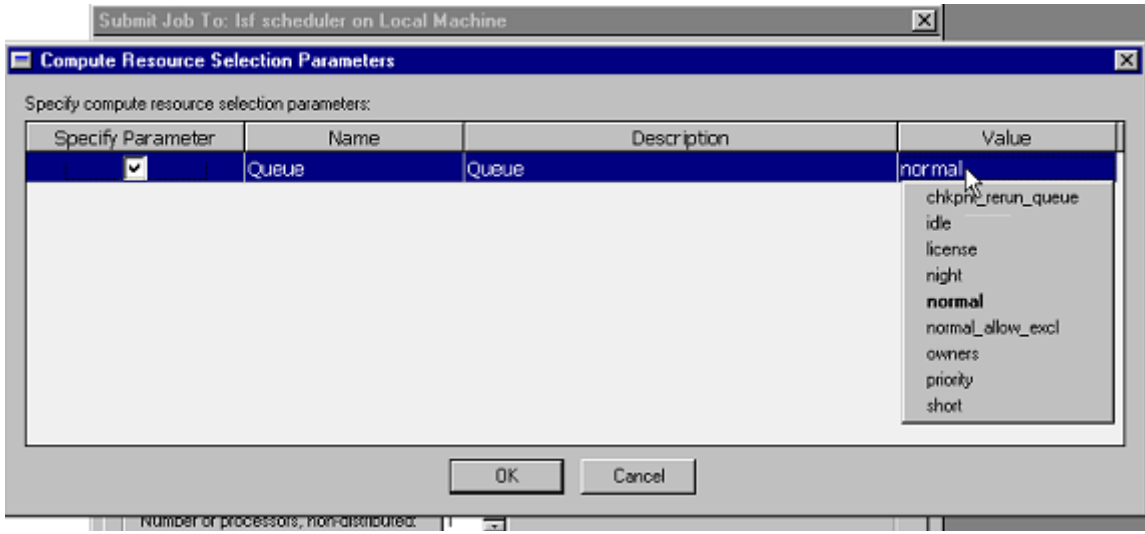
Resource Selection:

If you do not specify parameters for resource selection, LSF may submit jobs to any machine in the entire pool that is available.

- Resource selection parameters.

  Clicking the ellipsis button [...] opens a dialog for parameters specific to LSF, in this case **Queue**. The Specify Parameter checkbox in the dialog enables a parameter, and you can select

the Value for the Queue parameter from a drop down menu.



If you don't specify a parameter, the scheduler handles the situation.

- Specify node list

  Checking Specify node list enables the field for specifying a node list. In a computing environment where the available cores are not uniform, you can use this to have control over which resources your job will use.

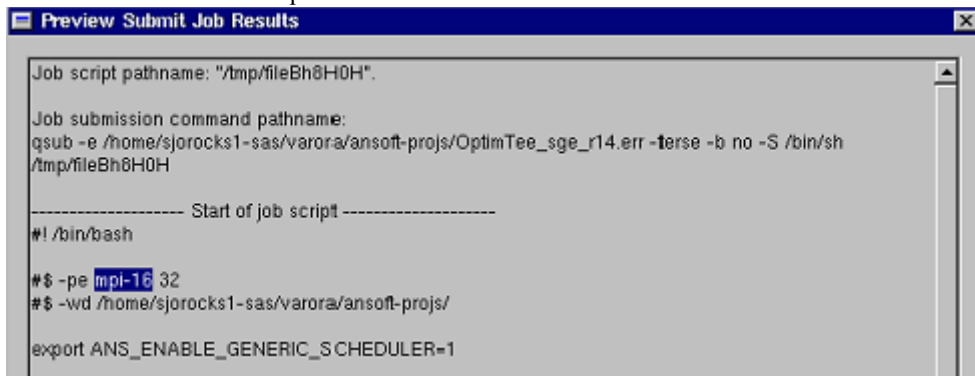
Job parallelization:

The Job parallelization fields let you specify

- Total number of tasks:
- Cores per distributed task.
- Whether nodes are for exclusive usage by this job
- Whether to limit the number of tasks per node to a value.

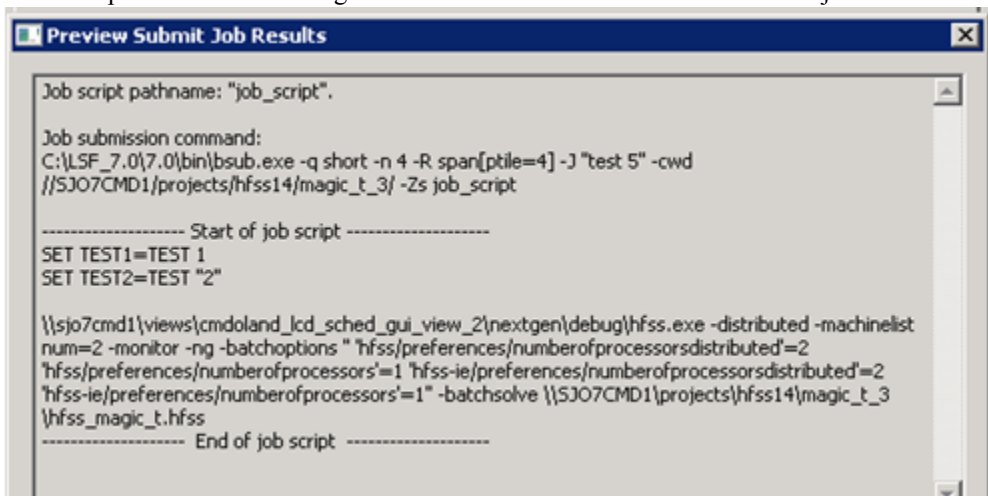
In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. Notice that a preview of

the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.

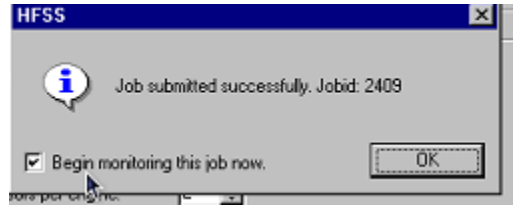


Preview submission

This opens a screen showing the bsub command to be used to submit the job.

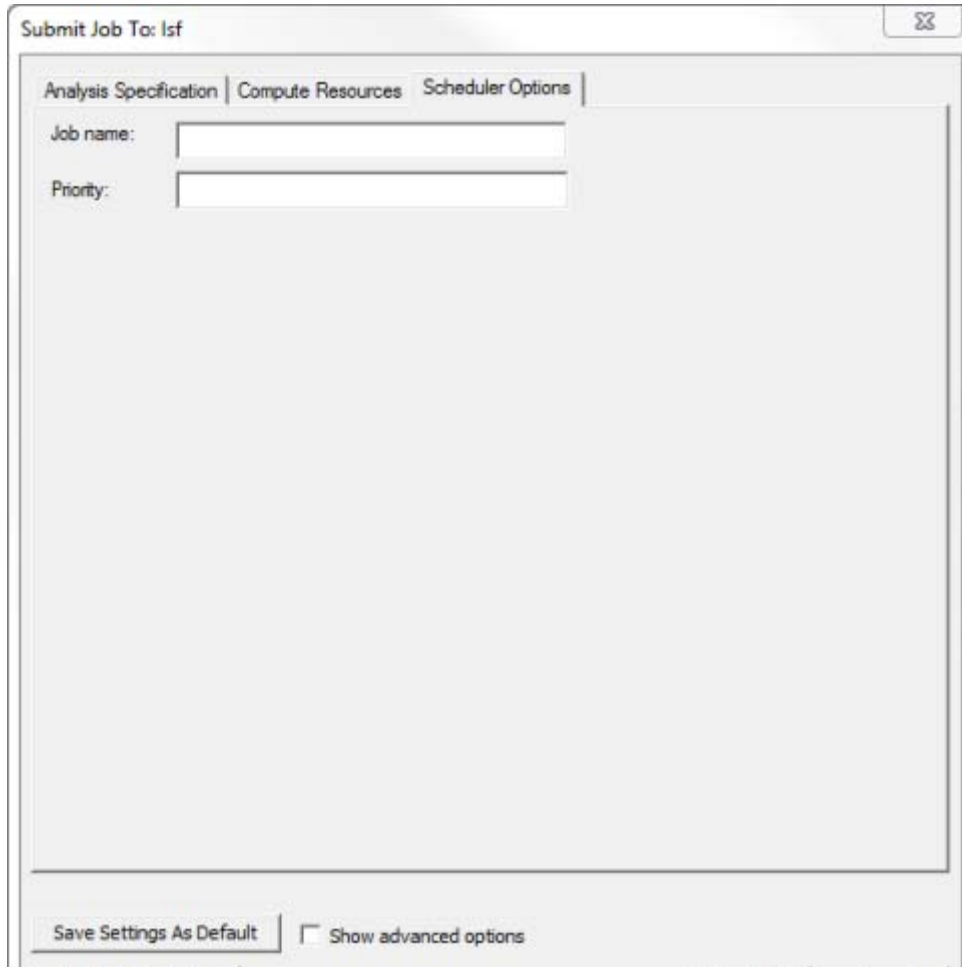


Submit job actually sends the batch command to the LSF cluster.A dialog reports a successful submit and presents a checkbox for monitoring.



Scheduler Options Tab

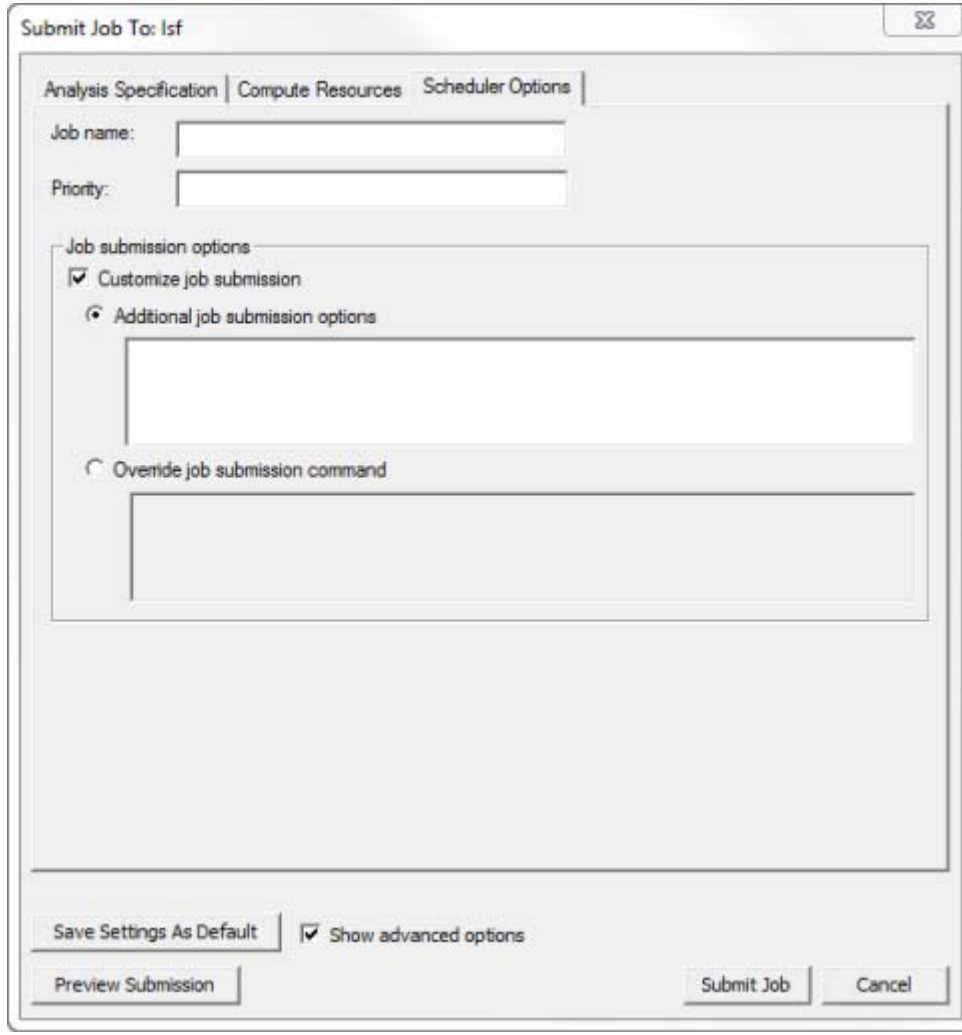
This tab lets you give Job name and Priority.



You continue to have the **Save Settings As Default**, **Preview Submission**, and **Submit Job** buttons.

If you check **Show advanced options**, you can also specify Job submission options.

The Job submission options permit you to customize a job submission in terms of addition job submission options, or override a job submission command.



When the "Override job submission" radio button is checked, the user specified options replace most of the job submission options, whereas when the "Additional job submission options" radio button is checked, the user specified options are appended to the bsub command.

You can see the effects of any custom additions by clicking **Preview Submission**.

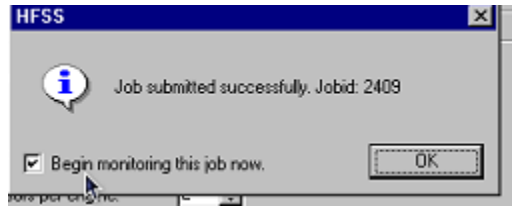
Related Topics

Integrated Job Monitoring for Job Management Interface for LSF

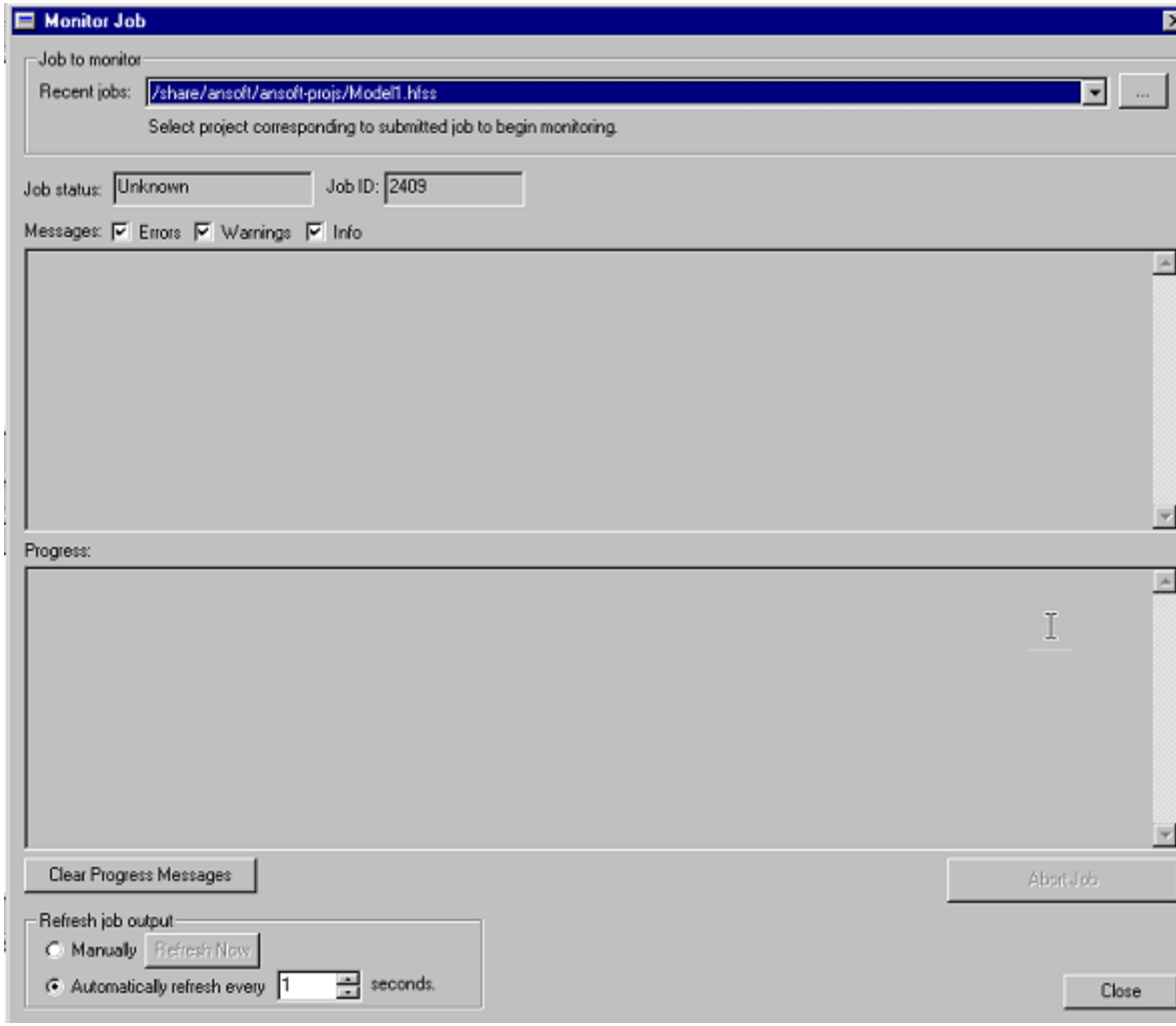
Scheduler Proxy Interfaces

Integrated Job Monitoring for Job Management Interface for LSF

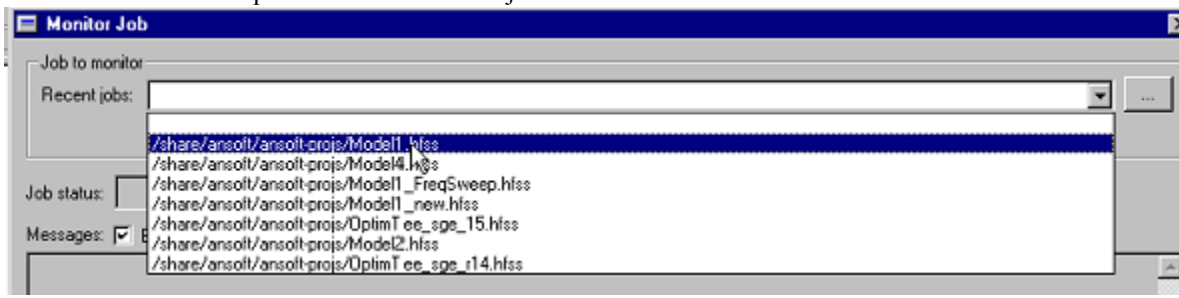
The job monitoring/control dialog is launched through the command **Tools>Job Management>Monitor Jobs...** or by checking **Begin monitoring this job now** in the information window reporting successful job submission.



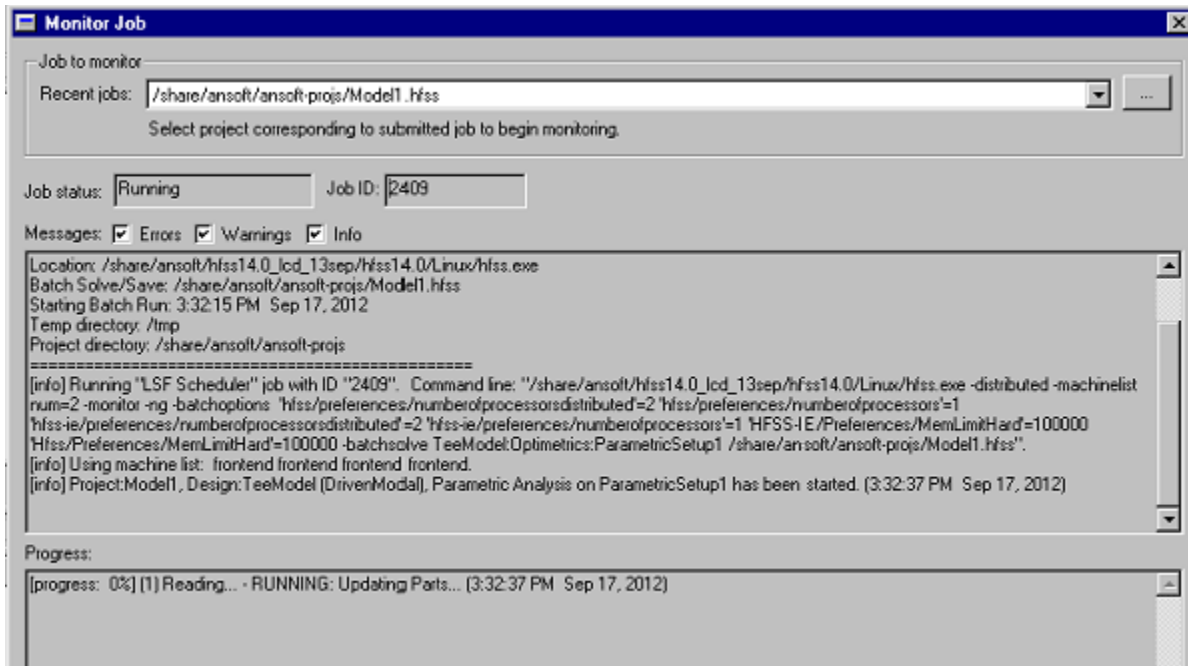
In response to either invocation, the **Monitor Job** dialog opens:



In this dialog, user selects the same project they submitted. You begin by selecting the job to monitor from a pull down list of recent jobs.



The lower left has corner of the dialog has options for manual refresh or to Automatically refresh every specified number seconds. After refresh, the messages for the job are displayed. The format of the text is essentially same as output of the '-monitor' command-line option.



If desired, you can use the check boxes to filter the messages listed in terms of whether to monitor Errors, Warnings, or Info messages.

Related Topics

Scheduler Proxy Interfaces

Job Management User Interface for LSF

LSF Command Used to Launch Remote Engine Processes

When using direct integration with LSF, the LSF **lsrun** command is used to launch remote engine processes. This command does not allow the LSF scheduler to track the resources used by these processes. As a result, the resource usage reported by the LSF **bacct** command will not include resources used by the remote engine processes.

Submitting ANSYS EM LSF Batch Jobs

The LSF bsub command may be used to submit ANSYS Electromagnetics jobs. The typical command format is:

`bsub` *bsub_args* *ansys_exe* *ansys_args*

where:

- *bsub_args* are the options of the `bsub` command,
- *ansysEM_exe* is the pathname of the ANSYS Electromagnetics desktop executable to launch, and
- *ansys_args* are the arguments to the ANSYS Electromagnetics desktop executable.

Related Topic

Job Management User Interface for LSF

Current Working Directory with LSF

The current working directory of the ANSYS Electromagnetics Desktop process must exist on all compute hosts where analysis may be run. This is needed because the ANSYS Electromagnetics job uses the LSF lsrun or blaunch commands to start the remote analysis engines. When these LSF commands start a remote process they always change the working directory to the working directory of the process submitting the command. If the directory does not exist on a remote host, then the lsrun or blaunch command will fail on that host, and the remote analysis engine will not be started on that host.

Quoting ANSYS EM Command or Arguments for LSF

If the ANSYS EM tool executable pathname (*ansys_exe*) or any of the arguments of the ANSYS tool command (*anssys_args*) contain characters which are interpreted by the command shell, then these special characters must be properly quoted to ensure that the correct command is launched by LSF. This is very common on Windows systems, where many executable file pathnames contain embedded space characters. A similar problem may occur on any platform if any of the *ansoft_args* require single quote, double quote or space characters. Note that the ANSYS Electromagnetics desktop command is processed by the shell twice. It is processed by the shell when the **bsub** command is processed, and again when the job is started.

bsub Arguments

The LSF **bsub** command has a large number of options that may be used to control the submission process. Only a few options which are often used with ANSYS Electromagnetics jobs are mentioned here. The following options may be used to submit serial or parallel LSF jobs.

-n *min_proc*`,` *max_proc* `or` **-n** *min_proc*

Submits a parallel job, specifying the number of processors (or slots) required for the job. Here, *min_proc* is the minimum number of processors, and *max_proc* is the maximum number of processors. If no maximum is specified, then exactly *min_proc* processors are requested. If PARALLEL_SCHED_BY_SLOT=Y in lsb.params, this option specifies the number of slots required to run the job, not the number of processors. If the **-n** command line option is not specified, then the job is submitted as a serial batch job.

-R `"span[ptile=n]"`

There are many ways to use the -R "res_req" option to the bsub command. We only cover -R "span[ptile=n]" here, because this option is very useful for ANSYS Electromagnetics jobs. When this option is specified, the LSF scheduler will allocate n processors (or slots) on each host to this job, even if more processors are available on the host.

-x

All hosts running this job operate in exclusive execution mode. The job will only run on a host having no other jobs running on that host. No other batch jobs will be started on a host while this job is running on that host. The **-m** *host_name* option of the **lsrun** or **lsgrun** commands may be used to force an interactive job to run on a host in exclusive mode. The **-m** *host_name* option of the **brun** command may be used to force a batch job to run on a host in exclusive mode.

See the LSF documentation for a complete list of options for the **bsub** command.

Related Topic

Job Management User Interface for LSF

Monitoring ANSYS EM LSF Batch Jobs

The LSF bpeek command may be used to monitor LSF batch jobs, if the Desktop command line for the job includes the -monitor command line option. Also see Integrated Job Monitoring for Job Management Interface for LSF.

ANSYS Electromagnetics desktop -monitor Command Line Option

The Ansoft -monitor command line option enables batch job output to the standard output and standard error streams. The warning, info, and progress messages are sent to the standard output stream. The error and fatal messages are sent to the standard error stream.

LSF bpeek Command

The LSF bpeek command may be used to monitor job progress. The command **pbeek [-f]** *job_id* displays the standard output and standard error produced by the job with id *job_id* from the job start to the current time (the time when the command is executed). This command is only valid for jobs that have not yet finished. When used with the **-f** option on Linux, the output of the job is displayed using the command tail **-f**, so that ongoing progress may be monitored.

In order to display messages to standard output and standard error, specify the **-monitor** command line option on the ANSYS EM tool command line. Then, these messages can be seen using the LSF bpeek command.

Related Topic

Job Management User Interface for LSF

Integrated Job Monitoring for Job Management Interface for LSF

Terminating ANSYS EM LSF Batch Jobs

To cancel or terminate an ANSYS EM LSF batch job, we recommend using the Monitoring Graphical User interface to terminate jobs cleanly, rather than using the bkill commands. Using this approach will allow the ANSYS EM batch job to shut down in an orderly fashion. Using the LSF **bkill** command without the **-s SIGTERM** option or simply terminating the job processes may cause some of the following problems:

- Some engine processes are not shut down, and continue to run
- LSF job is not fully removed
- Project lock file is not removed
- **Linux only:** MainWin core service processes (watchdog, mwrpcss and/or regss) are not stopped

Some of these may interfere with submission of additional LSF batch jobs. For example, it may be necessary to manually remove the project lock file to submit another batch job for the same project. On Linux, MainWin core service processes may also interfere with starting subsequent Ansoft batch jobs. Normally, these processes should timeout and end 15 seconds after the ANSYS Electromagnetics product shuts down. Any MainWin core service processes (watchdog, mwrpcss and/or regss) that continue to run for more than 15 seconds after the product has stopped may be hung. The hung processes may need to be manually killed, after ensuring that these processes are associated with an ANSYS EM job that has finished or terminated.

Related Topics

Integrated Job Monitoring for Job Management Interface for LSF

Example LSF bsub Command Lines

The following Linux and Windows examples all use HFSS as the ANSYS EM product, but similar command lines will work for all ANSYS EM products.

Linux Examples for LSF

If the hfss command is included in the **bsub** command line, then the entire hfss command will be processed by the command shell two times. The hfss command is processed when the **bsub** command is processed by the shell. It will be processed again when the hfss command is started by the scheduler.

Several examples show the entire hfss command line enclosed in double quotes ("), while the double quote (") characters within the hfss command line are replaced by escaped double quotes ("\""). This ensures that the quoted arguments of the hfss command are processed correctly. The last example shows how to use a shell script so that the hfss command line will be processed by the command processor only once. The hfss command is placed in the shell script, and then the shell script pathname is placed in the **bsub** command line. Then, the hfss command is only processed by the command processor when the job is started. When using this approach, the shell script should be accessible from all of the cluster hosts.

Serial job:

```
bsub -n 1 /Program Files/AnsysEMAnsysEM15.0/Win64/hfss -ng
-BatchSolve -machinelist num=4 ~/projects/OptimTee.hfss
```

The -n 1 option indicates that this job runs on one core.

Serial job that needs a minimum of 4GB:

```
bsub -n 1 -R "select[mem>4000]"
/Program Files/AnsysEM/AnsysEM15.0/Win64/hfss -ng
-BatchSolve -machinelist num=4 ~/projects/OptimTee.hfss
```

The -R "select[mem>4000]" option indicates that this needs a minimum of 4 GB memory.

Multi-processing job using 4 cores:

```
bsub -n 4 -R "span[ptile=4]"
"/Program Files/AnsysEM/AnsysEM15.0/Win64/hfss -ng -BatchSolve
-batchoptions -machinelist num=4 ~/projects/OptimTee.hfss"
```

- The -R "span[ptile=4]" option indicates that the four cores need to be on the same machine.
- The -batchoptions option indicates that HFSS should use four cores for multi-processing.
- The entire hfss command is in double quotes, and the double quotes enclosing the -batchoptions value are escaped. Each of these double quotes is replaced by the sequence "\"".

Distributed processing job using 4 engines:

```
bsub -n 4 /Program Files/AnsysEM/AnsysEM15.0/Win64/hfss -ng -
BatchSolve -Distributed ~/projects/OptimTee.hfss
```

- The -n 4 option indicates that the four cores are needed for the job.
- The -Distributed option indicates that this is a DSO job, so that multiple engines will be started. Because 4 cores are allocated to the job, the job will run 4 engines. The -Distributed option can have additional options, such as includetypes=xxx, excludetyeps=xxx, maxlevels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".

Distributed processing and multi-processing job using 4 cores, with 2 cores for multi-processing:

```
bsub -n 4 -R "span[ptile=2]" ~/projects/OptimTee.csh
```

Shell script (~/projects/OptimTee.csh):

```
#!/bin/csh
/Program Files/AnsysEM/AnsysEM15.0/Win64/hfss -ng -BatchSolve
-Distributed -machinelist num=2 -batchoptions
 ~/projects/OptimTee.hfss
```

- The **-n 4** option indicates that the four cores are needed for the job.
- The **-R "span[ptile=2]"** option indicates that the cores must be allocated in groups of two

cores on the same machine.

- The **-machinelist num=2** option indicates that this is a DSO job and that a total of two engines will be started.
- The **hfss** command is placed in the shell script (~/projects/OptimTee.csh). In the **bsub** command line, the **hfss** command is replaced by the shell script pathname.

Windows Examples for LSF

If the hfss command is included in the bsub command line, then the entire hfss command will be processed by the command processor cmd.exe two times. The hfss command is processed when the bsub command is processed by the command processor. It will be processed again when the hfss command is started by the scheduler.

The first three examples show the entire hfss command line enclosed in double quotes ("), while the double quote (") characters within the hfss command line are replaced by escaped double quotes (\"). This ensures that the quoted arguments of the hfss command are processed correctly. The remaining examples show how to use a batch file so that the hfss command line will be processed by the command processor only once. The hfss command is placed in a batch file, and then the batch file pathname is placed in the bsub command line. Then, the hfss command is only processed by the command processor when the job is started. When using this approach, the batch file should be accessible from all of the cluster hosts.

Serial job:

```
bsub -n 1 "\"C:\Program
Files\AnsysEM\AnsysEM15.0\Win64\hfss.exe\"
-Ng -BatchSolve \\host\share\projects\OptimTee.hfss"
```

- The -n 1 option indicates that this job runs on one core.
- The entire hfss command is in double quotes, and the double quotes enclosing the hfss pathname are escaped. Each of these double quotes is replaced by the sequence \".

Serial job that needs a minimum of 4GB:

```
bsub -n 1 -R "select[mem>4000]"
"\"C:\Program Files\AnsysEM\AnsysEM15.0\Win64\hfss.exe\"
-Ng -BatchSolve \\host\share\projects\OptimTee.hfss"
```

- The -R "select[mem>4000]" option indicates that this needs a minimum of 4 GB memory.
- The entire hfss command is in double quotes, and the double quotes enclosing the hfss pathname are escaped. Each of these double quotes is replaced by the sequence \".

Multi-processing job using 4 cores:

```
bsub -n 4 -R "span[ptile=4]"
"\"C:\Program Files\AnsysEM\AnsysEM15.0\Win64\hfss.exe\"
 -Ng -BatchSolve -batchoptions -machinelist num=4
\\host\share\projects\OptimTee.hfss"
```

- The -R "span[ptile=4]" option indicates that the four cores need to be on the same machine.
- The -batchoptions option indicates that HFSS should use four cores for multi-processing.
- The entire hfss command is in double quotes, and the double quotes enclosing the hfss pathname and the -batchoptions value are escaped. Each of these double quotes is replaced by the sequence \".

Distributed processing job using 4 engines:

```
bsub -n 4 \\host\share\projects\OptimTee.bat
```

Batch File (\\host\share\projects\OptimTee.bat) Contents:

```
"C:\Program Files\AnsysEM\AnsysEM15.0\Win64\hfss.exe"
-Ng -BatchSolve
-Distributed \\host\share\projects\OptimTee.hfss
```

- The -n 4 option indicates that the four cores are needed for the job.
- The -Distributed option indicates that this is a DSO job, so that multiple engines will be started. Because 4 cores are allocated to the job, the job will run 4 engines.
- The hfss command is placed in the batch file \\host\share\projects\OptimTee.bat. In the bsub command line, the hfss command is replaced by the batch file pathname.

Distributed processing and multi-processing job using 4 cores, with 2 cores for multi-processing:

```
bsub -n 4 -R "span[ptile=2]" \\host\share\projects\OptimTee.bat
```

Batch File (\\host\share\projects\OptimTee.bat) Contents:

```
"C:\Program Files\AnsysEM\AnsysEM15.0\Win64\hfss.exe" -Ng
-BatchSolve -Distributed -machinelist num=2 -batchoptions
\\host\share\projects\OptimTee.hfss
```

- The -n 4 option indicates that the four cores are needed for the job.
- The -R "span[ptile=2]" option indicates that the cores must be allocated in groups of two cores on the same machine.
- The -machinelist num=2 option indicates that this is a DSO job and that a total of two engines will be started.
- The hfss command is placed in the batch file \\host\share\projects\OptimTee.bat. In the bsub command line, the hfss command is replaced by the batch file pathname.

Known Issues for LSF

- Desktop or remote machine has multiple IP addresses, which we don't support.
- There are core dump files at the end of job's running. Results are computed correctly though
- On Windows, HFSS should be installed on every machine of cluster
- Firewall should be turned off on the machines in the cluster
- UAC should be disabled on Vista (only Windows)

- Sometimes LSF kills an HFSS job (for e.g. job gets preempted due to a high priority job). HFSS doesn't handle such a situation gracefully resulting in the presence of .lock file in the project directory. User needs to manually delete the lock file before continuing with further analysis.
- When an LSF job is killed, the MainWin services (watchdog, regss, and mwrcpss) could keep running. The result is that later jobs cannot start on the machine. The fix is to kill off these processes before starting a new job
- Analysis fails abruptly when running out of resources (cpu/memory/disk)

  The major reason for job failure is due to insufficient resources given to job

  Issue is addressed (via graceful failure) in the next release of ANSYS EM products

Related Topics

Integration with Platform's Load Sharing Facility (LSF)

General Terminology for LSF

What a Scheduler Does

Installation of ANSYS Electromagnetics Suite 15.0 on LSF Cluster

Integration of ANSYS Electromagnetics products with LSF

LSF Job Submission Guidelines

Troubleshooting for LSF

Command Line Enhancements for ANSYS EM Desktop Products

Troubleshooting for LSF

Ensure that the LSF 'lsrun' command is enabled.

Make sure there is no user error. For example:

- Use correct path to HFSS and full path to project
- Ensure sufficient resources (cpu/memory/disk) are allocated to job: check space in temp directory, space in project directory (consider quotas)
- ANSYS EM project to be available on the execution host
- Ensure job submitter has permissions to read/write to project directory and read/execute permissions to installation directory
- Check for locked project

Find if this is a standalone product issue:

Run HFSS interactively on the machine, outside of the scheduler, and see if it comes up and analyzes.

Send the following logs to development:

- Output of LSF batch job corresponding to the analysis of the ANSYS EM project: You can obtain this using LSF commands: use bacct if job's output is not redirected to a file. For example,

  ```
  "bacct -l <jobid>"
  ```

- Batch log generated by ANSYS Electromagnetics product (typically, projectname.log, in the same directory as the project file)
- Debug logs generated by ANSYS Electromagnetics product, during the running of LSF job. You can set these Environment variables using the Submit Job dialog, by turning on Advanced options to display the Environment field. Click the ellipsis button [...] access the Add Debug Environment Variables button.

  `Set ANSOFT_DEBUG_MODE to 1`

  It is good to collect another set of logs with above value set to 7

  Create \\shared_drive\ansdebug directory that is accessible by all machines in the cluster

  Set ANSOFT_DEBUG_LOG to a file in \\shared_drive\ansdebug' directory, for example,

  `\\shared_drive\ansdebug\anslog`

  `Set ANSOFT_DEBUG_LOG_SEPARATE to 1.`

  Set ANSOFT_LSF_LOG to a distinct file in \\shared_drive\ansdebug' directory, for example,

  `\\shared_drive\ansdebug\lsf.log`
- For each pair of the machines, between which the remote analysis fails, run "ping remote-machine" and note the output
- For each machine in the network, dump the network interfaces (for example, run "ifconfig -a") and note the output
- Email all 5 logs to development

  - Job's log generated by LSF

  - Log generated by ANSYS Electromagnetics product

  - Entire \\shared_drive\ansdebug folder

  - Output from ping command

  - Output related to compute node's network interfaces

Related Topics

Workarounds for LSF

Integration with Platform's Load Sharing Facility (LSF)

General Terminology for LSF

What a Scheduler Does

Installation of ANSYS Electromagnetics Suite 15.0 on LSF Cluster

Integration of ANSYS Electromagnetics products with LSF

LSF Job Submission Guidelines

Known Issues for LSF

Command Line Enhancements for ANSYS EM Desktop Products

Aborting an Analysis

Workarounds for LSF

- Some users reported 'core' dumps during HFSS analysis, though analysis results are fine. Workaround: Limit size of core dumps to 0 through the following job submit option:

  ```
  bsub -C 0 -n <number-of-cores> -q <queue-name>
  ```

Note Letter 'C' must be upper-case

Related Topics

Integration with Platform's Load Sharing Facility (LSF)

Integration with PBS (Portable Batch System) Professional from Altair Engineering

The PBS Pro scheduler is only supported on Linux. Jobs may be submitted in any of the following ways [unchanged from R14]:

- Using PBS commands (qsub, etc.) or the PBS gui (xpbs)

ANSYS Electromagnetics software has been tested with the following version of PBS Pro:

- PBS Pro 10.1.0.91350

Submitting Ansoft PBS Batch Jobs

The PBS qsub command may be used to submit ANSYS EM batch jobs. The typical command format is:

`qsub` *qsub_args script*

where:

- *qsub_args* are the options of the **qsub** command,
- *script* is the pathname of the job script.

The job script is a shell script containing the Ansoft batch command or commands to be run. If a batch command line contains any characters that are special to the shell running the script, then these special characters should be quoted, as needed. The job script may also contain PBS directives on lines before the first executable line of the script. Any **qsub** options on the command line will take precedence over the PBS directives in the job script.

When a PBS batch job is started, the job script runs as the job user in a new shell. In this shell environment, the path must include the directory containing the PBS commands. You should ensure that the path set in the shell startup script, i.e., .cshrc, .profile, .bashrc, etc. includes the directory containing the PBS commands.

Serial PBS Batch Jobs

In the PBS documentation, serial batch jobs are also called single-node jobs. In general, any job submitted without specifying the -l nodes=value command line argument, will run as a serial or single-node job.

See the section on Monitoring Ansoft PBS Batch Jobs for options that can facilitate monitoring of Ansoft batch jobs.

Parallel PBS Batch Jobs

In the PBS documentation, parallel batch jobs are also called multi-node jobs. When an Ansoft batch job is run as an PBS parallel job, the PBS scheduler will select the hosts for the distributed analysis job based on the qsub command line arguments, the PBS resource directives from the job script, and the status of the hosts when the job is run. The desktop process will be started on one of these hosts. The desktop process will obtain the list of hosts allocated to the job from the PBS scheduler, and start analysis processes on these hosts, as needed, using the PBS scheduler facilities. To run a PBS parallel job, the job must be submitted with a -l nodes=value **qsub** command line argument or with a -l nodes=value PBS directive in the job script.

See the section on Monitoring Ansoft PBS Batch Jobs for options that can facilitate monitoring of Ansoft batch jobs.

Related Topics

Monitoring Ansoft PBS Batch Jobs

Example PBS qsub Command Lines

What a Scheduler Does

qsub Arguments

The PBS **qsub** command has a large number of options for control of the submission process. In this section, we review the -l nodes=*value* command line option with Ansoft parallel batch jobs.

This option or directive has the following format:

`-l nodes=`*node_spec* `[+`*node_spec*`...][#`*suffix*`]`

where *node_spec* is one of the following

`nodename[:`*pc_spec*`[:`*pc_spec*`...]]`

Host name of the specified node, followed by optional **ppn** or **cpp** specifiers.

`[`*N*`][:`*property*`[:`*property*`...]][:`*pc_spec*`[:`*pc_spec*`...]]`

Optional number of nodes, followed by optional node properties, followed by optional **ppn** or **cpp** specifiers. If the number *N* is omitted, then the default value of 1 host is used.

Here, the optional **ppn** or **cpp** specifiers *pc_spec* are of form:

`ppn=X`

Number of processes (tasks) per node. Default is 1 if not specified.

`cpp=Y`

Number of CPUs (threads) per process. Default is 1 if not specified.

The optional global suffix, #suffix, which applies to all hosts has one of the following values:

`#excl`

This suffix requests exclusive access to the allocated nodes.

`#shared`

This suffix requests shared (i.e., non-exclusive) access to the allocated nodes.

The total number of requested processes is determined by adding up the product of the number of nodes and the number of processes per node for each *node_spec*. In general, this should match the number of distributed engines specified in the ANSYS Electromagnetics desktop -Machinelist num=*num_distributed_engines* command line option.

The number of CPUs per process (**cpp**) specified in the PBS **qsub** command line or in the PBS directives in the script file should generally match the number of processors per engine specified in the Desktop **-batchoptions** value.

See the PBS documentation for a complete list of options for the **bsub** command, and further information on running multi-node jobs.

Related Topics

Integration with PBS (Portable Batch System) Professional from Altair Engineering

Monitoring ANSYS EM PBS Batch Jobs

Example PBS qsub Command Lines

Monitoring ANSYS EM PBS Batch Jobs

The suggestions below may be used for ANSYS EM Batch jobs run under PBS.

PBS qstat Command

The PBS **qstat** command may be used to display information on jobs and queues. In this section, several qstat command line options that may be used to monitor job progress are described.

The **`qstat -a`** command displays information about all jobs in the system.

The **`qstat -r`** command displays information about all running jobs in the system.

The **`qstat -s`** command resembles the **qstat -r** command; the only difference is that a comment from the scheduler or batch administrator is also shown for each job.

The **`qstat -au`** *userid* command displays information about all jobs owned by user userid.

The **`qstat -f`** *jobid* command displays all available information about the job with id *jobid*.

See the PBS manual pages for more information.

ANSYS EM -monitor Command Line Option for PBS

The ANSYS EM **-monitor** command line option enables batch job output to the standard output and standard error streams. The warning, info, and progress messages are sent to the standard output stream. The error and fatal messages are sent to the standard error stream.

The PBS scheduler redirects the standard output and standard error streams of batch jobs to files specified in the **qsub -o** [*hostname*:]*pathname* and the **-e** [*hostname*:]*pathname* command line options, respectively. If either option is not specified, then the associated stream is redirected to the default file pathname for that stream.

The **qsub -j join** option controls whether the standard error stream for the job will be merged with the standard output stream for the job. A join value of **oe** indicates that the interleaved standard output and standard error will be sent to the standard output file or stream. A join value of **oe** indicates

that the interleaved standard output and standard error will be sent to the standard error file or stream. A join value of n indicates that the standard output and standard error streams will not be joined. If the **qsub -j join** option is not specified, then the standard error and standard output streams will not be joined.

A user can monitor the progress of a job by checking the standard output file for progress, info and warning messages, and checking the standard error file for error and fatal messages.

Related Topics

Integration with PBS (Portable Batch System) Professional from Altair Engineering

qsub Arguments

Example PBS qsub Command Lines

Example PBS qsub Command Lines

All of the following examples show how to submit Linux hfss jobs on PBS, but similar command lines and job scripts will work for all ANSYS EM products. Most of the following examples are PBS "Single-node jobs." The last example is a PBS "multi-node jobs"; this example demonstrates how to specify the allocation of threads, tasks and nodes to a job.

Serial job:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
/opt/AnsysEM/AnsysEM15.0/hfss -ng -BatchSolve
~/projects/OptimTee.hfss
```

Serial job that needs a minimum of 4GB memory and two hours of real (wallclock) time:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l walltime=2:00:00
#PBS -l mem=4gb
/opt/AnsysEM/AnsysEM/hfss -ng -BatchSolve
~/projects/OptimTee.hfss
```

Multi-processing job using 4 cores:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l ncpus=4
```

```
/opt/AnsysEM/AnsysEM15.0/hfss -ng -BatchSolve -batchoptions
-machinelist num=4
~/projects/OptimTee.hfss
```

- The #PBS -l ncpus=4 directive indicates that four cores or CPUs are allocated to this job.
- The -batchoptions option indicates that HFSS should use four cores for multi-processing.

Distributed processing job using 4 engines on a single host:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l ncpus=4
/opt/AnsysEM/AnsysEM15.0/hfss14/hfss -ng -BatchSolve -
Distributed -machinelist num=4
~/projects/OptimTee.hfss
```

- The #PBS -l ncpus=4 directive indicates that four cores or CPUs are allocated to this job.
- The -Distributed option indicates that this is a DSO job, so that multiple engines will be started. Because 4 cores are allocated to the job, the job will run 4 engines. The -Distributed option may now have additional options, such as includetypes=xxx, excludetyeps=xxx, max-levels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".

Distributed processing and multi-processing job using 8 cores on two nodes, running 4 engines (two per node) with 2 cores for multi-processing:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l nodes=2:ppn=2:cpp=2#excl
/opt/Ansoft/HFSS14/hfss14/hfss -ng -BatchSolve -Distributed
-machinelist num=4 -batchoptions ~/projects/OptimTee.hfss
```

- The PBS directive **#PBS -l nodes=2:ppn=1:cpp=2#shared** indicates that two nodes are requested [2], two processes (engines) run on each node [ppn=2], and each process will use two cores [cpp=2]. The hosts allocated to this job may not be used for any other jobs while this job is running [#excl].
- The **-machinelist num=4** option indicates that this is a DSO job and that a total of four engines will be started. This option is required for all batch jobs.

Related Topics

Integration with PBS (Portable Batch System) Professional from Altair Engineering

qsub Arguments

Monitoring Ansoft PBS Batch Jobs

Example PBS qsub Command Lines

Integration with Grid Engine (GE)

Before Sun was acquired by Oracle, this job scheduler was an open source product, and it was known as Sun Grid Engine (SGE). Since the acquisition, the product has been renamed Oracle Grid Engine (OGE), and new versions are expected to be closed source versions. The Open Grid Scheduler project hosted on SourceForge plans to continue maintaining an open source version. Any of the versions may also be called Grid Engine (GE).

The Grid Engine scheduler is only supported on Linux. With GE, jobs may be submitted in any of the following ways:

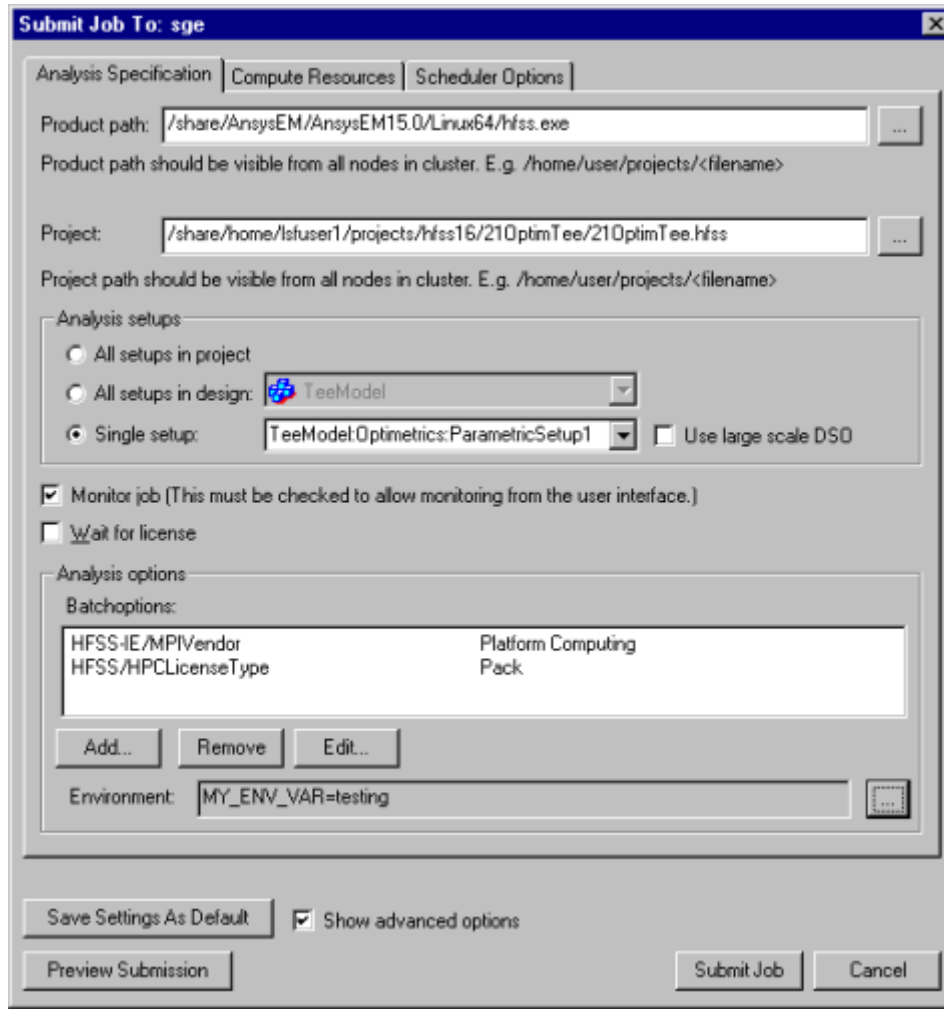
- Using SGE commands (qsub, etc.) or the SGE gui (qmon)
- Using the generic scheduler GUI in local mode
- Using the generic scheduler GUI in service mode

The ANSYS Electromagnetics Suite 15.0 release has been tested with the following versions of GE:

- SGE (Sun) 6.2u6
- OGS (Oracle) 2011.11
- Univa: 8.1.3

ANSYS Electromagnetics products support Grid Engine (GE) for Serial analysis, Multi Processing and Distributed Analysis. Models with parametric sweeps can use Large Scale DSO. With GE, the ANSYS EM job doesn't require graphics. ANSYS EM job's progress can be monitored through SGE commands or through the dialog opened through **Tools>Job Management>Monitor Jobs...**.

Besides the command line interface, you can also use a Job Management user interface to submit jobs



Related Topics

Job Management User Interface for SGE

Command Line Enhancements for ANSYS EM Desktop Products

Installation of ANSYS EM Tools on SGE

Windows:

Install on every node of cluster

Setup 'temp directory' to a path that is same on all nodes. For example, c:\temp

LINUX:

Install on a single node, on a shared drive.

Setup 'temp directory' to a path that is same on all nodes. For example, /tmp

Ensure that the product is available using the same path on all nodes

Permissions:

All users of the cluster should have read/write permissions to temp directory

All users should have read/execute permissions to installation directory

When a desktop scheduler GUI is run the same node as the job submission node, no other configuration is necessary: installation is sufficient. You select the scheduler through the desktop GUI. You need to ensure that scheduler commands are available in the path before you launch desktop.

Note There is no need to install RSM unless the you are using the scheduler GUI on a post processing node that is different than the than the job submission node. In this case, RSM must be configured with the scheduler type and path.

A post processing nodes is a node in the cluster that can run the ANSYS Electromagnetics desktop in graphical mode. A job submission node is a node in the cluster in which job submission commands are available.

Turn OFF firewall between cluster nodes.

Scenario 1: The post-processing node and job-submission node roles are served by distinct machines.

In this case, perform the following configuration:

The job-submission node should be configured to run RSM service, which serves as a proxy to scheduler. The RSM Service should be running as 'root' in order to facilitate jobs running using the credentials of the job's owner. A **configuration file in the RSM installation folder should be edited** to specify information regarding the scheduler that manages jobs on this cluster. A block labeled 'Scheduler' must be included within the 'AnsoftCOMDaemon' block. This block contains two string entries:

- SchedulerName: this contains the unique part of the scheduler proxy library name
- ConfigString: this contains a scheduler specific configuration string

The case of the SchedulerName string is significant on Linux because Linux file names are case sensitive. The case of the SchedulerName string is not significant on Microsoft Windows. In ANSYS Electromagnetics Suite 15.0, the possible scheduler names are: lsf and sge. The ConfigString entry is a scheduler specific configuration string, described below.

In addition, the AnsoftRSMService must be started with appropriate environment variables set. Generally, the environment variables must be set the same as they would be set for using the scheduler via command lines.

SGE Details

For SGE, the ConfigString entry must contain the search path for the SGE commands. It may contain a single directory, the directory containing the SGE commands. Alternatively, it may be a path,

with directories separated by the colon character ":", where the SGE command directory appears before any other directory containing files with the same name as any SGE commands.

Example ansoftrsmservice.cfg configuration file:

```
$begin 'AnsoftCOMDaemon'
   $begin 'Managed COM Servers'
   $end 'Managed COM Servers'
   $begin 'Scheduler'
   'SchedulerName'='sge'
   'ConfigString'='/opt/sge6.2u4/bin/lx24-amd64'
   $end 'Scheduler'
$end 'AnsoftCOMDaemon'
```

Scenario 2: The post-processing node and job-submission node roles are served by the same machine.

The **Select Scheduler...** command (as described in the Job Management User Interface for SGE section) is used to gather details about the scheduler. In this case, the Desktop process should be started in an environment suitable for submitting jobs to the scheduler.

The environment should be configured so that all SGE commands are found using the standard search path. In particular, search for the following commands in the search path should result in the SGE command being found: "qsub", "qdel", "qstat", and "qconf". No other command with the same name should appear before the SGE command in the search path.

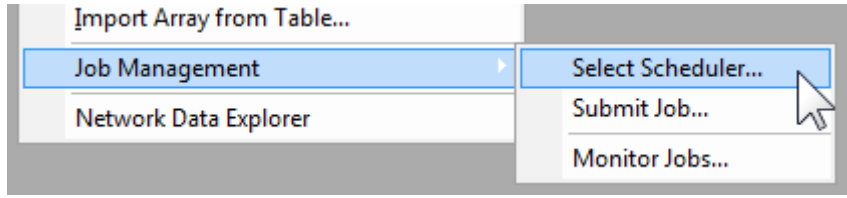
Related Topic

Job Management User Interface for SGE

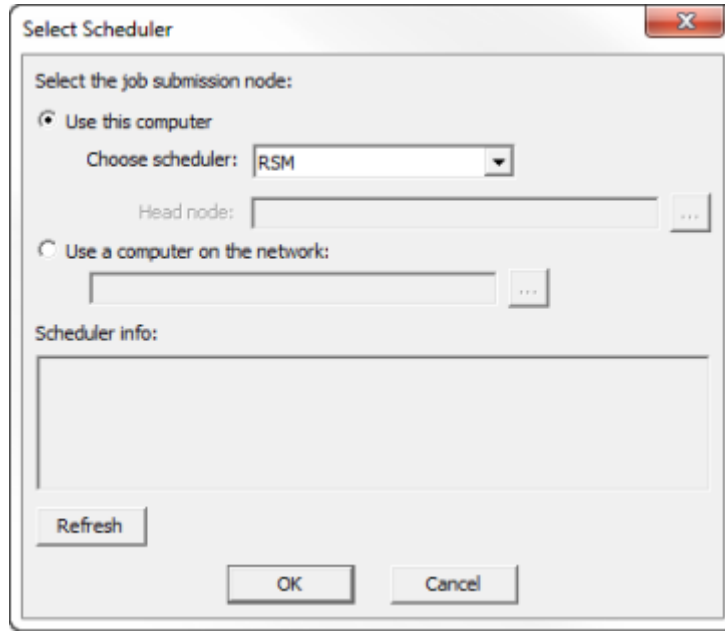
Command Line Enhancements for ANSYS EM Desktop Products

Job Management User Interface for SGE

The Job Management UI is accessed by running ANSYS Electromagnetics product Desktop on the designated 'Postprocessing node' of the cluster. The Desktop provides UI commands for Scheduler selection, Job submission and Job monitoring/control. You access the Scheduler User Interface by clicking **Tools>Job Management>Select Scheduler**.



Before you can use **Submit Job**, you must click **Select Scheduler** as the one-time initial step.This opens the **Select Scheduler** dialog.



Specify the following parameters:

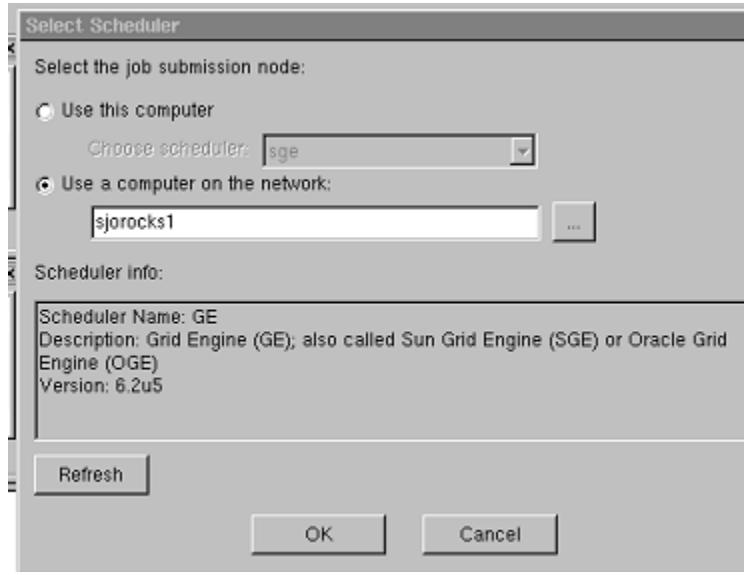
- Job submission node: this is the node on the cluster where scheduler commands (such as SGE's qsub) are allowed to run.

  Choose **Use this computer** if scheduler commands are enabled on the post-processing node.

  Choose **Use a computer on network** if the cluster is configured in a manner as to disallow job-submission from the post-processing node. Specify node name appropriately.

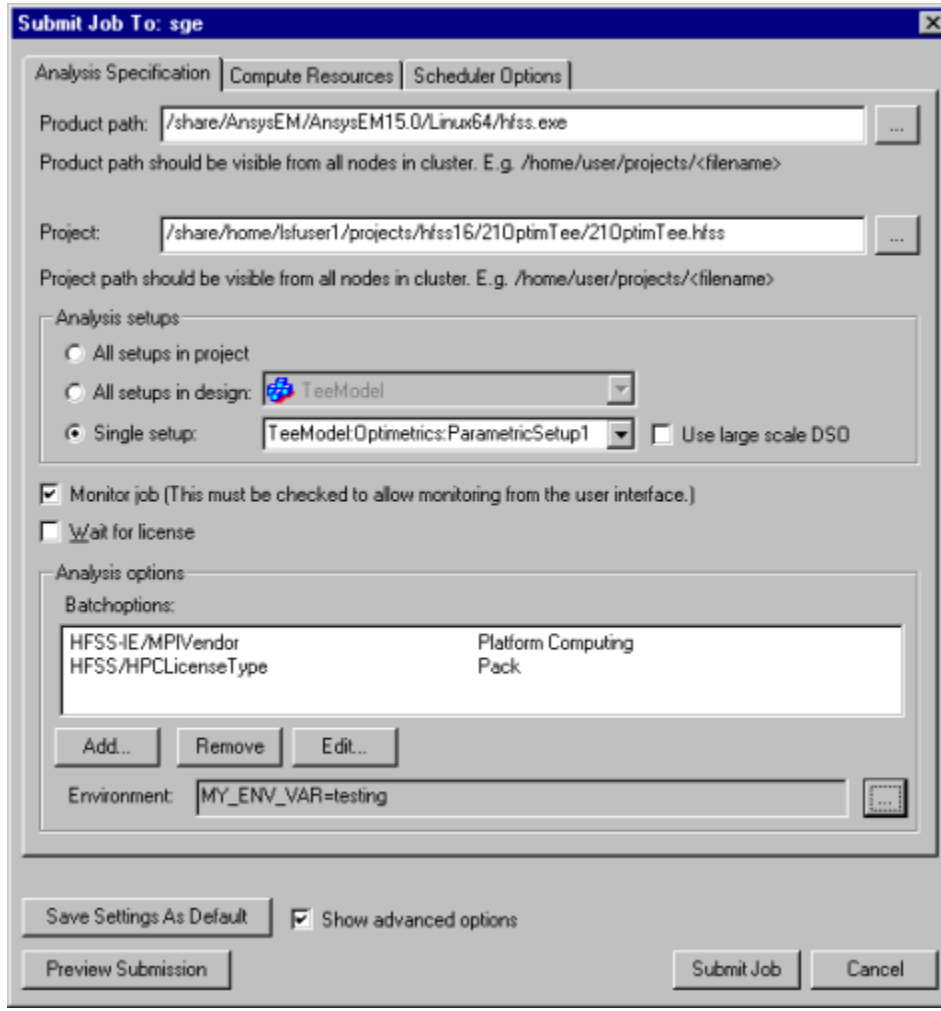
  Pre-requisites: For this choice to work, the job-submission node must already be configured with a running RSM service, as documented in Installation of ANSYS EM Tools on SGE.
- Scheduler: Available choices, depending on your installations, are: RSM, lsf, Windows HPC, and SGE. It is also possible for you to integrate their custom scheduler into this UI, through a scheduler proxy. When this is done, more choices will be available in the combo-box, one per custom scheduler proxy that is deployed in the installation

After specifying the job submission node, you can click **Refresh**. The scheduler information is then listed in the Scheduler info text field.



Once you select a scheduler, you can access the interface for job submission, monitoring and control. Click OK to close the dialog.

You access the Job submission UI by clicking **Tools>Job Management>Submit Job...**. This command launches a multi-tab dialog.



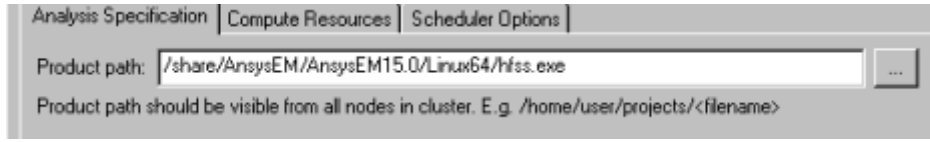
- The **Analysis Specification** tab has parameters to specify the product path, project model, the analysis setup and analysis options (including batchoptions) that affect analysis algorithms.
- The **Compute Resources** tab specifies the amount of compute resources and how to select specific resources from the available pool (for example, ParallelEnvironment is an SGE parameter).
- The **Scheduler Options** tab has analysis-independent parameters specific to the job, such as name, priority.

Analysis Specification tab for SGE

This tab lets you specify the following:

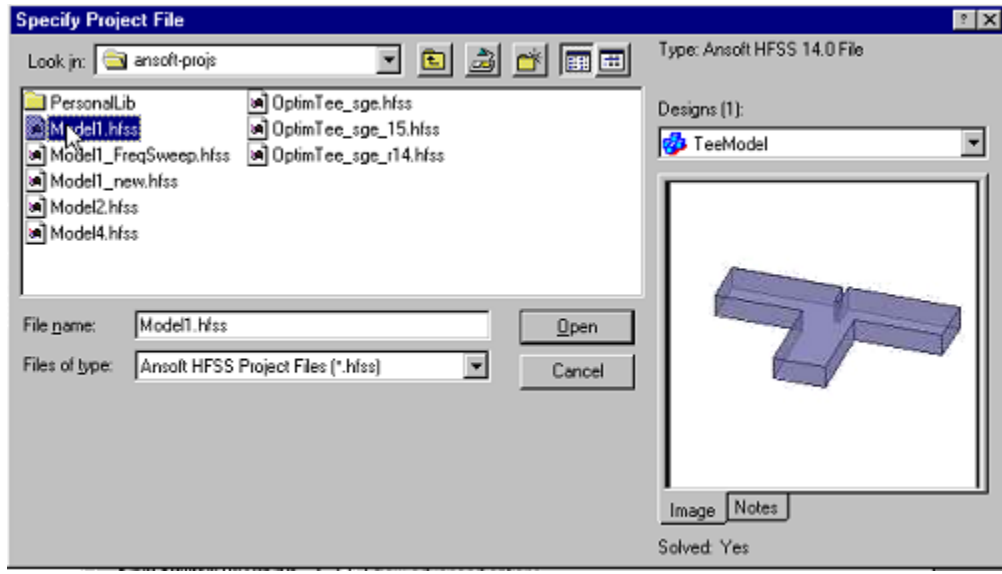
Product Path:

You can specify the product path.



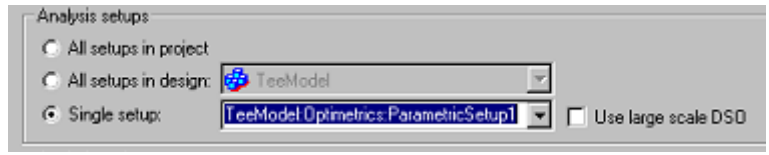
Project:

You can use the ellipsis button [...] to use a navigation window to browse. The path should be visible to all nodes in the cluster.



Analysis Setups:

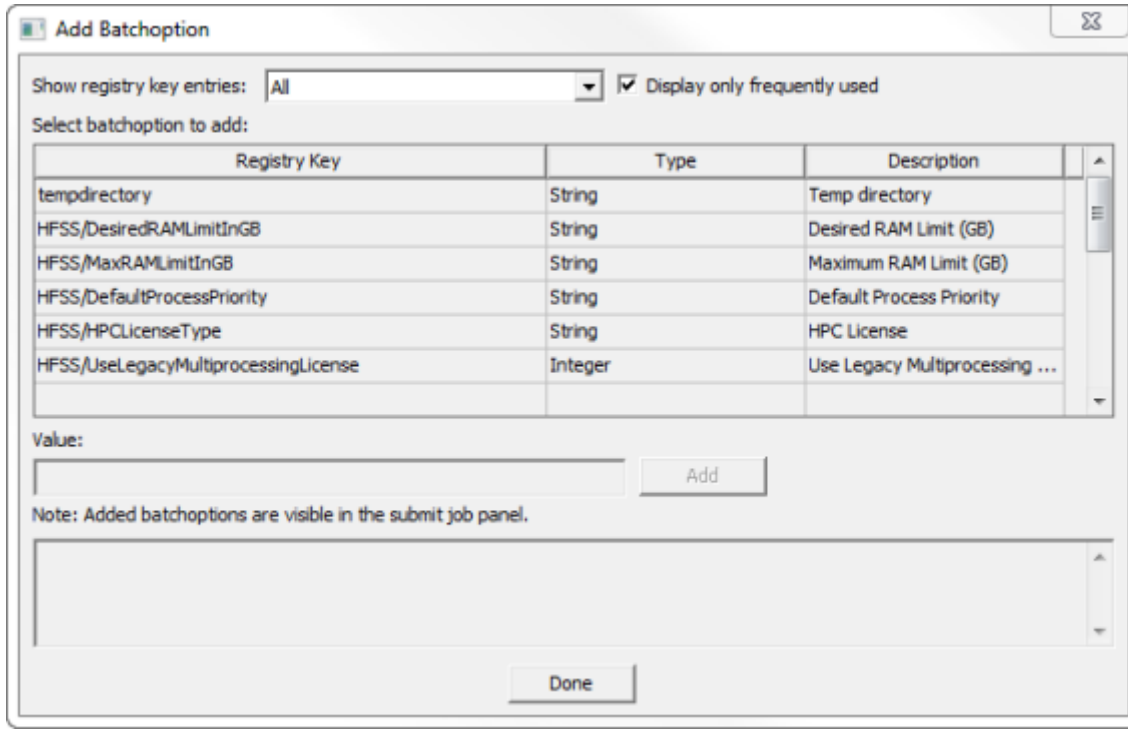
Specify All setups in the project, all in a design, or a single setup. If the setup includes a parametric sweep, the **Use large scale DSO** checkbox is enabled.



Analysis Options:

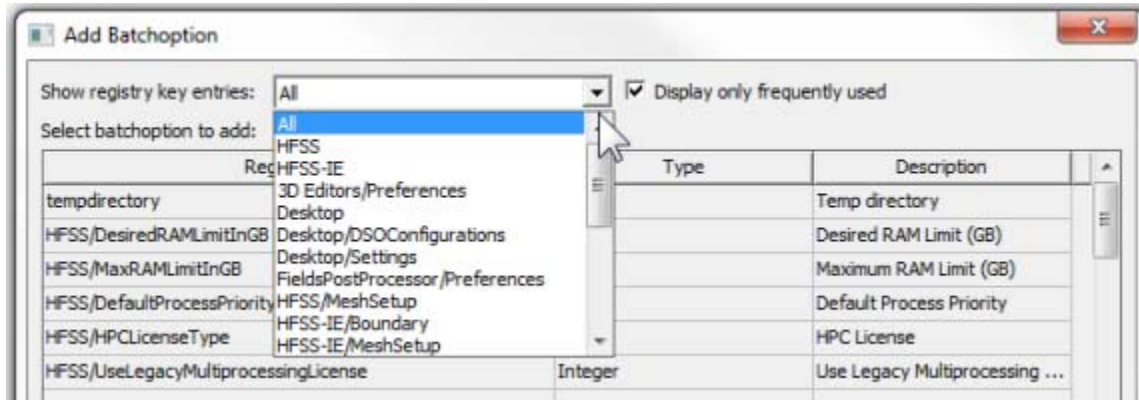
- You have checkboxes for whether to Monitor Job through the GUI, and whether to Wait for license.

- You can also specify Batchoptions. Click the **Add...** button to open a dialog for selecting the Batchoptions.

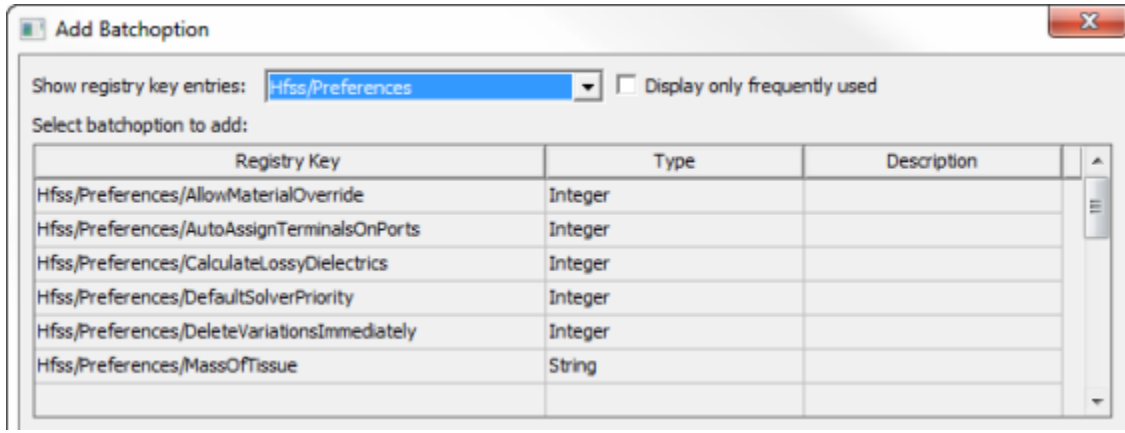


The lower Value field shows the legal values for the selected registry key. You can type the desired value into the upper text field under value. Click the **Add** button to accept the selection of the registry key with the specified value. Click **Done** close the **Add Batchoption** dialog.

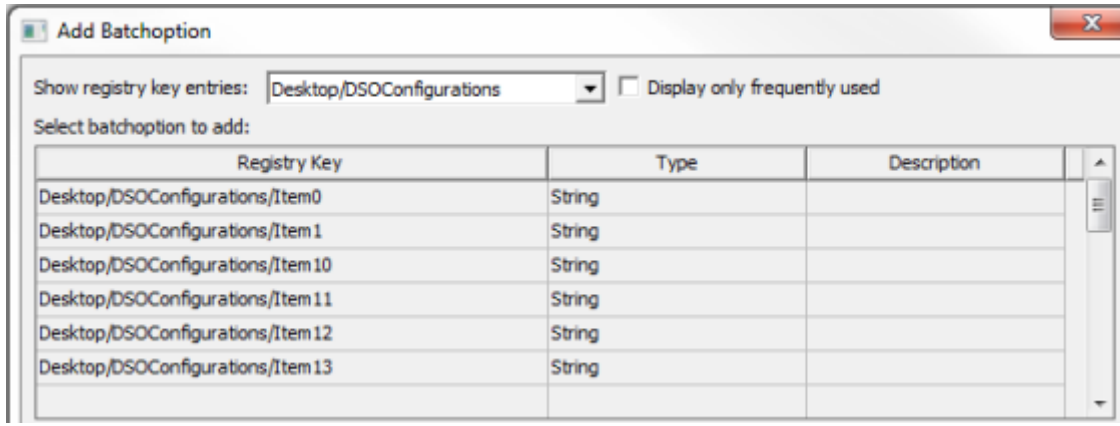
At the upper left, a drop down menu lets you specify which registry key categories to display, whether All, or selected category.



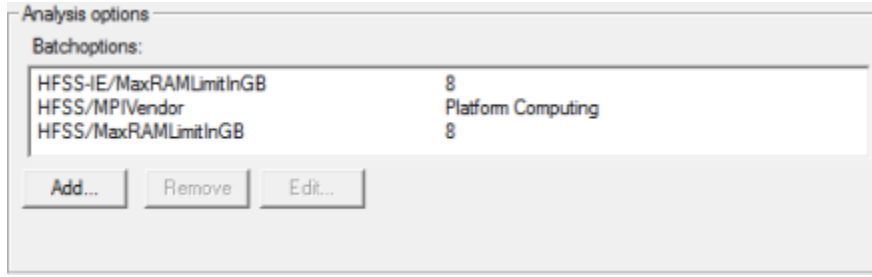
A checkbox lets you choose between displaying only frequently used entries (the default), or by unchecking, all options available for the selected group.



Note that for Large Scale DSO problems, you also have the option to specify predefined configurations.

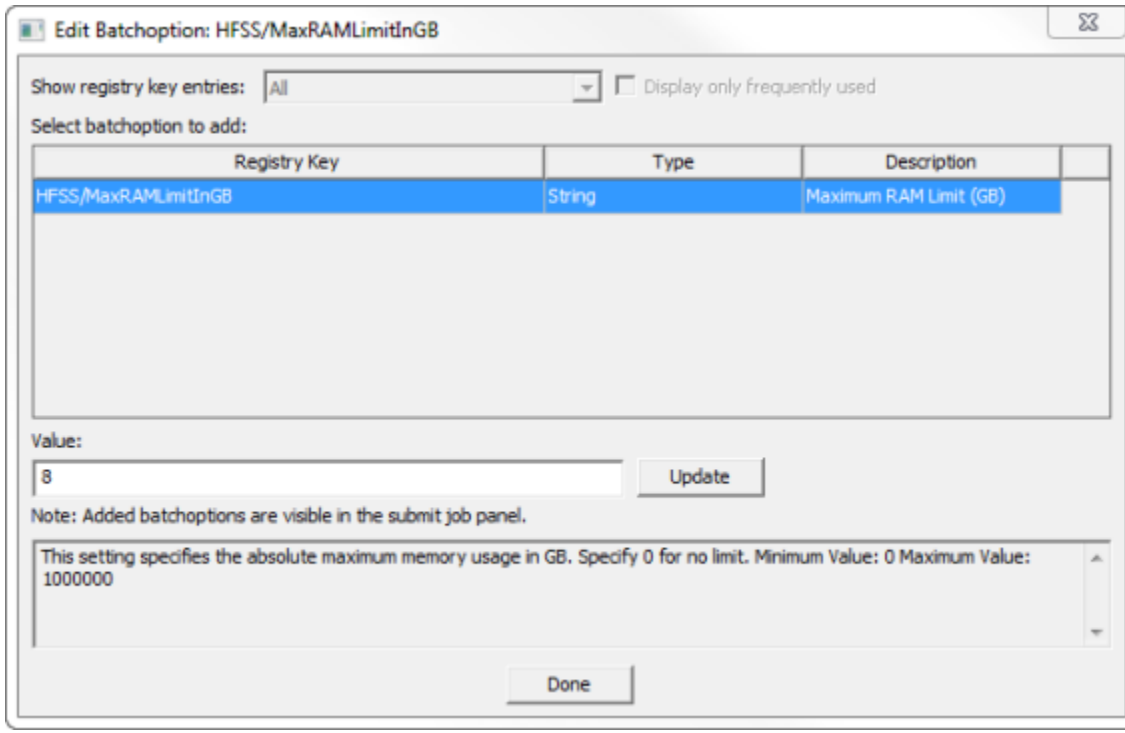


The added registry keys and values are listed in the Batchoptions field of the **Submit Job** dialog.



Selecting from the list enables buttons for removing or editing registry key values. Selecting a

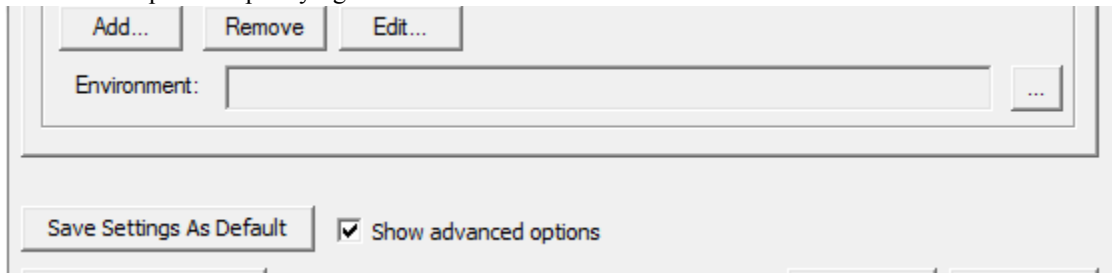
registry key and clicking **Edit...** opens the **Edit Batchoptions** dialog.



Here you can edit the Value field and update the value for the selected registry key.

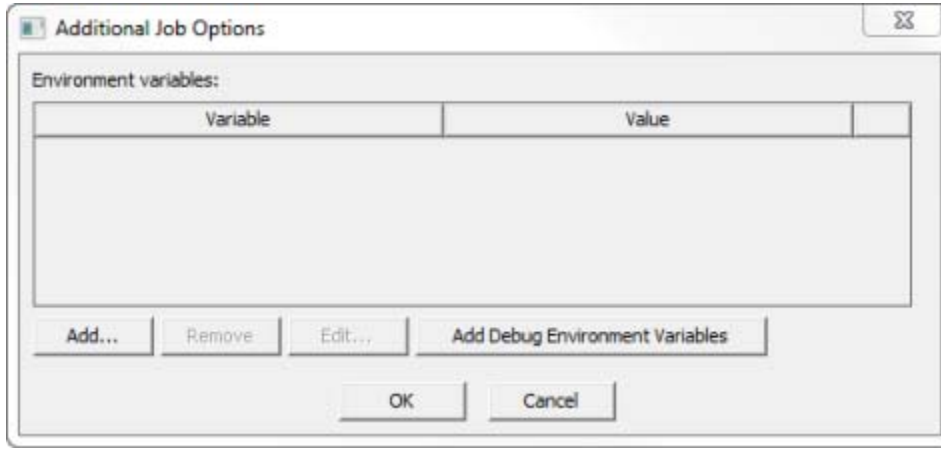
Environment:

If you check the Show advanced options check box, you can see the Environment field. This permits specifying Environment Variables.

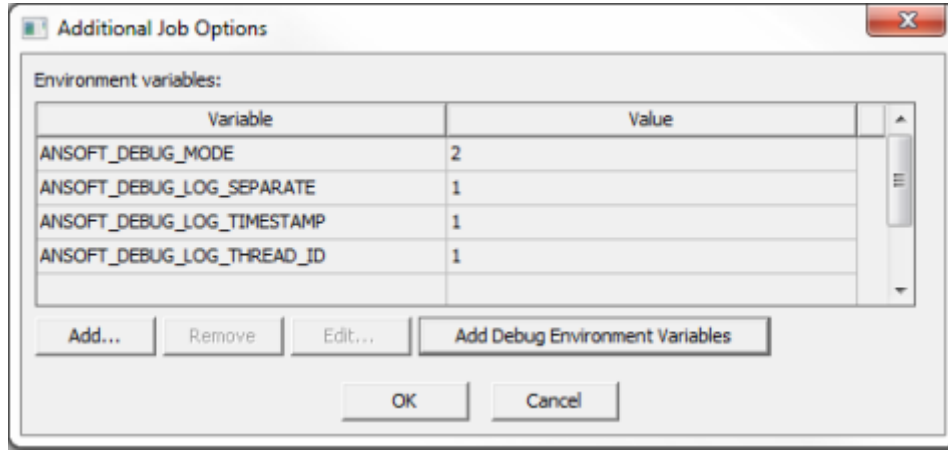


Click the ellipsis button [...] by the Environment field to open the **Additional Job Options**

dialog.



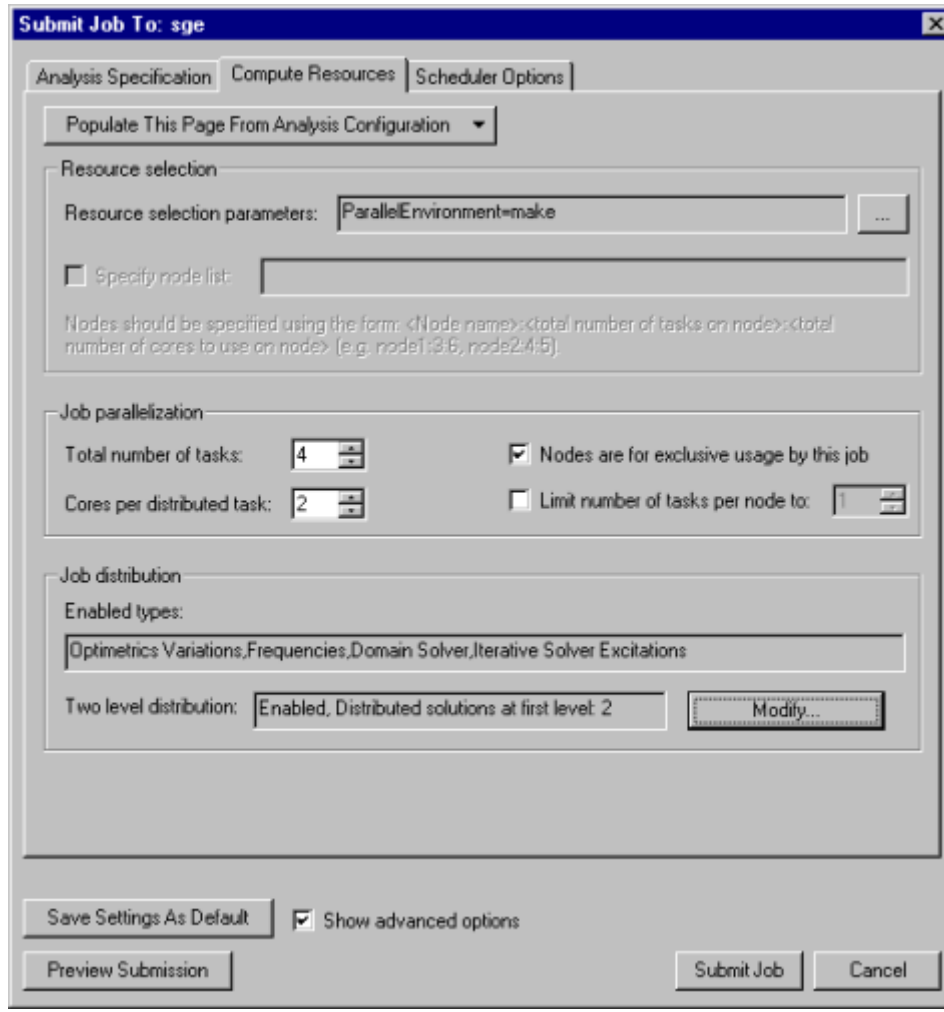
The **Add** button permits you to specify your environment variables. Once, added, you can select **Remove** or **Edit**. The **Add Debug Environment Variables** button adds variables of use in working with support.



- The **Save Settings as Default** button lets you save a current set of values as defaults the next time you invoke the Scheduler GUI. This can simply subsequent job submissions.

Compute Resources Specification Tab

This tab lets you provide Resource selection and Job parallelization parameters. Once you have specified parameters, you can Save Settings as Defaults, Preview a Submission, and Submit a job.



The **Populate This Page From Analysis Configuration** drop down menu lists all Analysis configurations that you have previously defined. See Editing Distributed Machine Configurations.

You can also populate this page yourself by specifying Resource selection, Job Parallelization parameters, and Job distribution parameters.

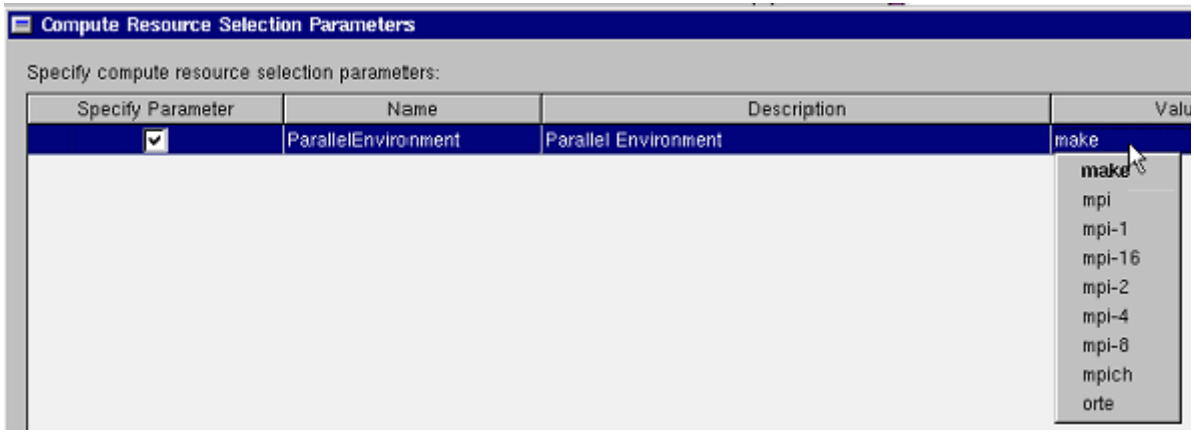
Resource Selection:

If you do not specify parameters for resource selection, SGE may submit jobs to any machine in the entire pool that is available.

- Resource selection parameters.

  Clicking the ellipsis button [...] opens a dialog for parameters specific to SGE, in this case **Par-**

allelEnvironment. The Specify Parameter checkbox in the dialog enables a parameter, and you can select the Value for the ParallelEnvironment parameter from a drop down menu.



If you don't specify a parameter, the scheduler handles the situation.

- Specify node list

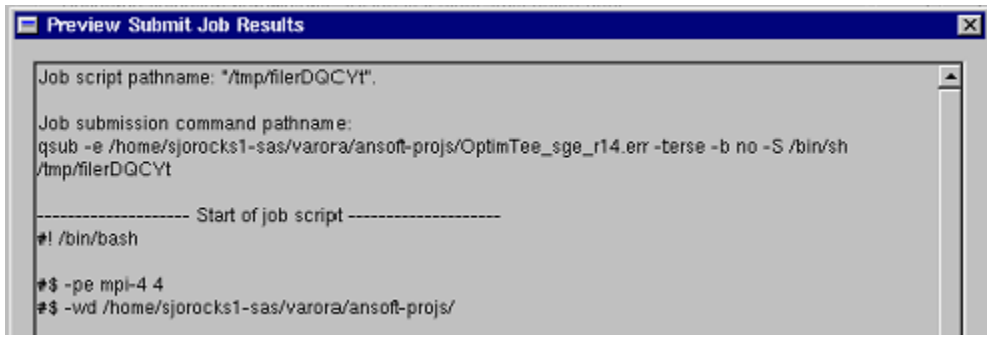
  Checking Specify node list enables the field for specifying a node list. In a computing environment where the available cores are not uniform, you can use this to have control over which resources your job will use. If your Analysis configuration contains a node list, you can use Populate this Page from Analysis Configuration.

Job parallelization:

The Job parallelization fields let you specify

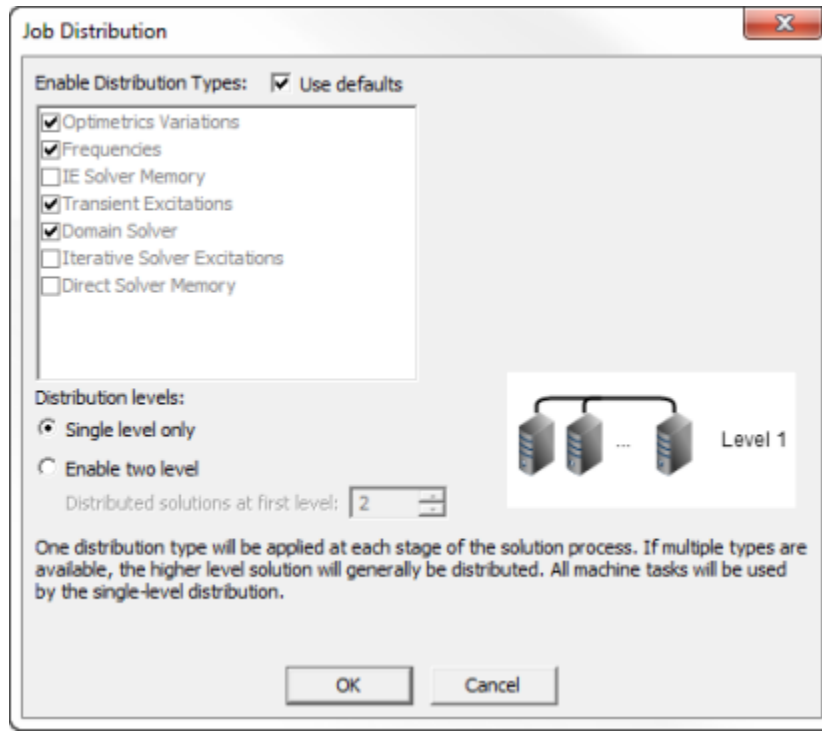
- Total number of tasks:
- Cores per distributed task.
- Whether nodes are for exclusive usage by this job
- Whether to limit the number of tasks per node to a value.

In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.



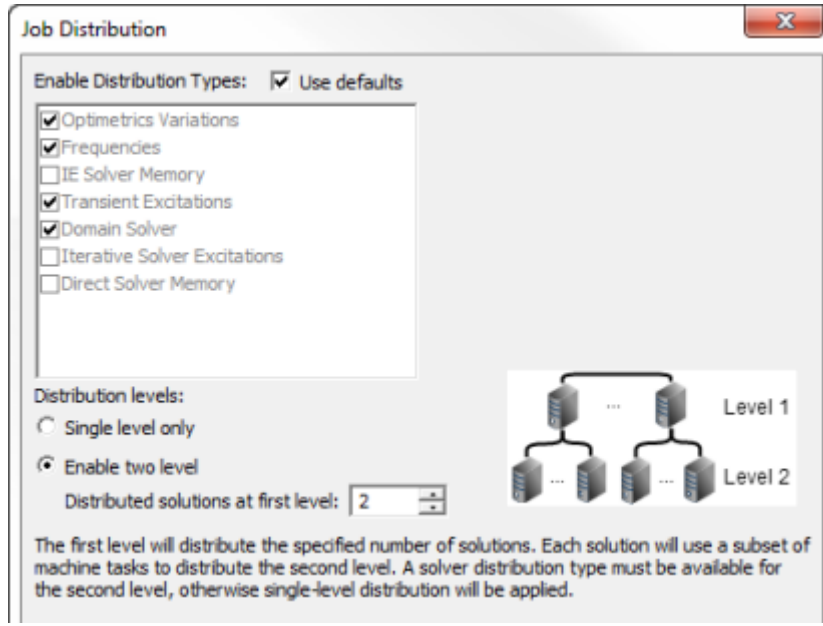
Job Distribution

- Enabled types, such as Variations, Frequencies, Transient Excitations, Solver Domains, Direct Solver and Iterative Solver.
- Two level distribution, which may be disabled. Click the **Modify** button to display the **Job Distribution** dialog.



Enabled Distribution types can modified here.

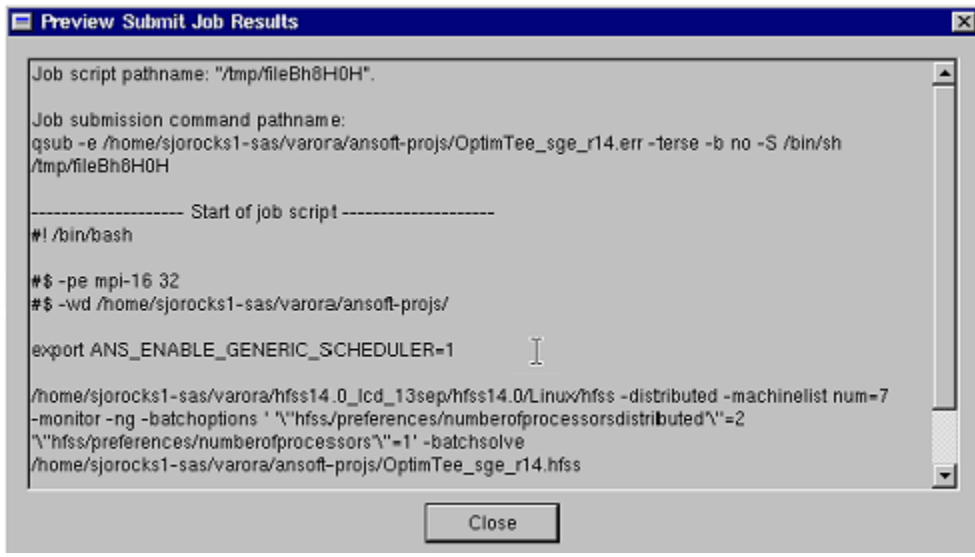
Second level distribution operates within DSO. If available and enabled you can specify a number of engines for level 1.



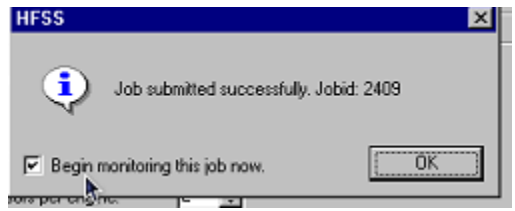
In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines, with two processors per engine, and also limit the number of engines per node to 4, the scheduler may increase the number of cores used in order to meet the limit specified for engines per node. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.

Preview submission

This opens a screen showing the qsub command to be used to submit the job.

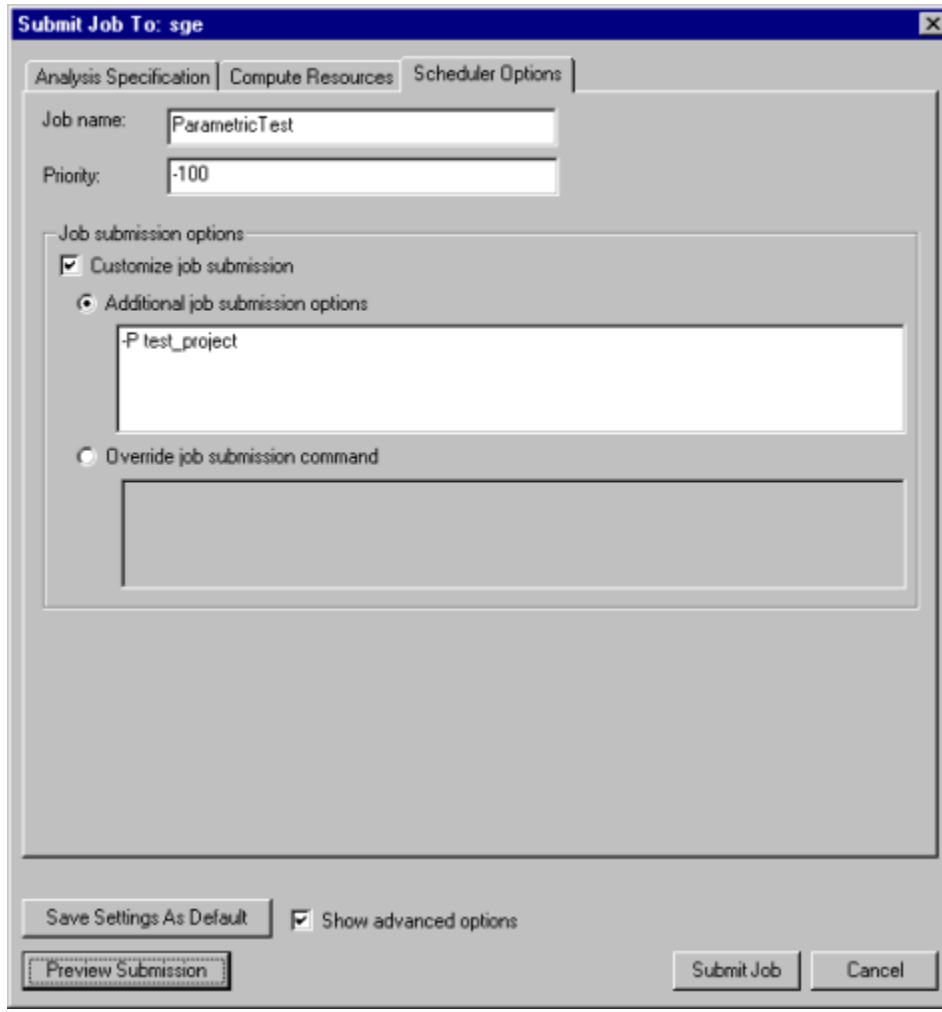


Submit job actually sends the batch command to the SGE cluster.A dialog reports a successful submit and presents a checkbox for monitoring.



Scheduler Options Tab

This tab lets you give Job name and Priority.



If you check **Show advanced options**, you can also specify Job submission options.

When the "Override job submission" radio button is checked, the user specified options replace most of the job submission options, whereas when the "Additional job submission options" radio button is checked, the user specified options are appended to the bsub command.

Text in the enabled field is appended to the bsub command. You can see the effects of any custom additions by clicking **Preview Submission**.

You continue to have the **Save Settings As Default**, **Preview Submission**, and **Submit Job** buttons.

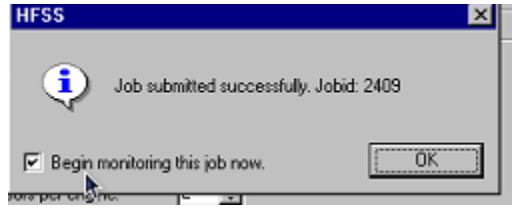
Related Topics

Integrated Job Monitoring for Job Management Interface for SGE

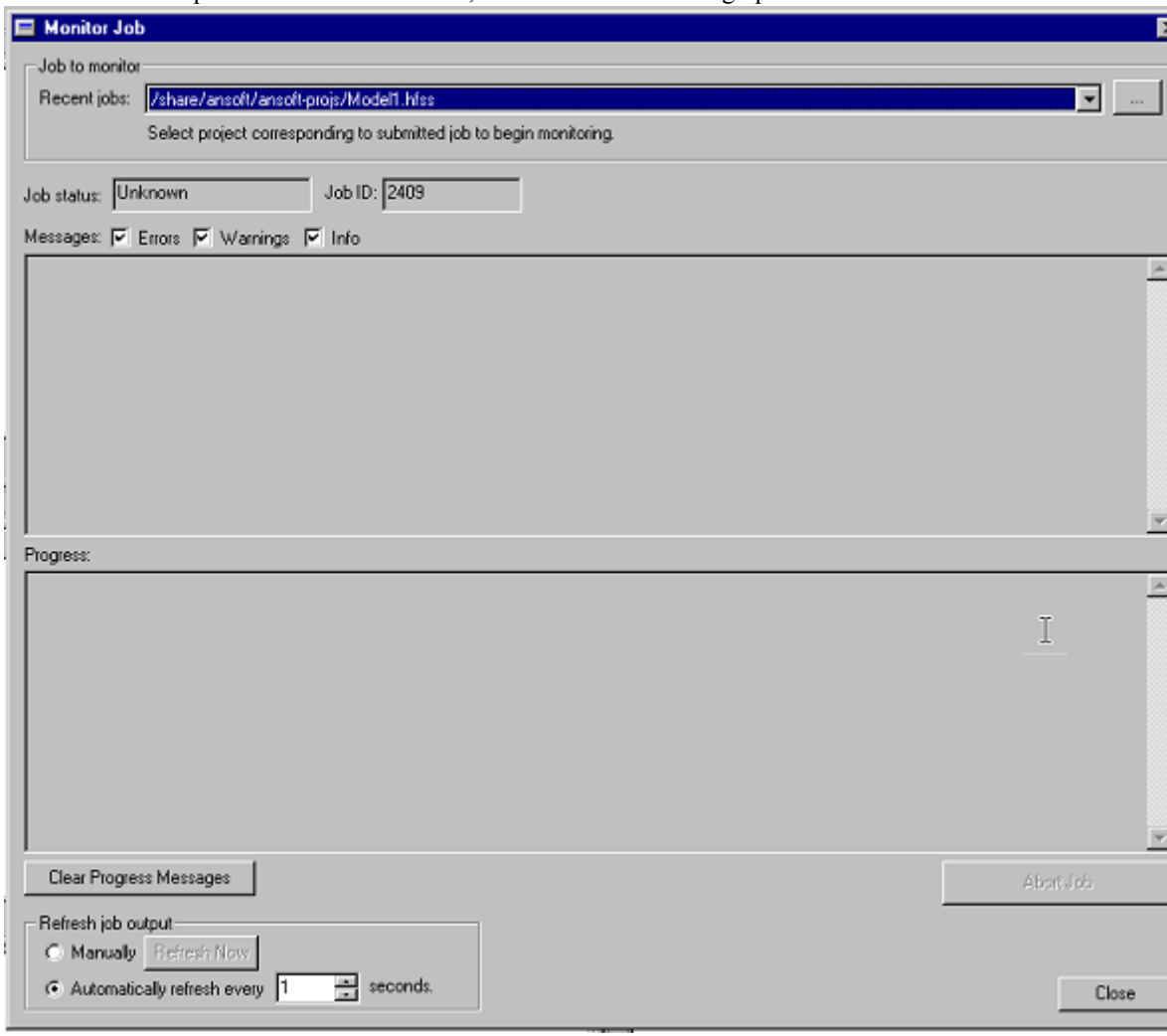
Scheduler Proxy Interfaces

Integrated Job Monitoring for Job Management Interface for SGE

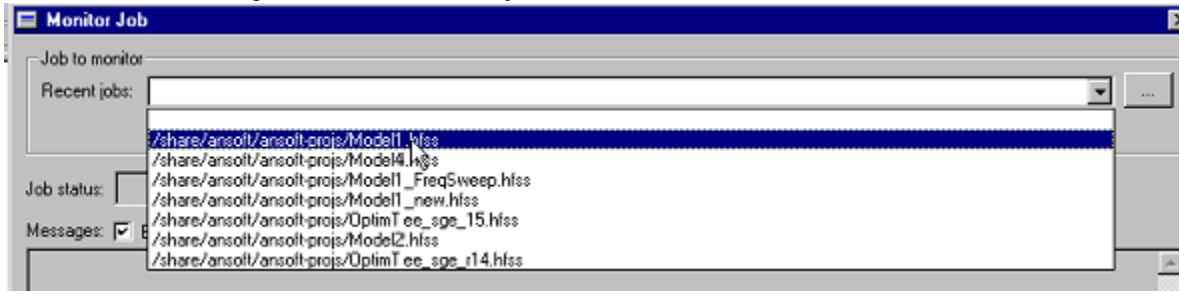
The job monitoring/control dialog is launched through the command **Tools>Job Management>Monitor Jobs...** or by checking **Begin monitoring this job now** in the information window reporting successful job submission.



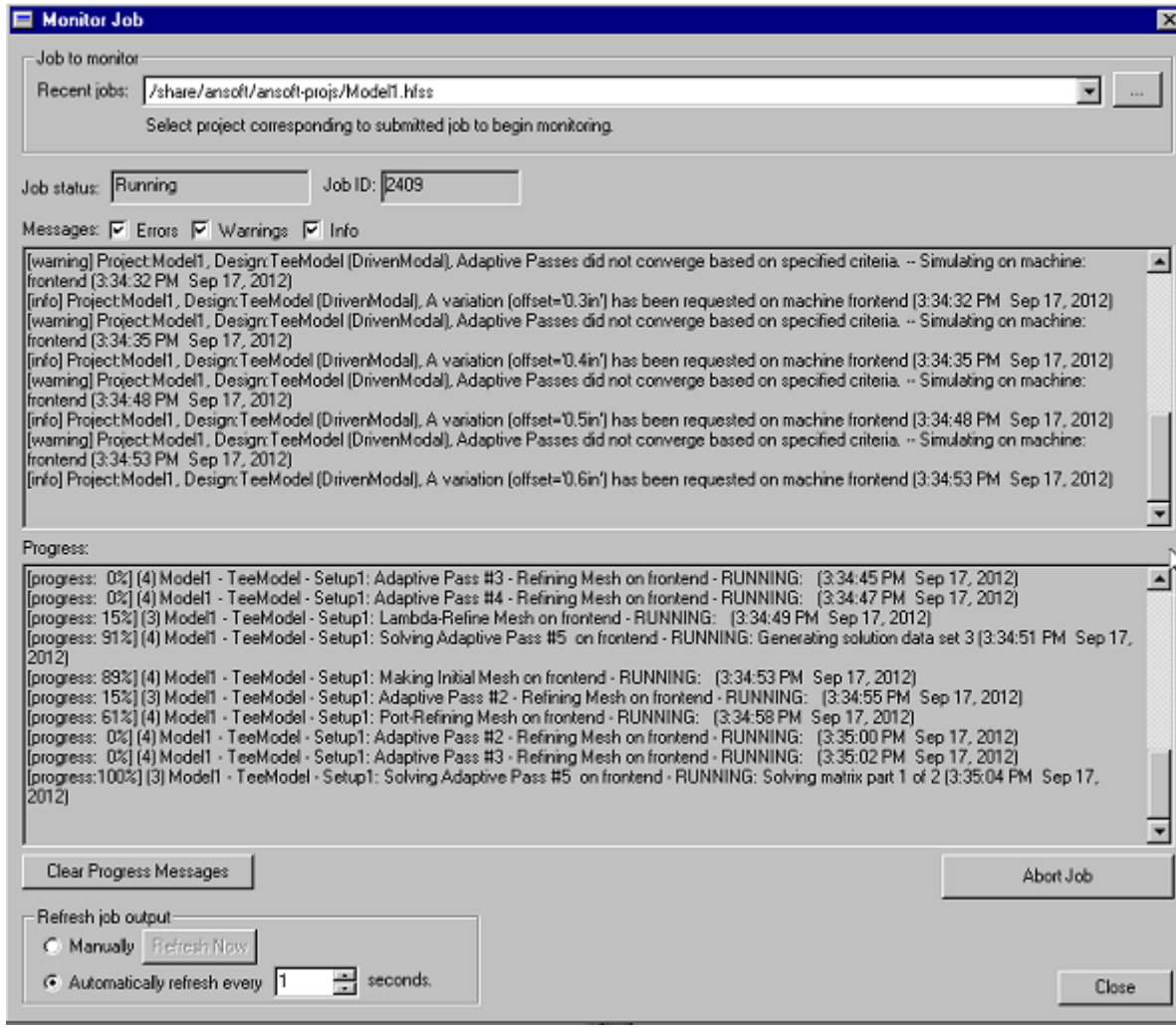
In response to either invocation, the **Monitor Job** dialog opens:



In this dialog, user selects the same project they submitted. You begin by selecting the job to monitor from a pull down list of recent jobs.



The lower left has corner of the dialog has options for manual refresh or to Automatically refresh every specified number seconds. After refresh, the messages for the job are displayed. The format of the text is essentially same as output of the '-monitor' command-line option.



If desired, you can use the check boxes to filter the messages listed in terms of whether to monitor Errors, Warnings, or Info messages.

The **Abort Job** button lets you abort a job.

Related Topics

Scheduler Proxy Interfaces

Job Management User Interface for SGE

SGE Commands for Information About Jobs and Cluster Configuration

The following SGE commands are especially useful for getting information about the cluster configuration or for getting information about running or completed jobs. This list only contains a few of the most common commands. Consult the SGE man pages for a complete list and more details.

`qconf -help`: The first line displays the SGE version

`qacct -j` *job-id* : Displays a log of the completed job with id job-id (if accounting is enabled)

`qstat -j` *job-id* : Displays a log of the running job with id job-id

`qconf -sc`: Show all complex attributes

`qconf -spl`: Show a list of all parallel environments

`qconf -sp` *pe-name* : Show details of parallel environment named pe-name

`qconf -sql`: Show a list of all queues

`qconf -sq` *queue-name* : Show details of queue named queue-name

`qconf -sconf`: Show configurations

Submitting ANSYS EM SGE Batch Jobs via the Command Line

The SGE qsub command may be used to submit ANSYS EM jobs. Typical command formats are:

`qsub` *qsub_args ansysEM_exe ansys_args*

`qsub` *qsub_args job_script*

`qsub` *qsub_args* `[ -]`

where:

- *qsub_args* are the options of the qsub command,
- *ansysEM_exe* is the pathname of the ANSYS EM tool executable to launch,
- *ansys_args* are the arguments to the Ansoft tool command, and
- *job_script* is a shell script containing the ANSYS Electromagnetics desktop command to run.

In the first format, the ANSYS EM desktop command and its arguments are specified on the **qsub** command line. In the second format, the pathname of a shell script containing the ANSYS EM desktop command and its arguments is specified on the **qsub** command line. In the third format, the command is omitted or replaced with a hyphen; this indicates that the command or script will be taken from stdin.

Quoting ANSYS EM Command or Arguments for SGE

If the ANSYS EM tool executable pathname (*ansysEM_exe*) or any of the arguments of the ANSYS EM tool command (*ansysEM_args*) contain characters which are interpreted by the command shell, then these special characters must be properly quoted to ensure that the correct command is launched by SGE. This is especially important when using the first form of the **qsub** command, as the ANSYS EM desktop command is processed by the shell twice in this case. It is processed by the shell when the **qsub** command is processed, and again when the job is started.

Serial SGE Batch Jobs

In general, ANSYS EM batch jobs may be submitted as SGE serial jobs without any special considerations.

See Monitoring ANSYS EM SGE Batch Jobs for options for monitoring ANSYS EM batch jobs.

Parallel SGE Batch Jobs

When an ANSYS EM batch job is run as an SGE parallel job, the SGE scheduler will select the hosts for the distributed analysis job, and start the desktop process on one of these hosts. The desktop process will obtain the list of hosts from the SGE scheduler, and start analysis processes, as needed, using the SGE scheduler facilities. To run an SGE parallel job, the job must be submitted to an SGE parallel environment (PE).

If the qmaster tcp port is not configured as a service, but rather via the environment variable SGE_QMASTER_PORT, this variable must be set in the ANSYS EM batch job environment. This is needed because the ANSOFT EM desktop uses the "qrsh -inherit" command to launch engine processes.

See Monitoring ANSYS EM SGE Batch Jobs for options for monitoring Ansoft batch jobs.

Setting Up an SGE Parallel Environment (PE)

To allow ANSYS EM batch jobs to distribute analysis engines to multiple hosts, the job must be run in a parallel environment (PE) in which the control_slaves parameter is set to TRUE. This setting is required to allow the ANSYS EM desktop to start analysis engines on hosts other than the local host, i.e., the host where the ANSYS EM desktop is running.

Here is a sample parallel environment configuration:

```
pe_name            ans_test1
slots              999
user_lists         NONE
xuser_lists        NONE
start_proc_args    /bin/true
stop_proc_args     /bin/true
allocation_rule    $round_robin
control_slaves     TRUE
job_is_first_task  FALSE
urgency_slots      min
accounting_summary TRUE
```

The user_lists and xuser_lists parameters are ACLs (access control lists) used to control which users have permission to use the parallel environment. The user_lists setting gives permission to use the PE. The xuser_lists setting denies permission to use the parallel environment. The xuser_lists settings override the user_lists settings.

The start_proc_args and stop_proc_args parameters contain the pathname and arguments for the parallel environment startup and shutdown scripts. No startup or shutdown scripts are needed for

parallel ANSYS Electromagnetics batch jobs. The setting /bin/true may be used as the value for these scripts; this utility does nothing and returns an exit code indicating success (0).

The parallel environment allocation_rule parameter will affect how the analysis engine tasks are distributed across the hosts allocated to the job. The $round_robin setting distributes the tasks across the hosts in a round robin fashion, resulting in the load being relatively evenly distributed over all of the hosts. The $fill_up setting allocates all slots on a host before distributing the tasks to another host; the result is that most hosts are either fully utilized or completely unused. See the sge_pe man page for other settings for this parameter.

The control_slaves parameter must be set to TRUE, as described above.

The job_is_first_task parameter also affects how tasks are allocated. When submitting a job to run in a parallel environment, the number of parallel tasks, n, is specified on the command line. If this setting is TRUE, then the job process is considered one of the tasks, and only (n-1) additional tasks are allocated to the job. If the setting is FALSE, then the job process is not considered to be one of the tasks, and n additional tasks are allocated for the job.

See the sge_pe man page for more information about these and other PE parameters.

A parallel environment does not run tasks directly. Instead, the tasks are distributed to queues associated with the parallel environment. In order to complete the setup of a parallel environment, one or more queues need to be associated with the parallel environment. The queue pe_list parameter is used to specify the parallel environments (PEs) supported by the queue. This is an important step; **if no queues support a given PE, then jobs submitted to that PE will not run**.

Parallel Batch Job Command Line Considerations

The number of engines run on a host will depend on the total number of distributed engines, and the number of hosts allocated to the job. The memory required on a host depends on the number of engines running on the host and on the memory needed for each engine. The **qsub** command **-l resource=***value*,... or **-q** *queue_list* command line options specify that the parallel batch job run on machines with sufficient memory and other resources.

Related Topics

Monitoring ANSYS EM SGE Batch Jobs

ANSYS Electromagnetics desktop -monitor Command Line Option for SGE

Example SGE qsub Command Lines

Issue with qrsh (SGE)

Issue with MainWin Core Services for SGE

What a Scheduler Does

Recommended Practices for SGE Clusters

Scheduler Proxy Interfaces

Monitoring ANSYS EM SGE Batch Jobs

The suggestions below may be used for SGE serial jobs and for SGE parallel jobs.

SGE qstat Command

The SGE **qstat** command may be used to display information on jobs and queues. If the **-j [***job_list***]** option is included, then information on jobs is displayed. If the **-j [***job_list***]** option includes a job list, then the displayed information is limited to the jobs in the job list.

The **-u** *user*,... option limits the output to jobs associated with users in the user list. If the **-u** *user*,... option is not specified, then information on queues or jobs of the current user are displayed.

The **-t** option displays extended information about the subtasks of each displayed job. This is equivalent to the **-g t** option. The **-r** option displays extended information about the resource requirements of the displayed jobs.

See the SGE manual pages for more information.

Related Topics

Integration with Sun Grid Engine (SGE)

Job Management User Interface for SGE

Integrated Job Monitoring for Job Management Interface for SGE

ANSYS Electromagnetics desktop -monitor Command Line Option for SGE

Example SGE qsub Command Lines

Issue with qrsh (SGE)

Issue with MainWin Core Services for SGE

ANSYS EM Desktop -monitor Command Line Option for SGE

The **-monitor** command line option enables batch job output to the standard output and standard error streams. The warning, info, and progress messages are sent to the standard output stream. The error and fatal messages are sent to the standard error stream.

The SGE scheduler redirects the standard output and standard error streams of batch jobs to files specified in the **qsub -o [[***hostname***]:]***path*,... and the **-e [[***hostname***]:]***path*,... command line options, respectively. If either option is not specified, then the associated stream is redirected to the default file pathname.

The **qsub -j y[es] | n[o]** controls whether the standard output and standard error streams are merged. If the y or yes value is specified, then the standard error stream is merged into the standard output stream. If the **-e** *host_and_path* option is also specified in this case, the *host_and_path* setting is ignored. If the n or no value is specified, or if this option is not specified, then the standard error stream and standard output stream are not merged.

You can monitor the progress of a job by checking the standard output file for progress, info and warning messages, and checking the standard error file for error and fatal messages.

Related Topics

Integration with Sun Grid Engine (SGE)

Monitoring ANSYS EM SGE Batch Jobs

Example SGE qsub Command Lines

Recommended Practices for SGE Clusters

Issue with qrsh (SGE)

Issue with MainWin Core Services for SGE

Example SGE qsub Command Lines

All of the following examples show how to submit Linux HFSS jobs on SGE, but similar command lines will work for all ANSYS Electromagnetics products.

Serial job using command line:

```
qsub -b y /opt/AnsysEM/AnsysEM15.0/hfss -ng -BatchSolve
~/projects/OptimTee.hfss
```

- The **-b y** option indicates that hfss is launched directly from the command line, instead of using a script.
- No queue is specified, so the default queue will be used

Serial job with a hard runtime limit of 15 minutes:

```
qsub -b y -l h_rt=00:15:00 /opt/AnsysEM/AnsysEM15.0/hfss
-ng -BatchSolve ~/projects/OptimTee.hfss
```

- The -l h_rt=00:15:00 option indicates that this job has a "hard" runtime limit of 15 minutes.

Serial job using a script, with a runtime limit specified in the script:

```
qsub  ~/sge/scripts/OptimTee.csh
```

- The **-b y** option is absent, so the script ~/sge/scripts/OptimTee.csh will be run when the job starts.
- The script file OptimTee.csh may contain SGE directives in addition to the command(s) to run. In this example, a directive with a hard runtime limit if 15 minutes is included in the script.

Script file contents:

```
#!/bin/csh
#$ -l h_rt=00:15:00
/opt/AnsysEM/AnsysEM15.0/hfss -ng -BatchSolve
~/projects/OptimTee.hfss
```

- The SGE directive #$ -l h_rt=00:15:00 is equivalent to including -l h_rt=00:15:00 on the qsub command line.

Distributed processing job using 4 engines:

```
qsub -b y -pe pe1 4 /opt/AnsysEM/AnsysEM15.0/hfss
-ng -BatchSolve -Distributed -machinelist num=4 ~/projects/
OptimTee.hfss
```

- The **-b y** option indicates that hfss is launched directly from the command line, instead of using a script.
- The **-pe pe1 4** command_line option indicates that this is a parallel job running under the pe1 parallel environment, and that 4 cores or processors are allocated to this parallel job.
- The "-machinelist num=n" option is now required for batch jobs.

- The **-Distributed** option indicates that this is a DSO job, so that multiple engines will be started. Because 4 cores are allocated to the job, the job will run 4 engines. The -Distributed option may now have additional options, such as includetypes=xxx, excludetypes=xxx, maxlevels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".

Related Topics

Integration with Sun Grid Engine (SGE)

Monitoring ANSYS Electromagnetics Suite SGE Batch Jobs

ANSYS Electromagnetics desktop -monitor Command Line Option for SGE

Recommended Practices for SGE Clusters

Issue with qrsh (SGE)

Issue with MainWin Core Services for SGE

Recommended Practices for SGE Clusters

The following subsections contain recommendations on how to set up an SGE cluster for efficiently running ANSYS Electromagnetics Suite serial and parallel jobs. These recommendations require the cluster administrator to make configuration changes.

Submitting Exclusive Jobs

Consumable Memory Limits

Serial Jobs in SGE

Parallel Jobs in SGE

Using Multithreading with Parallel Jobs

Submitting Exclusive Jobs

In many cases, clusters are used to run "large" ANSYS Electromagnetics Suite batch jobs. That is, these are jobs that may require a large quantity of resources, such as processors, memory, disk space, or run time. One way to ensure that the resources needed by the batch job are available to the job is to run the job in an "exclusive" mode. That is, any host running the job is not available for use by any other jobs. There is no SGE built in mechanism for specifying that a job is "exclusive". SGE is extensible, and it is not difficult to configure the cluster to allow exclusive jobs. The steps below show one way to do this. This example requires SGE 6.2u3 or later. Note that specifying a job as "exclusive" may delay the start of the job if there are not enough hosts available to run the job exclusively.

1. Use the command qconf -mcto add a new complex to the table of complexes. Recommended attributes are:
   - name : exclusive
   - shortcut : excl
   - type : BOOL
   - relop : EXCL
   - requestable : YES

- consumable : YES
- default : 0
- urgency : 0

2. Set the value of "exclusive" to TRUE for each execution host using the command qconf -me hostname, where hostname is the name of the host. The values of all host configuration parameters may be displayed using the command qconf -se hostname. The "complex_values" line should look similar to:

   complex_values exclusive=TRUE, but other values may also be included.

3. When submitting a job, the job will be "exclusive" if the value "excl" is included in the resource list specified by the qsub -l option. If the resource list does not include "excl" then the job will not be exclusive, and other jobs may run on the same host or hosts as this job.
4. Example qsub command line for exclusive serial job:

   ```
   qsub -b y -l excl /opt/AnsysEM/AnsysEM15.0n/hfss -ng -
   BatchSolve -machinelist num=1 ~/projects/OptimTee.hfss.
   ```

   Although serial jobs use only one slot, no other jobs will run on the host where this job is running, even if additional slots are present.

5. Example qsub command line for exclusive parallel job using eight engines, each using a single thread of execution:

   ```
   qsub -b y -l excl -pe pe1 8 /opt/Ansoft/HFSS14.0/hfss14.0/hfss
   -ng -BatchSolve -Distributed -machinelist num=8 ~/projects/
   OptimTee.hfss
   ```

   None of the hosts used for this job will be allowed to run other jobs while this job is running.

Consumable Memory Limits

SGE contains several built-in complexes related to memory, including mem_total, for example, but none of these are "consumable". If a job is submitted with resource list including one of these non-consumable memory complexes (such as mem_total), then the job will run on a host or hosts only if sufficient memory is available. If a second job is submitted, the memory request for the second job is compared to the original total when determining if the job may run on a host. This may result in both jobs running out of memory. For example, if host A has mem_total=16G of memory, and two jobs are submitting with option "-l mt=16G", then both jobs could run on host A, if sufficient slots are available on host A.

SGE allows complexes to be "consumable" to avoid this type of problem. If a complex is consumable and a job requests x amount of the complex in the -l resource list, then the available amount of the resource is decreased by x for subsequent jobs. For the same example as above, if the mem_total complex was consumable, then the first job would run on host A. This would decrease the available mem_total from 16G to 16G-16G = 0. The second job could not run on host A because there is no memory available for this job.

The steps below show how to set up a consumable resource called physical_memory to accomplish the same thing. We do not recommend changing the behavior of the built-in complexes (such as mem_total) because other scripts may expect normal behavior of the built-in complexes.

1. Use the command qconf -mcto add a new complex to the table of complexes. Recommended attributes are:
   - name : physical_memory
   - shortcut : phys_mem
   - type : MEMORY
   - relop : <=
   - requestable : YES
   - consumable : YES
   - default : 0
   - urgency : 0
2. Set the value of "physical_memory" to an appropriate value for each execution host using the command qconf -me hostname, where hostnameis the name of the host. The appropriate value is the actual physical memory on each host. Because the type is MEMORY, the K, M, and G suffixes may be used to represent kilobytes, megabytes and gigabytes. The values of all host configuration parameters may be displayed using the command qconf -se hostname. The "complex_values" line should look similar to:

   complex_values physical_memory=16G,

   but other values may also be included, and the memory value should be appropriate for the host.
3. When submitting a job, the physical memory requirement per slot may be specified in the resource list as follows: -l phys_mem=mem_needed. The number of slots assigned to the job on a specific host will be limited by the number of slots available on the host, and also by the physical_memory available on the host.

Serial Jobs in SGE

If a serial job is submitted with the option -l phys_mem=mem_neededincluded, then the job may only run on a host in which the remaining physical_memory is equal to or greater than the mem_neededvalue.

Example 1: Host A has physical_memory=16G, and host B has physical_memory=8G. If mem_neededis 8G, the job may run on either host A or host B. If mem_neededis 16G, then the job may only run on host A.

Example 2: Host A has physical_memory=16G, and host B has physical_memory=8G. Job 1 is already running on host A, and it was submitted with option -l phys_mem=8G. If job 2 is submitted with option -l phys_mem=16G, then job 2 cannot start until job 1 finishes, because only host A has 16GB of physical_memory. If job 2 is submitted with option -l phys_mem=8G, then job 2 may start immediately, and run on either host A or host B, because both hosts have 8G of physical_memory remaining.

Parallel Jobs in SGE

Because the consumable setting for physical_memory is YES (and not JOB), each slot of the job requires a physical_memory of mem_needed. The number of slots on a host assigned to the job is limited by the number of available slots on the host. It is also limited by the physical_memory available on the host; the number of slots assigned to the job cannot exceed the available physical_memory on the host divided by the mem_neededspecification.

Example 1: Execution host A and execution host B both have 4 slots per host (configured in the queue associated with the parallel environment). Host A has physical_memory=16G and host B has physical_memory=8G (shown by commands qconf -se Aand qconf -se B). If a a job is submitted that requires 6 slots and 4G per slot, it will be able to run, with 4 slots on host A and 2 slots on host B. The qsub command might look like: qsub -l phys_mem=4G -pe pe_name 6 command args

Example 2: Same as example 1, except that 7 slots are requested. In this case, the job will never run. Although there are 8 slots available on hosts A and B, only two of the slots on host B are usable by this job because it only has physical_memory of 8G. With only 6 slots total available to this job (4 on host A and 2 on host B), the job can not start. In this case the command might look like: qsub -l phys_mem=4G -pe pe_name 7 command args

Using Multithreading with Parallel Jobs

For large jobs it may be useful to combine multiprocessing with distributed processing. Distributed processing refers to starting multiple processes, in which each process performs a portion of the analysis. These processes may run on the same host or on different hosts. The number of processes running at the same time is known as the number of "analysis engines". Multiprocessing refers to using multiple threads within a single process to decrease the run time of the process. Multiprocessing may also be called multi-threaded processing.

As a concrete example of combining multiprocessing with distributed processing, an analysis could run with four engines, where each engine uses two threads. In order to distribute the processing load so that no processor is overloaded, one slot is generally allocated per thread, so 8 slots would be needed for this example (4 engines * 2 threads per engine = 8 threads). The four engines could all run on a single host, or they could be distributed across 2, 3 or 4 hosts, depending on available slots. Each engine represents a single process, so the two slots for each engine must be allocated on the same host.

This section describes how to set up an SGE cluster so that a specified number of slots per host may be requested when a job is submitted. This procedure will require the cluster administrator privileges. This capability may be used to submit parallel jobs in which one engine runs on each host, and the number of slots per host matches the number of threads used by each engine.

1. Let n be the largest number of slots available on any host used for the jobs. Create a separate parallel environment for each value of the number of slots per host from 1 to n. For example, pe_sph1 is a parallel environment in which one slot is allocated to the job per host, pe_sph2 is a parallel environment in which two slots are allocated to the job per host, etc. The command qconf -ap pe_name may be used to create each new parallel environment. The allocation_rule parameter should be set to the number of slots per host, an integer from 1 to n. The control_slaves parameter should be set to TRUE, as described above. The slots parameter should be set to the maximum number of slots managed by this parallel_environment, which is typi-

cally set to a large number, such as 999. The other parameters should be set to values appropriate for the cluster. For example, the pe_sph2 parallel environment might have the following parameters:

- pe_name : pe_sph2
- slots : 999
- user_lists : NONE
- xuser_lists : NONE
- start_proc_args : /bin/true
- stop_proc_args : /bin/true
- allocation_rule : 2
- control_slaves : TRUE
- job_is_first_task : FALSE
- urgency_slots : min
- accounting_summary : TRUE

2. When submitting a job, use the parallel environment where the slots per host matches the number of threads per engine.

The batchoptions setting -machinelist num=n is required. This should be set to match the number of slots per host. With any analysis, a portion of the analysis may not be distributed across multiple engines.

Example qsub command line for running distributed processing with four engines and multiprocessing with two threads per engine:

```
qsub -V -b y -pe pe_sph2 8 "/opt/AnsysEM/AnsysEM15.0/hfss -ng -
BatchSolve -Distributed -machinelist num=4 -batchoptions
"projects/OptimTee.hfss"
```

The -Voption indicates that the all environment variables in the submission environment should be copied to the job environment.

- The -b y option indicates that hfss is launched directly from the command line, instead of using a script.
- The -pe sph2 8 command_line option indicates that this is a parallel job running under the pe_sph2parallel environment so that two slots are allocated to this job from each host, and that 8 slots in total are allocated to this parallel job.
- The -Distributed option indicates that this is a DSO job, so that multiple engines will be started. The -Distributed option may now have additional options, such as includetypes=xxx, excludetypes=xxx, maxlevels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".
- The -machinelist num=4option indicates that a total of four engines will be started.
- The entire hfss command is in double quotes, and the double quotes enclosing the -batchoptions value are escaped. Each of these double quotes is replaced by the sequence "\"".

Related Topics

Integration with Sun Grid Engine (SGE)

Monitoring ANSYS Electromagnetics Suite SGE Batch Jobs

ANSYS Electromagnetics desktop -monitor Command Line Option for SGE

Issue with qrsh (SGE)

Issue with MainWin Core Services for SGE

Issue with qrsh (SGE)

ANSYS EM parallel batch jobs use the SGE **qrsh** command to launch engine processes on remote hosts. If the **qrsh** command is not working correctly, then the parallel job is unable to launch engine processes on remote hosts. If this problem occurs, the batch log for the job typically includes one or more error messages indicating that a COM engine was unable to be started on a remote host. If this occurs, the user or cluster administrator should verify that the SGE qrsh command is working correctly, and correct the problem if the SGE **qrsh** command is not working correctly.

The **qrsh** command may be tested by running a simple command on a specified host, such as **qrsh -l hostname=host1 hostname** or **qrsh -l hostname=host1 ls /tmp**, where host1 is the remote host name. The first test should simply echo back the hostname of the remote machine. The second test should list the contents of the /tmp directory on the remote machine.

The failures of the SGE qrsh command are associated with the following global sge configuration parameters, listed below with values that may cause the failures:

```
qrsh_command                 /usr/bin/ssh -t
rsh_command                  /usr/bin/ssh -t
rlogin_command               /usr/bin/ssh -t
```

If these parameter settings are removed, then the SGE built-in mechanisms are used for qrsh, rsh, and rlogin. No problems with the built-in versions have been reported. The SGE qconf -sconf global command may be used to view these parameter settings. The SGE qconf -mconf global command may be used to modify or remove these parameter settings.

Issue with MainWin Core Services for SGE

By default, SGE creates a temporary directory for each SGE batch job, and deletes this temporary directory and its contents when the job finishes. SGE sets the TMP and TMPDIR environment variables of the job environment to point to this temporary directory. ANSYS EM desktop software starts the MainWin Core Services on startup, if they are not already running. After the ANSYS EM desktop software finishes, the MainWin Core Services time out and automatically shut down. The MainWin Core Services use the TMP and/or TMPDIR directories to store temporary data. If this temporary data is removed before the services shut down, then the services do not shut down automatically. Normally, SGE will remove the temporary directory and its contents before the services time out. The result is that these extraneous service processes run forever. If this problem occurs, each Ansoft batch job starts an additional set of these services that never shut down. This can result in an excessive number of processes running on the host where the ANSYS EM desktop is started. The names of the service processes are:

- watchdog
- regss
- mwrpcss

Workaround for Issue with MainWin Core Services

One way to avoid this problem is to modify the environment in which the ANSYS EM desktop runs so that the TMP and TMPDIR environment variables do not point to the directory which will be immediately removed by SGE when the job finishes. This can be done by copying the value of the TMPDIR environment variable to the ANS_SGE_TMPDIR environment variable, and unsetting the TMPDIR and TMP environment variables. The services ignore the ANS_SGE_TMPDIR environment variable, but if this variable is set, then it will be used as the temporary directory for the rest of the ANSYS EM software.

Here is an example bash wrapper script that may be used to work around this issue. In this example, the product is hfss, but the same approach will work for any ANSYS EM product. In this example, the script is named sge_hfss and is in the AnsysEM software installation directory. When an ANSYS Electromagnetics desktop job is submitted to the SGE scheduler, the script (sge_hfss, in this example) should be submitted instead of hfss. The script will modify the environment, as needed, then start hfss. When the analysis finishes, the script returns the exit status of hfss.

An alternative is to place the script in an arbitrary directory, and modify the script to include an absolute path to the product (hfss in this example).

Script contents:

```
#! /bin/bash

# This script will not correctly process arguments containing
# spaces or other characters special to the shell.

# Create hfss command line
# In this example, sge_hfss and hfss are in the same directory
# An alternative is to use an absolute path for the hfss command
cmd0=$0
cmd="${cmd0/%sge_hfss/hfss} $@"

# Fix environment variables
export ANS_SGE_TMPDIR=${TMPDIR}
unset TMPDIR
unset TMP

# Run the hfss command and return the exit status
```

```
${cmd}
exit $?
```

Related Topics

Integration with Sun Grid Engine (SGE)

Monitoring ANSYS EM SGE Batch Jobs

ANSYS Electromagnetics desktop -monitor Command Line Option for SGE

Example SGE qsub Command Lines

Issue with MainWin Core Services for SGE

Recommended Practices for SGE Clusters

Command Line Enhancements for ANSYS EM Desktop Products

Any **Tools>Options** setting can be specified via command line, using corresponding registry keys.

This feature is available in all desktop products.

HFSS Examples

```
hfss.exe -batchsolve
-batchoptions  -machinelist num=2
"'HFSS/HPCLicenseType'=pool
projectname.hfss
```

This example demonstrates how to set the same options as the previous example, but here using a registry.txt file:

```
hfss.exe -batchsolve -batchoptions registry.txt
projectname.hfss
```

Registry.txt contains:

```
$begin 'Config'-machinelist num=2
'HFSS/HPCLicenseType'=pool
$end 'Config'
```

Distributed Jobs

An ANSYS EM batch job which distributes the analysis over several hosts may also be called a distributed job. To submit a distributed job, the following ANSYS EM desktop command line options should be used:

- The -Distributed option should be present, and the -Local option should be absent. When running as a batch job under one of the schedulers with direct integration, this option is a directive to the job to 1) obtain the list of hosts allocated to the job, directly from the scheduler, and to 2) use the scheduler to launch the analysis engines on the hosts allocated to the job. The -Distributed option may now have additional options, such as includetypes=xxx, excludetypes=xxx, maxlevels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".
- The -Machinelist num=*num_distributed_engines* option must be included, where *num_dis-*

tributed_engines is the total number of analysis engines to be started on the hosts assigned to the job.

Other examples:

- Serial Job on a Single Processor
- Distributed Job using Four Processors
- Multiprocessing Job Using Four Cores
- Distributed Analysis and Multi-Processing in the Same Job

Serial Job on a Single Processor

Suppose HFSS is installed at "C:\Program Files\AnsysEM\AnsysEM15.0\Win64\ and you are using RSM for DSO:

```
C:\Program Files\AnsysEM\AnsysEM15.0\win64\HFSS.exe -ng -
BatchSolve -machinelist num=2
-monitor \\shared_drive\projs\OptimTee.hfss
```

User is using LSF for remote-analysis/DSO

```
bsub -n 1 C:\Program Files\AnsysEM\AnsysEM15.0\win64\HFSS.exe -
ng -BatchSolve -machinelist num=3 -monitor -local
\\shared_drive\projs\OptimTee.hfss
```

Distributed Job using Four Processors

Ansoft RSM

```
C:\Program Files\AnsysEM\AnsysEM15.0\win64\\HFSS.exe -ng -
Batchsolve -monitor -Distributed
-machinelist list="10.1.1.221, 10.1.1.222, 10.1.1.223,
10.1.1.224" \\shared_drive\projs\OptimTee.hfss
```

LSF

```
bsub -n 4 C:\Program Files\AnsysEM\AnsysEM15.0\win64\HFSS.exe -
ng -Batchsolve -monitor
-Distributed -machinelist num=4
 \\shared_drive\projs\OptimTee.hfss
```

Multiprocessing Job Using Four Cores

Multi-processing job using 4 cores

```
bsub -n 4 -R "span[ptile=4]" C:\Program
Files\AnsysEM\AnsysEM15.0\win64\HFSS.exe -ng -monitor
-Local -BatchSolve -machinelist num=4 -batchoptions
\\shared_drive\registry.txt  \\shared_drive\projs\OptimTee.hfss
```

This requests 4 cores to come from the same machine, as multi-processing needs cores to be on the same machine

Distributed Analysis and Multi-Processing in the Same Job

Distributed-processing using 4 engines and multi-processing using 4 cores, using a total of 16 cores

```
bsub -n 16 -R "span[ptile=4]" c:Program
Files\AnsysEM\AnsysEM15.0\win64\hfss.exe  -ng
-BatchSolve -Distributed -machinelist num=4
-batchoptions \\shared_drive\registry.txt
\\shared_drive\projs\OptimTee.hfss
```

Related Topics

Running HFSS from a Command Line

Integration with Platform's Load Sharing Facility (LSF)

General Terminology for LSF

What a Scheduler Does

Installation of ANSYS Electromagnetics Suite 15.0 on LSF Cluster

Integration of ANSYS Electromagnetics products with LSF

LSF Job Submission Guidelines

Known Issues for LSF

Troubleshooting for LSF

Aborting an Analysis

Integrating ANSYS EM Tools with Third Party Schedulers

This document indicates how to create a dynamically linked library to allow integration of ANSYS EM tools with an arbitrary scheduler environment. Each scheduler proxy library is used for a single specific scheduler environment. If the library is installed with a valid name and in the correct location, then it will automatically be loaded and used by ANSYS EM tools.

- Introduction
- Common Requirements for Running Jobs
- Using a Shared Library (Linux) or a DLL (Microsoft Windows)
- Scheduler Proxy Interfaces
- Using an IronPython Program for Scheduler Integration

Introduction

ANSYS EM Software Tools may be run as serial or parallel jobs on a cluster under control of a scheduler. Serial jobs are run using a single analysis engine at any one time on a single host. If the tool performs multiple analyses (for a frequency sweep or a parametric analysis, for example), the analyses are performed one after the other. Parallel jobs are run using multiple analysis engines running in parallel on the same host or on separate hosts. For parts of the analysis (such as meshing), the parallel job may use only a single analysis engine on a single host. Other parts of the analysis

(such as a frequency sweep, parametric analysis or DDM, for example) may be distributed to multiple analysis engines running in parallel.

- Serial Jobs
- Parallel Jobs

Serial Jobs

When an ANSYS EM batch analysis runs as a serial job, the analysis engines run on the same host as the desktop process. The desktop process does not need to interact with the scheduler to get the names of hosts allocated to the job or to start processes on other hosts.

Parallel Jobs

For a parallel job, the desktop process starts multiple analysis engines that run in parallel. These engines may be started on the host where the desktop process is running, or on other hosts allocated to the job. The desktop process interacts with the scheduler to obtain information on the hosts that are allocated to the job, and to start engines on the local host or on other hosts allocated to the job. This document provides information on how to facilitate this interaction between the desktop process and the scheduler controlling the cluster.

For some popular job schedulers in a standard configuration, ANSYS EM provides an "out of the box" integrated solution that will work with the scheduler. In this case, the ANSYS EM installation includes code that will determine if the analysis is running as a scheduler job and communicate with the scheduler when needed. For other schedulers, the code to obtain information about the hosts allocated to a job and to distribute portions of the job to hosts assigned to the job is not provided in the installation. In order to facilitate using ANSYS EM Software Tools with other schedulers, the user may provide a way for ANSYS EM Tools to interact with the scheduler. Currently, two general approaches are available to users.

In the first approach, the user creates a shared library (on Linux) or a dynamically linked library (on Microsoft Windows) to provide communication between the ANSYS EM Tool and the scheduler. This library is loaded by the ANSYS EM Tool at runtime, and if the ANSYS EM Tool is running as part of a scheduler job, the ANSYS EM Tool interacts with the library to get information from the scheduler, and to start additional processes on specified hosts. Each such library implements the same set of extern "C" functions needed to mediate the interactions between the ANSYS EM Tool and the scheduler. The details of these functions are described below.

In the second approach, the user creates an IronPython program to provide communication between the ANSYS EM Tool and the scheduler. This program is loaded by the ANSYS EM Tool at runtime, and if the ANSYS EM Tool is running as part of a scheduler job, the ANSYS EMs Tool uses the IronPython program to get information from the scheduler, and to start additional processes on specified hosts. Each python script contains a class implementing a specified interface, which contains functions needed to mediate the interactions between the ANSYS EM Tool and the scheduler. The details of the interface are described below. The IronPython interface is equivalent to the extern "C" functions used in the first approach.

Common Requirements for Running Jobs

The following requirements must be met for serial and parallel jobs to run successfully. They apply whether using "out of the box" scheduler integration or scheduler integration using a library or using an IronPython program. When we refer to host requirements, the requirements apply to all hosts that may be allocated to an ANSYS EM serial or parallel batch job.

Installation Requirements

The ANSYS EM installation directory tree must be accessible from all cluster hosts using the same path. One way to achieve this is to place the ANSYS EM installation on a shared drive that is accessible to the cluster hosts using the same pathname. On Windows, this may require the use of UNC names to refer to the installation directory. Another option is to install the ANSYS EM tool locally on each cluster host using the same local directory pathname.

Project File and Directory Requirements

The directory containing the project file must also be available from all cluster hosts using the same path. The project file and the containing directory must be readable and writable by the user account used to run the job. The controlling process for a distributed job is called the Desktop process, and it reads from and writes to the project file and other files in the same directory and its subdirectories. Although only the Desktop process reads from and writes to this directory, the Desktop process may be started on any of the hosts allocated to the job, so all hosts should have access to this directory using the same pathname.

Using a Shared Library (Linux) or a DLL (Microsoft Windows)

This section describes how to create a dynamically linked library to allow integration of ANSYS Electromagnetics Suite 15.0 with an arbitrary scheduler environment. Each scheduler proxy library is used for a single specific scheduler environment. If the library is installed with a valid name and in the correct location, then it will automatically be loaded and used by ANSYS Electromagnetics Suite 15.0.

Installation Details

The scheduler proxy library must be installed in the schedulers subdirectory of the Ansoft installation directory. For example, if the ANSYS EM installation directory is C:\Program Files\AnsysEM\AnsysEM15.0\win64, then the scheduler proxy library must be installed in directory C:\Program Files\AnsysEM\AnsysEM15.0\Win64\schedulers.

The scheduler proxy library base name must match "libprefix_scheduler" on Windows and "liblibprefix_scheduler" on Linux. The extension must be a valid extension for a dynamically loaded library on the platform where it is used. The scheduler proxy library name prefix libprefix shall be unique, so it does not conflict with other scheduler proxy libraries in the same directory. To avoid confusion, the scheduler proxy library name should be all lower case on OSs where file names are case sensitive.

Related Topics

Build Information for Scheduler Proxy Library

Implementation Details for Custom Scheduler Integration

Testing Your Scheduler Integration

Troubleshooting Custom Scheduler Integration

Build Information for Scheduler Proxy Library

This section contains the recommended compiler and linker settings for building a scheduler proxy library.

- 32 Bit Microsoft Windows
- 64 Bit Microsoft Windows
- Linux

32 Bit Microsoft Windows

The proxy library should be compiled and linked as a 32 bit DLL, using the following recommended compiler and linker options:

Compiler Options

- Use of MFC: Use Standard Windows Libraries
- Character Set: Use Multi-Byte Character Set [/D "_MBCS"]
- Runtime Library: Multi-threaded DLL [/MD]
- Calling Convention: __cdecl [/Gd (default)]

Linker Options:

- Create a DLL [/DLL]
- 32 bit code [MACHINE:X86]

64 Bit Microsoft Windows

The proxy library should be compiled and linked as a 64 bit DLL, using the following recommended compiler and linker options:

Compiler Options

- Use of MFC: Use Standard Windows Libraries
- Character Set: Use Multi-Byte Character Set [/D "_MBCS"]
- Runtime Library: Multi-threaded DLL [/MD]
- Calling Convention: __cdecl [/Gd (default)]

Linker Options:

- Create a DLL [/DLL]
- 32 bit code [MACHINE:X64]

Linux

The proxy library should be compiled and linked as shared library (*.so) file. The following compiler and linker options are recommended when building using gcc/g++:

Compiler Options

- Generate 32 bit code: [-m32]

- Generate position independent code, suitable for use in a shared library: [-fpic]
- Generate code compatible with pthreads library: [-pthread]

Linker Options:

- Create a shared object file: [-shared]
- Generate 32 bit code: [-m32]
- Generate position independent code, suitable for use in a shared library: [-fpic]
- Generate code compatible with pthreads library: [-pthread]

Implementation Details for Custom Scheduler Integration

Function Name Prefix

Each exported function will have a scheduler specific function name prefix. The function name prefix will be the same as the library name prefix, except that it is converted to upper case. For example, if the library name prefix is "lsf", then the function name prefix is "LSF". In the examples below, we use FN_PREFIX to denote the function name prefix.

The scheduler proxy library must provide implementations of the following extern "C" functions:

- IsProductLaunchedInYourEnvironment
- GetTempDirectory
- GetMachineListAvailableForDistribution
- GetMessageStringToRegisterForSigTerm
- LaunchProcess
- GetUseRsmForEngineLaunch
- GetThisJobID
- GetSchedulerDisplayName

IsProductLaunchedInYourEnvironment

Purpose

Determine if the program is running in the context of the scheduler for which this library was written.

Signature

```
extern "C" bool FN_PREFIX_IsProductLaunchedInYourEnvironment();
```

Arguments

None.

Return Value

Returns true if the current process is running as a job of the scheduler. Otherwise, false is returned.

Notes

For many schedulers, the presence of certain environment variables or their values may be checked to determine if the current process is running as a job of the scheduler.

GetTempDirectory

Purpose

Get the pathname of the temporary directory provided by the scheduler for the current job. The pathname is an empty string if the scheduler does not provide a temporary directory for the current job.

Signature

```
extern "C" bool FN_PREFIX_GetTempDirectory(char * buffer,
unsigned int* length);
```

Arguments

`buffer`: Pointer to a character buffer to contain the temporary directory path name or NULL.

`length`: Pointer to a location to contain the length of the buffer. Must be a valid pointer to an unsigned int.

Return Value

If argument buffer is NULL, then then required length of the buffer is stored in the location to which argument length points, and true is returned.

If argument buffer is not NULL, then the value to which argument length points (the buffer length) is checked. If it is large enough to contain the pathname of the temporary directory, including the terminal null byte, then the pathname is copied to the buffer and true is returned. If the buffer length is insufficient for the pathname of the temporary directory, then the buffer is unchanged, and false is returned.

Notes

To get the pathname of the temporary directory, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the pathname. After creating a buffer of the appropriate size, the infrastructure calls this function again, passing the pointer to the buffer in the buffer argument and a pointer to the size of the buffer in the length argument.

GetMachineListAvailableForDistribution

Purpose

Get the list of hosts allocated to the current job. A host will appear in the list multiple times if the scheduler has allocated multiple processors or cores on the host to the job. The number of times the host appears in the list is equal to the number of processors or cores of the host that are allocated to the current job. The list is a text string containing a space separated list of host-names.

Signature

```
extern "C" bool
FN_PREFIX_GetMachineListAvailableForDistribution(char * buffer,
unsigned int* length);
```

Arguments

`buffer:` Pointer to a character buffer to contain the list of machines available for distribution or NULL.

`length:` Pointer to a location to contain the length of the buffer. Must be a valid pointer to an unsigned int.

Return Value

If argument buffer is NULL, then then required length of the buffer is stored in the location to which argument length points, and true is returned.

If argument buffer is not NULL, then the value to which argument length points (the buffer length) is checked. If it is large enough to contain the lists of hosts, including the terminal null byte, then the list is copied to the buffer and true is returned. If the buffer length is insufficient for the list of hosts, then the buffer is unchanged, and false is returned.

Notes

To get the list of hosts for distribution, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the list. After creating a buffer of the appropriate size, the infrastructure calls this function again, passing the pointer to the buffer in the buffer argument and a pointer to the size of the buffer in the length argument.

The hostnames in the list provided by this function shall be used in calls to LaunchProcess(). These host names must be in a format that is accepted by that function. See the section below on LaunchProcess.

GetMessageStringToRegisterForSigTerm

Purpose

Obsolete. The string copied to the buffer should be an empty string.

Signature

```
extern "C" bool FN_PREFIX_GetMessageStringToRegisterForSigTerm
(char * buffer, unsigned int* length);
```

Arguments

buffer: Pointer to a character buffer to contain the string or NULL.

length: Pointer to a location to contain the length of the buffer. Must be a valid pointer to an unsigned int.

Return Value

If argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned.

If argument buffer is not NULL, then the value to which argument length points (the buffer length) is checked. If it is large enough to contain the string, including the terminal null byte, then the string is copied to the buffer and true is returned. If the buffer length is insufficient for the string, then the buffer is unchanged, and false is returned.

Notes

To get the string, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the string. After creating a buffer of the appropriate size, the infra-

structure calls this function again, passing the pointer to the buffer in the buffer argument and a pointer to the size of the buffer in the length argument.

LaunchProcess

Purpose

Launch a local or remote process to run an analysis engine. This function is called by the ANSYS Electromagnetics desktop application to launch an engine process on a specified host. The hostname is one of the names in the list provided by the GetMachineListAvailableForDistribution function. See the GetMachineListAvailableForDistribution section above. If the hostname does not refer to the local host, then this function shall use the scheduler to launch the engine on the specified host. If the hostname refers to the local host, then the engine may be started as a child process, or it may be started using the scheduler.

Signature

```
extern "C" int FN_PREFIX_LaunchProcess(const char* hostName,
const char* exePathName, const char* arg1, const char* arg2);
```

Arguments

`hostName`: The name of the host where the process is to be launched.

`exePathName`: The pathname of the analysis engine executable to be started.

`arg1`: The first argument of the analysis engine command line.

`arg2`: The second argument of the analysis engine command line.

Return Value

Returns 0 on success. Returns a non-zero value if an error occurs.

Notes

The hostName argument will be one of the hostnames provided by the function `GetMachineListAvailableForDistribution()`.

If the hostName argument is the same as the current host, then the analysis engine process may be started as a child process. If the hostName argument is not the same as the current host, then the analysis engine process will be started on the remote host using the facilities available in the scheduler environment. The command line of the analysis engine process is `exePathName arg1 arg2`. The command line arguments `arg1` and `arg2` may contain newlines, tabs, spaces or other characters that are interpreted by the command processor, such as single quote (') or double quote (") characters, or dollar signs ($). Newlines or tabs may be replaced by spaces, if the newline or tab characters cannot be easily handled. If the analysis engine command is processed by a shell, then it may be necessary to quote any special characters in the `exePathName` or in the arguments so that the special meaning is removed. If a scheduler command is used to request the scheduler to launch the command to start the engine process, the analysis engine command may be processed by the shell twice: once when the scheduler command is processed, and a second time when the analysis engine process is started. If this is the case, then the quoting of special characters needs to account for two passes through the command processor.

GetUseRsmForEngineLaunch

Purpose

This function is optional. If this feature is not needed, then the function need not be implemented. Most schedulers should not need this feature.

For some schedulers, it may be desirable for the Ansoft RSM service to launch the engine processes instead of using the scheduler proxy library. For example, if the scheduler proxy library is limited to launching one process per host, then the scheduler proxy library may be used to launch one Ansoft RSM service executable per host, and the Ansoft RSM executable will launch all of the engine processes.

If the Ansoft RSM service should be used to launch engine processes for this scheduler, then this function shall be implemented and it shall return true.

If the Ansoft RSM service should not be used to launch engine processes for this scheduler, then this function is not required. If it is implemented, it should return false. If it is not implemented, it will be treated the same as if it was implemented and returns false.

Signature

```
extern "C" bool FN_PREFIX_GetUseRsmForEngineLaunch(void)
```

Arguments

None.

Return Value

Returns true if the Ansoft RSM service should be used to launch engine processes for this scheduler. Returns false if the Ansoft RSM service should not be used to launch engine processes for this scheduler.

Notes

This function is optional. If not implemented, then it is treated the same as if it was implemented and returns false.

GetThisJobID

Purpose

Get a string identifying the job currently running in the scheduler environment. This string is displayed to the end user to identify the job.

Signature

```
extern "C" bool FN_PREFIX_GetThisJobID(char * buffer, unsigned
int* length);
```

Arguments

`buffer`: Pointer to a character buffer to contain the Job ID or NULL.

`length`: Pointer to a location to contain the length of the buffer. Must be a valid pointer to an unsigned int.

Return Value

If argument buffer is NULL, then then required length of the buffer is stored in the location to

which argument length points, and true is returned.

If argument buffer is not NULL, then the value to which argument length points (the buffer length) is checked. If it is large enough to contain the string identifying the current job, including the terminal null byte, then the job ID is copied to the buffer and true is returned. If the buffer length is insufficient for the job ID, then the buffer is unchanged, and false is returned.

Notes

To get the job ID, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the job ID. After creating a buffer of the appropriate size, the infrastructure calls this function again, passing the pointer to the buffer in the buffer argument and a pointer to the size of the buffer in the length argument.

For many schedulers, the job ID may be obtained from the value of an environment variable.

GetSchedulerDisplayName

Purpose

Get a string identifying the scheduler associated with the current scheduler proxy library. This string is displayed to the end user to identify the scheduler.

Signature

```
extern "C" bool FN_PREFIX_GetSchedulerDisplayName(char *
buffer, unsigned int* length);
```

Arguments

buffer: Pointer to a character buffer to contain the scheduler display name or NULL.

length: Pointer to a location to contain the length of the buffer. Must be a valid pointer to an unsigned int.

Return Value

If argument buffer is NULL, then required length of the buffer is stored in the location to which argument length points, and true is returned.

If argument buffer is not NULL, then the value to which argument length points (the buffer length) is checked. If it is large enough to contain the scheduler display name, including the terminal null byte, then the scheduler display name is copied to the buffer and true is returned. If the buffer length is insufficient for the scheduler display name, then the buffer is unchanged, and false is returned.

Notes

To get the scheduler display name, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the scheduler display name. After creating a buffer of the appropriate size, the infrastructure calls this function again, passing the pointer to the buffer in the buffer argument and a pointer to the size of the buffer in the length argument.

The scheduler display name is generally a fixed string.

Scheduler Proxy Interfaces

Scheduler proxy supports following new graphical interface functions. The scheduler specific prefix of each function is not shown in this listing.

void Initialize(const std::string& config):

Initialize the proxy library for scheduler interaction. The **config** argument contains scheduler specific initialization information.

int CheckEnvironment(std::string& msg):

Check the environment in which the proxy library is running.

- Returns 0 (success) if the environment is appropriate for submitting jobs to the scheduler.
- Returns a non-zero error code if the environment is incorrect. If a non-zero error code is returned, an error message to display to the user is written to the msg argument.

int GetSchedulerInfo(std::string& msg, std::string& schedulerName, std::string& schedulerDescription, std::string& schedulerVersion):

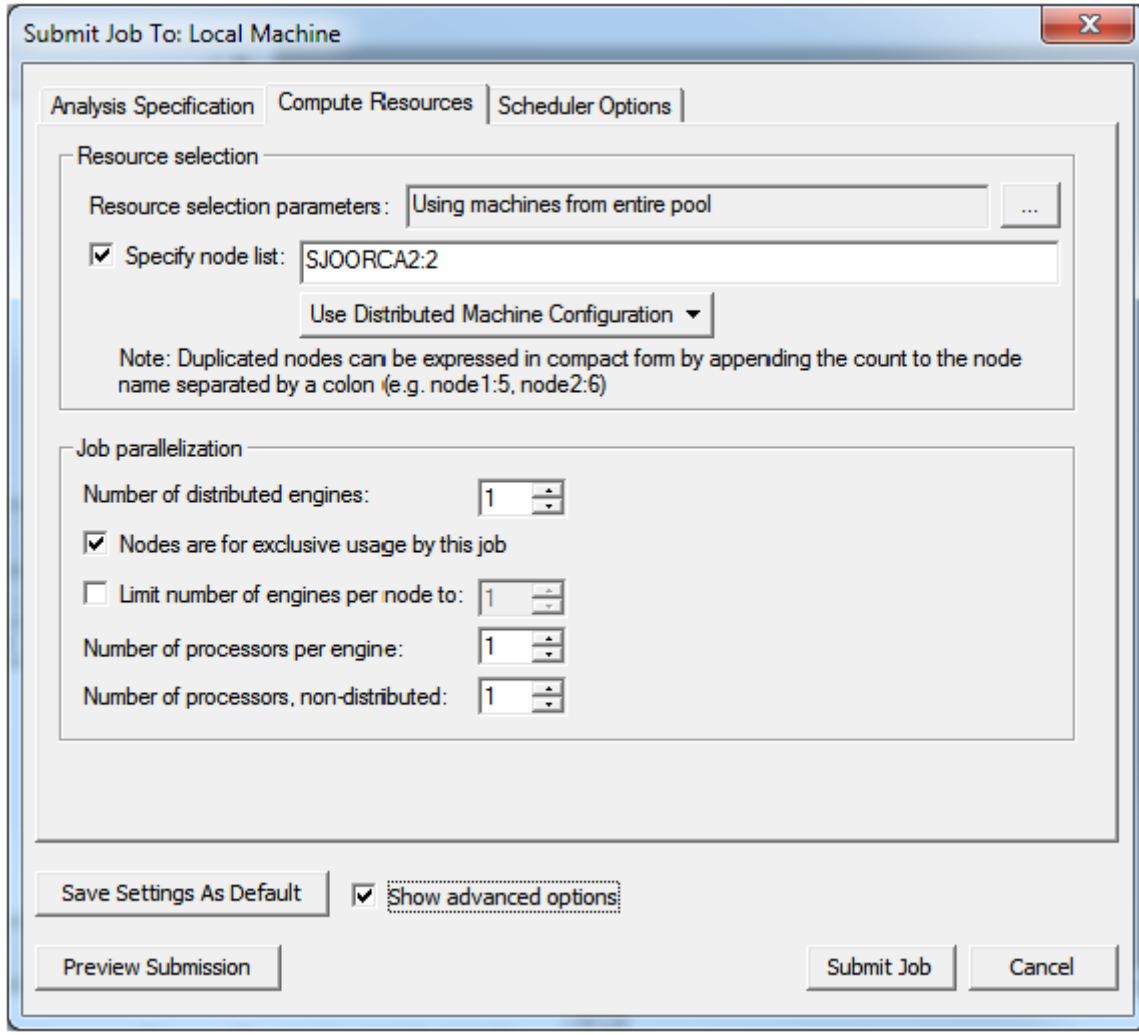
This function returns some basic information about the scheduler with which the scheduler proxy library interacts.

- On success, 0 is returned, and the scheduler name, scheduler description, and scheduler version are written to the **schedulerName**, **schedulerDescription** and **schedulerVersion** arguments.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

int GetComputeResourceAttributes(std::string& msg, AttributeDefinitionsStruct& attributeDefs):

The **Compute Resource Selection Parameters** dialog allows the user to specify scheduler specific resources. This function returns the information used to create and populate the Com-

pute Resource Selection Parameters dialog.



Each line in the dialog is defined by a single attribute definition in the **attributeDefs** argument. An attribute definition defines the name and description of an attribute, as well as information about the allowed values and the default value. In general, only the most commonly specified job attributes are included in the **attributeDefs** argument.

- On success, 0 is returned, and the attribute definitions are written to the **attributeDefs** argument.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.
- If the scheduler proxy library does not support any attributes using this approach, the

attributeDefs argument will contain no attribute definitions, and 0 will be returned.

int AbortJob(std::string& msg, const std::string& jobID, bool force, const SubmissionUserStruct& submissionUser):

This function requests the scheduler to abort a job identified by the **jobID** argument. If the force argument is true, then errors should be ignored (the exact behavior is scheduler specific). The **submissionUser** argument contains information about the client user (the user running the Desktop process). The request to abort the job should run in the context of this user. If no user is specified, then the request to abort the job runs as the user of the process or thread running the function.

- If the request is successfully submitted, then 0 is returned.
- If there is an error, then a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

int SubmitUniformJob(std::string& msg, std::string& jobID, const CmdLineStruct& cmdLineInfo, const JobParallelizationStruct& jobParallelization, const UniformComputeResourcesStruct& computeResources, const JobOptionsStruct& jobOptions, const JobAtributesStruct& jobAttributes, const SubmissionUserStruct& submissionUser,const IJobParameters* jobParametersCB):

This function submits a job to the scheduler.

- On success, 0 is returned, and the job identifier of the newly submitted job is written to the **jobID** argument.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

This function is used to submit jobs to the scheduler in which the resources allocated to the job are uniformly distributed across the nodes allocated to the job. All other arguments are input arguments, and they are described below:

The **cmdLineInfo** argument contains the command line arguments. The first argument is the command name.

The **jobParallelization** argument contains information on how the job should be parallelized. It contains the following integral parameters:

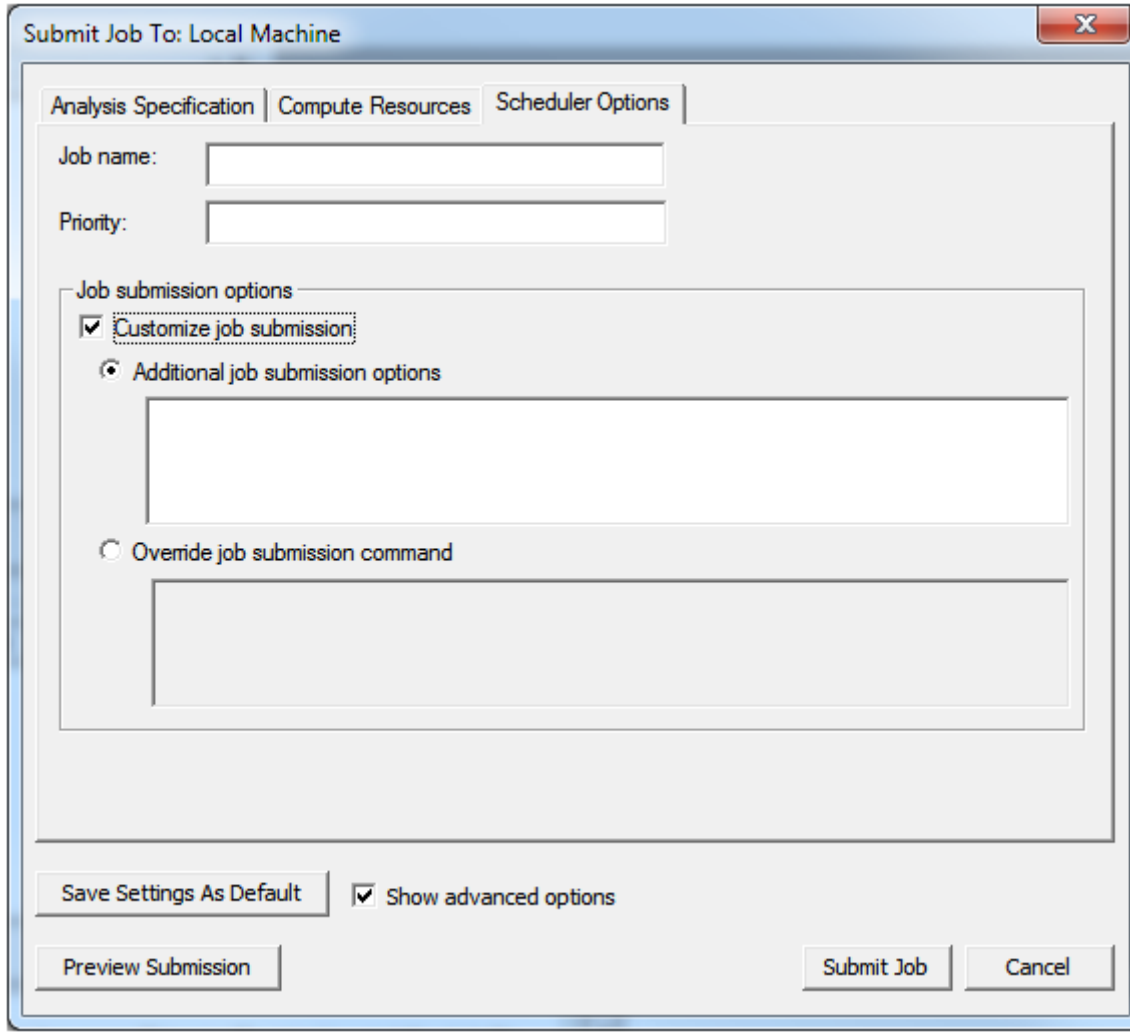
- the total number of distributed engines,
- the number of cores to allocate for each distributed engine,
- the maximum number of engines to allocate to a single node (optional), and
- the number of cores to allocated for the non-distributed portion of the analysis.
- It also contains a boolean parameter indicating whether nodes used for this job should be exclusively allocated to this job.

The **computeResources** argument is a reference to an object of type **UniformComputeResourcesStruct**. This **struct** contains zero or more resource attribute settings for the job. Each resource attribute setting consists or a resource name and a resource value. The resource name is the name of one of the resources defined in the **AttributeDefinitionsStruct** filled in by the **GetComputeResourceAttributes()** function. The resource attribute value is the value specified for the resource attribute

by the user using the **Compute Resource Selection Parameters** dialog. If no resource attributes are specified by the user in this dialog, then the **computeResources** argument will contain no resource attribute settings.

The **jobOptions** argument contains the environment variable settings for the job.

The **jobAttributes** argument contains job submission attributes which are not necessarily related to the compute resources allocated to the job. The job name and the requested job priority are included in this data structure. The **SchedulerOptions** tab of the **Job Submission** dialog allows the user to either specify additional job submission options or to specify all submission options, replacing the settings from the other **Job Submission** dialog controls.



The user specified submission options are included in this data structure, as well as a boolean

setting indicating whether the user specified options are in addition to the automatically generated options, or whether they replace the automatically generated submission options.

The **submissionUser** argument contains information about the client user (the user running the Desktop process). The job is submitted to the scheduler to run as this user.

The **jobParametersCB** argument is a pointer to an object that implements the **IJobParameters** interface. This interface allows the scheduler proxy library to get additional information about the job. Specifically, the **GetWorkingDirectory()** interface function returns the working directory to be used for the job.

The **cmdLineInfo** argument contains the command line arguments. The first argument is the command name.

int SubmitNonUniformJob(std::string& msg, std::string& jobID, const CmdLineStruct& cmdLineInfo, const JobParallelizationStruct& jobParallelization, const NonUniformComputeResourcesStruct& computeResources, const JobOptionsStruct& jobOptions, const JobAtributesStruct& jobAttributes, const SubmissionUserStruct& submissionUser,const IJobParameters* jobParametersCB):

This function submits a job to the scheduler.

- On success, 0 is returned, and the job identifier of the newly submitted job is written to the **jobID** argument.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

This function is used to submit jobs to the scheduler in which the nodes to use and the number of engines to run on each node are specified by the user. All other arguments are input arguments, as for the **SubmitUniformJob()** function. These input arguments are the same as for the **SubmitUniformJob()** function, except that the **computeResources** argument is a reference to a **NonUniformComputeResourcesStruct**, as described below:

The **computeResources** argument is a reference to an object of type **NonUniformComputeResourcesStruct**. This object contains a vector of pairs, where each pair consists of the name of a node in the cluster, and the number of engines to run on the node.

int PreviewUniformJob(std::string& msg, std::string& preview, const CmdLineStruct& cmdLineInfo, const JobParallelizationStruct& jobParallelization, const UniformComputeResourcesStruct& computeResources, const JobOptionsStruct& jobOptions, const JobAtributesStruct& jobAttributes, const SubmissionUserStruct& submissionUser, const IJobParameters* jobParametersCB):

This function is similar to the **SubmitUniformJob()** function, but instead of submitting the job, text representing how the job will be submitted is written to the preview argument. Typically the preview text includes the job submission command and the contents of the job script created for the job. For some schedulers, this content may not be meaningful, so the text returned could be different.

- On success, 0 is returned, and the job preview text is written to the preview argument.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

The other arguments are input arguments with the same meaning as for the **SubmitUniformJob()** function. The **submissionUser** argument is ignored for this function.

Testing Your Scheduler Integration

One way to test these functions is to run the analysis for an ANSYS EM product in batch mode. When running in batch mode, a batch log file is created in the same directory as the project file. The batch log file has the same base name is the project file, with an extension of ".log". For example, if the project file is TestProject123.hfss, then the batch file is TestProject123.log. The batch log file contains useful information about the analysis run.

See the product specific help for details on running the product in batch mode, and for the command line options to use for distributed analysis.

- Testing IsProductLaunchedInYourEnvironment
- Testing GetSchedulerDisplayName and GetThisJobID
- Testing GetTempDirectory
- Testing GetMachineListAvailableForDistribution
- Testing LaunchProcess
- Testing GetUseRsmForEngineLaunch

Testing IsProductLaunchedInYourEnvironment

This function should be tested first. If the ANSYS EM application is not able to load and run this function, or if it returns false, then none of the other functions will be called. If the batch analysis is running in a scheduler environment, and this function returns true, then there will be an "info" message near the beginning of the batch log indicating that the analysis is running as a scheduler job. This message will include the scheduler display name returned by the function GetSchedulerDisplayName, and it will also include the job ID returned by the function GetThisJobID. If the batch analysis is not running in a scheduler environment, then none of the messages will include a scheduler display name or job ID.

If this message does not appear when running in a scheduler environment, ensure that the scheduler proxy library is named correctly, that it is built correctly, that it is installed in the correct directory, and that the function name prefix is the same is the library prefix converted to upper case.

Testing GetSchedulerDisplayName and GetThisJobID

As described above, when running a batch job in a scheduler environment, the scheduler display name and the job ID will appear in an "info" message near the beginning of the batch log. The values returned by these functions are copied to this message verbatim, so they can be directly compared to the expected values.

Testing GetTempDirectory

Many schedulers create a temporary directory for each job and delete the directory after the job finishes. One way to verify that this function is working correctly is to determine the pathname that the scheduler uses for the temporary directory and to monitor the contents of the temp directory as the job is running. If the analysis engines write files to this directory as the job

runs, then this function is working.

Testing GetMachineListAvailableForDistribution

This function is used for distributed analysis. The analysis may be distributed across several machines if portions of the analysis are independent. For example, frequency sweeps, parametric analysis and domain decomposition allow different portions of the analysis to be distributed across machines. The analysis in a batch job will be distributed to multiple processors or hosts if the analysis includes a setup that may be distributed (e.g., a frequency sweep or parametric analysis) and the **-Distributed** option is included in the desktop command line. The list of machines is displayed in an "info" message near the beginning of the batch log. The list in the info message can be directly compared to the expected list of machines.

To verify that the machine list is constructed correctly for a variety of cases, it may be necessary to test several jobs with different resource requirements and verify that the machine list is correct in each case. For example, one may run batch analyses with the following resource requirements:

- One processor on one host
- Several processors on one host
- One processor on each of several hosts
- Several processors on each of several hosts

Testing LaunchProcess

This function is used to launch analysis engines in the case where the analysis is distributed across multiple hosts. The analysis may be distributed across several machines if portions of the analysis are independent. For example, frequency sweeps, parametric analysis and domain decomposition allow different portions of the analysis to be distributed across machines. The analysis in a batch job will be distributed to multiple processors or hosts if the analysis includes a setup that may be distributed (e.g., a frequency sweep or parametric analysis) and the **-Distributed** option is included in the desktop command line. The list of machines is displayed in an "info" message near the beginning of the batch log. The batch log may also contain info messages when portions of the analysis distributed to different machines start or finish. These messages usually include the name of the host when the analysis ran or will run. One can verify that the analysis is actually running on the expected host or hosts using the Linux ps command or the Windows Task Manager.

In general, one analysis engine is started for each occurrence of each host in the list of machines available for distribution. For example, if the list of hosts is "hostA hostA hostA hostB hostB", then a total of 5 engines would be started, three on hostA and two on hostB. In some cases, an additional engine is started to perform the portion of the analysis which is not distributed; if this is the case, the non-distributed engine is idle during the portion of the analysis which is distributed. If this occurs in the case where the list of hosts is "hostA hostA hostA hostB hostB", then a total of 6 engines would be started, but at most 5 engines would be active at any given time. When each analysis engine is running, it may start additional child processes to do a portion of the analysis, but these are not counted as additional analysis engines because the parent of the sub-engine is inactive (waiting for the sub-engine results) when the sub-engine is active.

Testing should be sufficient to demonstrate that the scheduler proxy library can start multiple engine processes on the desktop host, and can also start multiple engine processes on other hosts.

Testing GetUseRsmForEngineLaunch

In most cases, this function will not be implemented or tested. If this function is implemented and returns true, then the ANSYS Electromagnetics desktop application will not start the analysis engines using the LaunchProcess function directly. Instead, the ANSYS Electromagnetics desktop application will start one AnsoftRSMService process on each host using the `LaunchProcess` function, and the engine processes will be started by these AnsoftRSMService processes. One may check for these processes using the Linux ps command or the Windows Task Manager. One AnsoftRSMService process should run on each host. These processes will be named ansoftrsmservice.exe or AnsoftRSMService.exe. These processes will be started on each host before any analysis engine is started on the host, and will remain running until the job is complete.

Troubleshooting Custom Scheduler Integration

None of the Proxy Functions are Called

There are several problems which could result in none of the proxy functions being called.

The scheduler proxy library must be installed in the schedulers subdirectory of the Ansoft installation directory. The installation directory is set in the registry entry `HKEY_CURRENT_USER/Software/Ansoft/Product/Version/Desktop/InstallationDirectory`, where Product is the ANSYS EM product name (e.g, HFSS) and Version is the ANSYS EM product version (e.g., 15.0).

The scheduler proxy library name must match "*_scheduler.dll" on Windows and "lib*_scheduler.so" on Linux. If the library name does not match this format, then the library will not be loaded. In addition, the

function name prefix must be the same as the library name prefix converted to upper case. For example, if the library name prefix is "abc", then the function name prefix is "ABC". In this example, the library name is "abc_scheduler.dll" on Windows, and "libabc_scheduler.so" on Linux. In this example, the full name of the IsProductLaunchedInYourEnvironment function is ABC_IsProductLaunchedInYourEnvironment on Windows and Linux, and it must have extern "C" linkage.

Verify that the compile and link flags follow the guidelines in the section "Build Information", above. Incorrect compile or link flags may prevent the library from being loaded by the ANSYS Electromagnetics product.

If there is a problem with calling the `IsProductLaunchedInYourEnvironment` function, then none of the other functions will be called. The other functions are only called if the `IsProductLaunchedInYourEnvironment` function is successfully called and returns true.

Troubleshooting IsProductLaunchedInYourEnvironment Function

Verify that the conditions specified in the section "None of the Proxy Functions are Called" are met.

Verify that this function returns true when called in an environment running under the scheduler, and that it returns false when called in an environment not running under the scheduler.

Troubleshooting GetSchedulerDisplayName

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

Verify that the scheduler display name is a valid ASCII string.

Verify that, if argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned. The required buffer length must include space for the string null terminator.

Verify that, if argument buffer is not NULL and the value to which argument length points (the buffer length) is large enough to contain the display name, including the terminal null byte, then the display name is copied to the buffer and true is returned.

Troubleshooting GetThisJobID

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

Verify that the job ID is a valid ASCII string.

Verify that, if argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned. The required buffer length must include space for the string null terminator.

Verify that, if argument buffer is not NULL and the value to which argument length points (the buffer length) is large enough to contain the job ID, including the terminal null byte, then the job ID is copied to the buffer and true is returned.

Troubleshooting GetTempDirectory

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

Verify that the temporary directory name is a valid ASCII string.

Verify that, if argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned. The required buffer length must include space for the string null terminator.

Verify that, if argument buffer is not NULL and the value to which argument length points (the buffer length) is large enough to contain the temporary directory pathname, including the terminal null byte, then the temporary directory pathname is copied to the buffer and true is returned.

Troubleshooting GetMachineListAvailableForDistribution

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

Verify that the list of hosts is a valid ASCII string containing a space separated list of host names. A host name will appear in the list a number of times equal to the number of processors or cores available to the job on that host.

Verify that, if argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned. The required buffer length must include space for the string null terminator.

Verify that, if argument buffer is not NULL and the value to which argument length points (the buffer length) is large enough to contain the list of hosts, including the terminal null byte, then the list of hosts is copied to the buffer and true is returned.

Troubleshooting LaunchProcess

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

The hostName argument is a host name from the list returned by the GetMachineListAvailableForDistribution function. Verify that the LaunchProcess function can accept host names in the format returned by the GetMachineListAvailableForDistribution function.

The exePathName argument is the pathname of the analysis engine executable to be started. This pathname may contain spaces or other characters special to the shell. Ensure that the LaunchProcess function is able to handle such cases.

The arg1 and arg2 arguments may contain newlines, tabs, single quotes, spaces, dollar signs, and other characters which may be special to the shell. Ensure that the LaunchProcess function is able to handle such cases. If needed, the newline characters may be replaced by other whitespace characters. One or both of these arguments could also be an empty string; verify that the empty string is correctly passed to the engine process command line.

If a scheduler command is used to launch the engine process on a remote machine, the engine command line may be processed by the shell twice, once when the scheduler command is processed by the shell, and again when the engine command is processed by the shell. In such cases, the quoting of characters special to the shell will need to be take these two passes through the shell into account. In some implementations, it may be necessary or convenient to use different approaches for launching engine processes on the local machine and on remote machines; if this is done, verify that the approach used to determine whether the hostName argument represents the local machine is correct.

Troubleshooting GetUseRsmForEngineLaunch

In most cases, this function will not be implemented. If it is implemented, then follow the suggestions below.

`Verify that the IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

If the RSM should be used for launching engines, verify that this function returns true. Otherwise, verify that this function returns false.

Using an IronPython Program for Integration with a Scheduler

This section describes how to create an IronPython program for integration with a scheduler. Each such program is used for a single specific scheduler environment. If the program is installed with a valid name and in the correct location, then it will automatically be loaded and used by ANSYS EM tools.

Installation Details

The IronPython program must be installed in the schedulers subdirectory of the ANSYS EM installation directory. For example, if the installation directory is C:\Program Files\AnsysEM\AnsysEM15.0\Win64, then the IronPython program must be installed in directory C:\Program Files\AnsysEM\AnsysEM15.0\Win64\schedulers.

The program file extension must be ".py". Select the program name so that it does not conflict with other IronPython programs in the same directory. If the Operating System or file system treat file names in a case sensitive manner, the file extension ".py" must be lower case.

Python Programming Notes

The scheduler program will be run in the IronPython environment both on Microsoft Windows and on Linux. There are some differences between IronPython and CPython. The version of IronPython in use as of June 9, 2011 is 2.6.1.

Implementation Details

The program must contain the following:

Import the ISchedulerPluginExtension interface as follows:

```
from Ansys.Ansoft.SchedulerPluginDotNet import
ISchedulerPluginExtension
```

Define a class which implements the ISchedulerPluginExtension interface. In this document, this class is named SamplePluginExtension, but any class name may be used. The class member functions are described in the next section. The class definition will look similar to the following:

```
class SamplePluginExtension(ISchedulerPluginExtension):

  def GetName(self):
    return "SamplePluginExtension"

  def GetDescription(self):
    return "Example python script plugin extension"
. . .
```

Include the following line in the program so that the class that you have defined, SamplePluginExtension, is loaded by the infrastructure:

```
ExtensionRegistrar.RegisterPluginExtension(SamplePluginExtensio
n())
```

The infrastructure will make the EntensionRegistrar object available in the environment where the program is loaded.

Each of the functions to be implemented in the SamplePluginExtension class is described below.

- GetName
- GetDescription
- IsProductLaunchedInYourEnvironment
- GetSchedulerDisplayName
- GetThisJobID
- GetUseRsmForEngineLaunch
- GetTempDirectory
- GetMessageStringToRegisterForSigTerm
- GetMachineListAvailableForDistribution
- LaunchProcess

GetName [IronPython]

Purpose

Return a short string containing the name of the plugin extension. This string is used to identify the scheduler plugin extension in logs or program output.

Signature

```
GetName(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the name of the plugin extension.

Notes

The plugin extension name is generally a fixed string.

GetDescription [IronPython]

Purpose

Return a string containing the description of the plugin extension. This string is used to identify the scheduler plugin extension in logs or program output.

Signature

```
GetDescription(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the description of the plugin extension.

Notes

The plugin extension description is generally a fixed string.

IsProductLaunchedInYourEnvironment [IronPython]

Purpose

Determine if the program is running in the context of the scheduler for which this program was written.

Signature

```
IsProductLaunchedInYourEnvironment(self)
```

Arguments (excluding self)

None.

Return Value

Returns True if the current process is running as a job of the scheduler. Otherwise, False is returned.

Notes

For many schedulers, the presence of certain environment variables or their values may be checked to determine if the current process is running as a job of the scheduler.

GetSchedulerDisplayName [IronPython]

Purpose

Get a string identifying the scheduler associated with the current plugin extension. This string is used to identify the scheduler.

Signature

```
GetSchedulerDisplayName(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the description of the scheduler for which this plugin extension was written.

Notes

The scheduler display name is generally a fixed string.

GetThisJobID [IronPython]

Purpose

Get a string identifying the job currently running in the scheduler environment. This string is displayed to the end user to identify the job.

Signature

```
GetThisJobID(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the Job ID for the current job.

Notes

For many schedulers, the job ID may be obtained from the value of an environment variable.

GetUseRsmForEngineLaunch [IronPython]

Purpose

For some schedulers, it may be desirable for the AnsoftRSM program to launch the engine processes instead of using the scheduler plugin extension directly. For example, if the plugin extension is limited to launching one process per host, then the plugin extension may be used to launch one AnsoftRSM executable per host, and the AnsoftRSM executable will launch all of the engine processes.

If AnsoftRSM should be used to launch engine processes for this scheduler, then this function shall return True.

If AnsoftRSM should not be used to launch engine processes for this scheduler, then this function shall return False.

Signature

```
GetUseRsmForEngineLaunch(self)
```

Arguments (excluding self)

None.

Return Value

Returns True if AnsoftRSM should be used to launch engine processes for this scheduler. Returns False if the plugin extension should be used to directly launch engine processes for this scheduler.

Notes

If this function returns True, then the plugin extension will directly launch only one process on each host.

GetTempDirectory [IronPython]

Purpose

Get the pathname of the temporary directory provided by the scheduler for the current job. The pathname is an empty string if the scheduler does not provide a temporary directory for the current job.

Signature

```
GetTempDirectory(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the pathname of the temporary directory provided by the scheduler for the current job. Returns an empty string if no temporary directory is provided by the scheduler for the current job.

Notes

If the return value is an empty string, then the temporary directory specified on the command line or in the registry will be used.

GetMessageStringToRegisterForSigTerm [IronPython]

Purpose

Obsolete. This function should return an empty string.

Signature

```
GetMessageStringToRegisterForSigTerm(self)
```

Arguments (excluding self)

None.

Return Value

Returns an empty string.

Notes

This function should always return an empty string.

GetMachineListAvailableForDistribution [IronPython]

Purpose

Get the names of the hosts allocated to the current job. A host name will appear in the output string multiple times if the scheduler has allocated multiple processors or cores on the host to the job. The number of times the host name appears in the string is equal to the number of processors or cores of the host that are allocated to the current job. The output is a text string containing the host names separated by space characters.

Signature

```
GetMachineListAvailableForDistribution(self)
```

Arguments (excluding self)

None.

Return Value

A string containing the names of the hosts allocated to the job, separated by space characterss. The number of times the host appears in the string is equal to the number of processors or cores of the host that are allocated to the current job.

Notes

The hostnames in the string provided by this function shall be used in calls to LaunchProcess(). The host names must be in a format that is accepted by that function. See the section below on LaunchProcess.

LaunchProcess [IronPython]

Purpose

Launch a local or remote process to run an analysis engine. This function is called by the ANSYS Electromagnetics desktop application to launch an engine process on a specified host. The hostname is one of the names provided by the GetMachineListAvailableForDistribution function. See the GetMachineListAvailableForDistribution section above. If the hostname does not refer to the local host, then this function shall use the scheduler to launch the engine on the specified host. If the hostname refers to the local host, then the engine may be started as a child process, or it may be started using the scheduler.

Signature

```
LaunchProcess(self, hostName, exePathName, arg1, arg2)
```

Arguments (excluding self)

hostName: The name of the host where the process is to be launched.

exePathName: The pathname of the analysis engine executable to be started.

arg1: The first argument of the analysis engine command line.

arg2: The second argument of the analysis engine command line.

Return Value

Returns 0 on success. Returns a non-zero value if an error occurs.

Notes

The hostName argument will be one of the hostnames provided by the function GetMachineListAvailableForDistribution().

If the hostName argument is the same as the current host, then the analysis engine process may be started as a child process. If the hostName argument is not the same as the current host, then the analysis engine process will be started on the remote host using the facilities available in the scheduler environment. The command line of the analysis engine process is exePathName arg1 arg2. The command line arguments arg1 and arg2 may contain spaces or other characters that are interpreted by the command processor, such as backslash (\), single quote (') or double quote (") characters, or dollar signs ($). If the analysis engine command is processed by a shell, then it may be necessary to quote any special characters in the exePathName or in the arguments so that the special meaning is removed. If a scheduler command is used to request the scheduler to launch the command to start the engine process, and that command is processed by a command shell, then the analysis engine command may be processed by the shell twice: once when the scheduler command is processed, and a second time when the analysis engine process is started. If this is the case, then the quoting of special characters needs to account for two passes through the command processor.

The command line arguments arg1 and arg2 may be empty strings. These arguments must be preserved, even if they are empty strings. In some versions of the IronPython subprocess module, empty argument strings are discarded, resulting in an incorrect number of command line arguments. A workaround for this issue is to replace an empty string argument by a string consisting of a single space character.

RSM Integration with Job Management UI

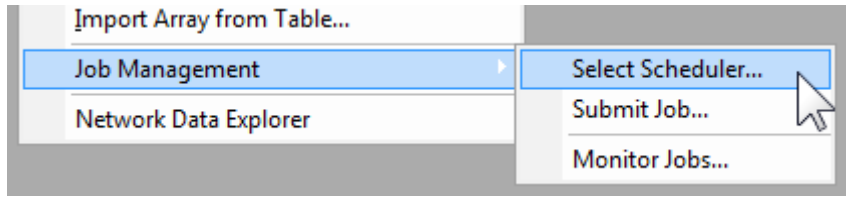
ANSYS Electromagnetics supports its own Remote Simulation Management (RSM) software, along with other High Performance Computing (HPC) software management programs (see High Performance Computing (HPC) Integration).

When do you need RSM?

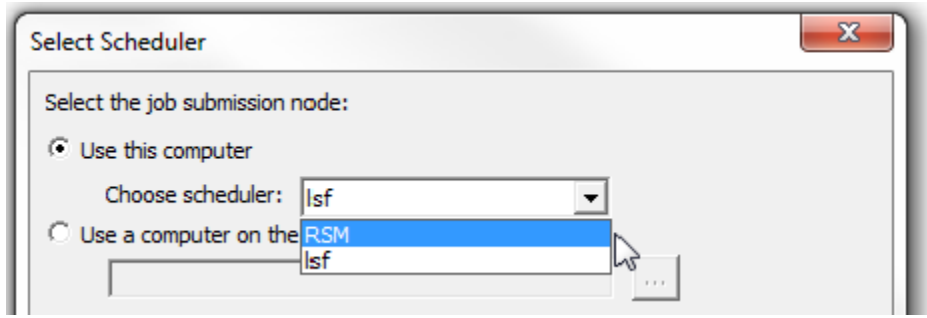
RSM is in general required if you want to run remote or distributed simulations. However, if you have a separate scheduling system that ANSYS Electromagnetics supports, and you plan to run batchsolve simulations only, then you may not need to install RSM. For details of installation and configuration of RSM, see the **ANSYS Electromagnetics Installation Guides**.

Job Management UI for RSM

You can use the Job Management UI to submit batch jobs to RSM. The Job Management UI is accessed by running ANSYS Electromagnetics product Desktop on the designated 'Postprocessing node' of the cluster. The Desktop provides UI commands for Scheduler selection, Job submission and Job monitoring/control. You access the Scheduler User Interface by clicking **Tools>Job Management>Select Scheduler...**

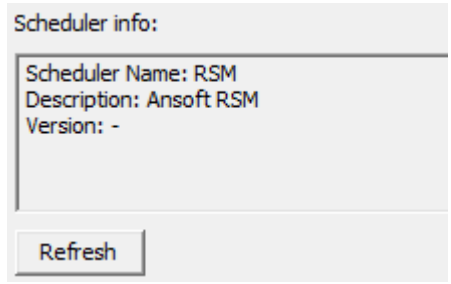


Click Select Scheduler to display the selection dialog. A drop down lists potential schedulers, (which can include RSM, lsf, Windows HPC, or sge, depending on the environment).



If you select a scheduler that is not supported in your environment, you receive a warning message.

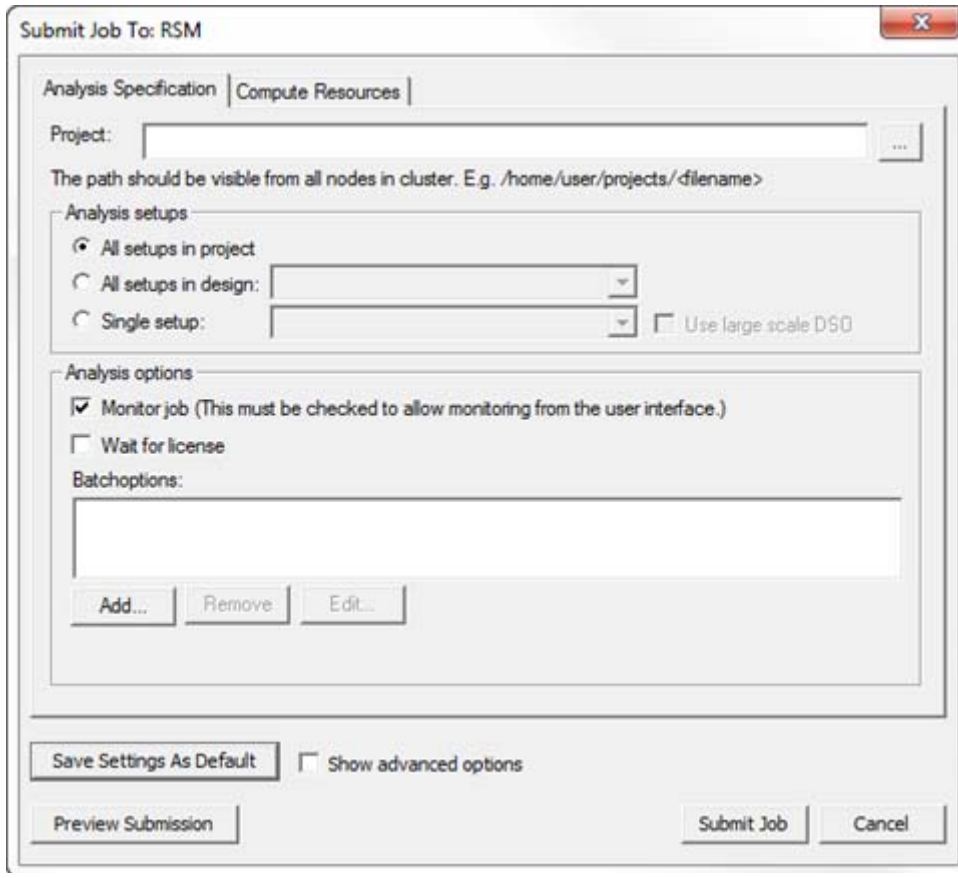
After selecting a scheduler, you can click Refresh to display information for that scheduler.



Once you have selected a scheduler supported in your environment, you can go through the following steps to submit a batch job.

1. Setup and prepare model on local workstation
2. Copy the input project (or folder, if the project references external files) from a personal workstation to a shared-drive on cluster (say project is copied to /home/projects/spool/test.adsn).
   - In the RSM environment, you are required to specify a machine-list. (See the HPC and Analysis Options.) For example, say the machine-list is: 3 cores from 'm1' and 3 cores from 'm2', for a total of 6 engines. You select the list on the **Compute Resources** tab of the **Submit Job to RSM** dialog, as described below.
3. Open a remote-desktop session (or equivalent such as vnc session) on the node corresponding to the first machine of job's machine-list, 'm1' in this case. Launch Desktop graphically on 'm1'.

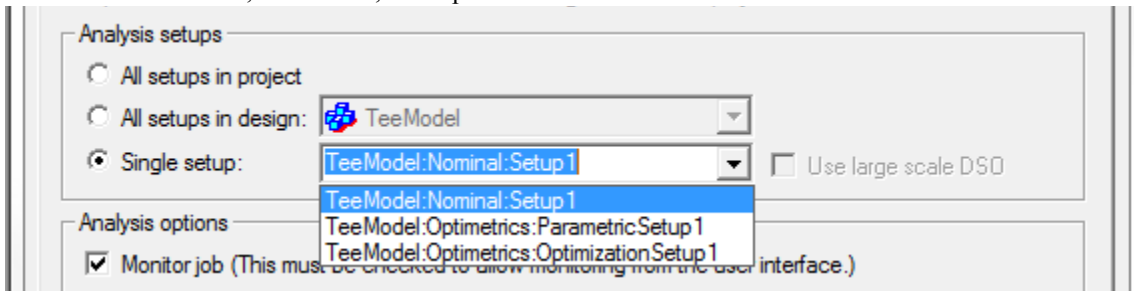
4. Run **Tools>Job Management>Submit Job...** The standard Job Submission dialog.



- The **Analysis Specification** tab has parameters to specify the input project model, the analysis setup and analysis options (including batchoptions) that affect analysis algorithms.
- The **Compute Resources** tab specifies the amount of compute resources and how to select specific resources from the available pool.

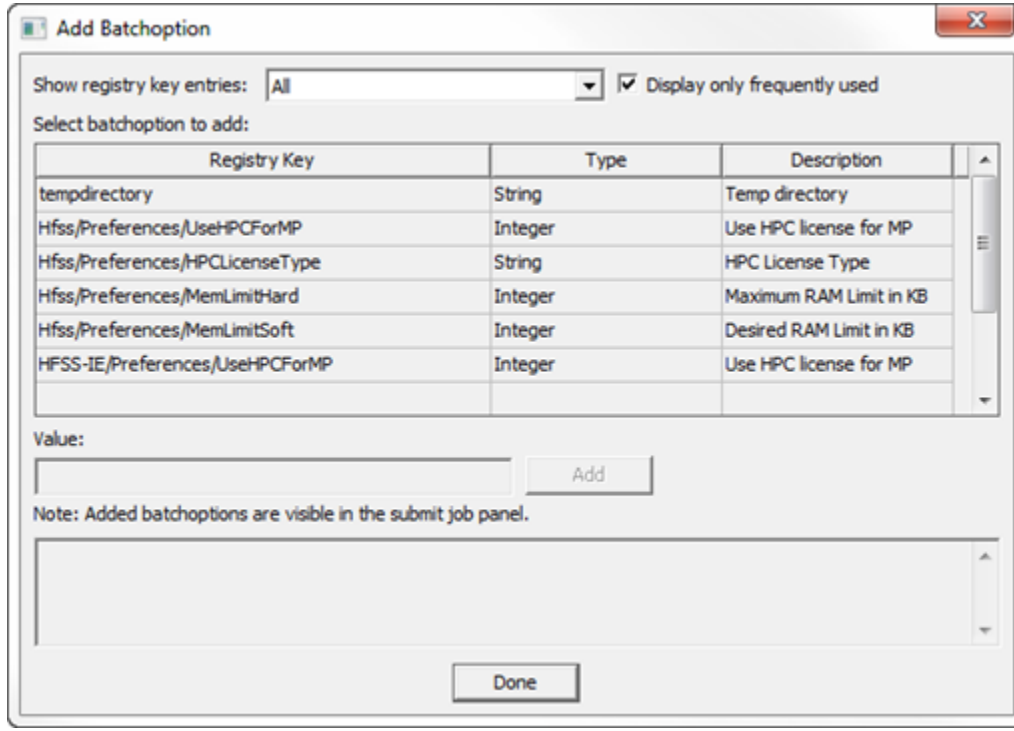
5. Use the ellipsis button [...] to open a browser to select the project.
6. In the Analysis setups field, you can select radio buttons for All setups in the project, All setups in the design, or a single setup. For instance, the OptimTee example includes setups for

Nominal, Parametric, and Optimization.

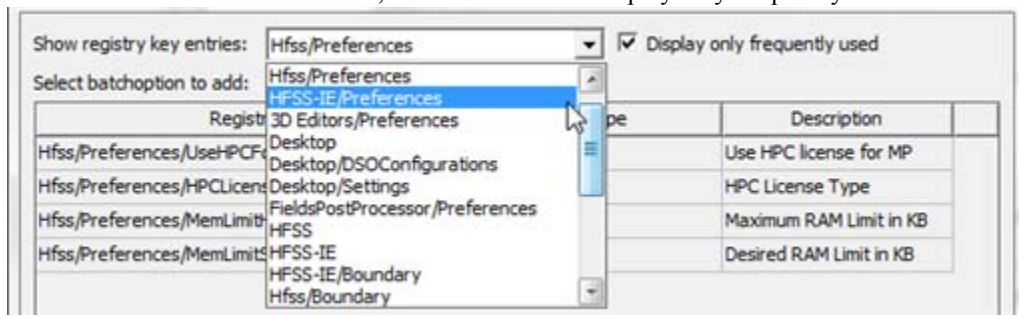


For Parametric setups, you have the option to select Use Large Scale DSO. For details on how and when you use this feature, see Job Management Interface for Large Scale DSO.

7. The Analysis options includes checkboxes for Monitoring the job, whether to wait for a license, and a field for adding Batchoptions. via a graphical interface, or as text.
   - If you intend to Monitor the job through a user interface, you must check Monitor job. You can then monitor this job through the **Tools>Job Management>Monitor Jobs...** command or by checking the dialog that opens when you submit the job.
   - The Batchoptions field allows you to add additional -batchoptions parameters, either as text, or by using a dialog with selection menus. Click the **Add** button to view the **Add Batchoption** dialog.

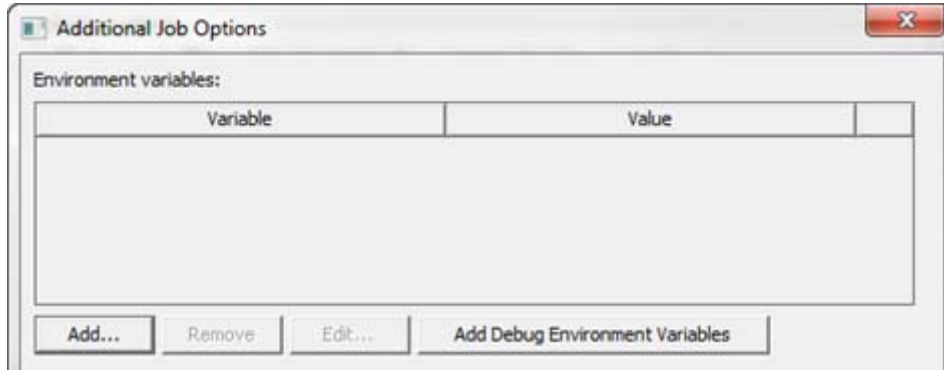


The Show registry key entries field lets you filter the entries displayed, by means of drop down menu selection, and a checkbox to Display only frequently used entries.



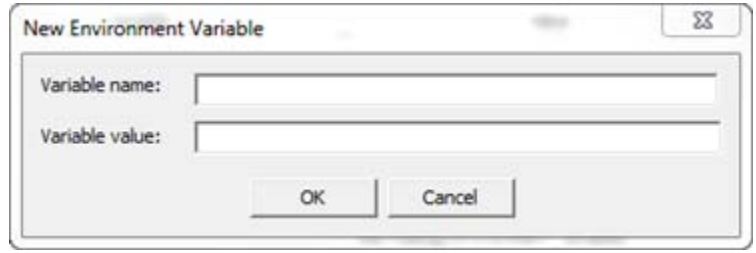
- When you have selected a batchoption, you can type the value in the field, and click the **Add** button to add the option to the batchcommand.
- In the **Submit Job To:** dialog, you can enable Show advanced options to display an additional, field Environment.

This field is for environment variables, for instance, for debugging features or other variable controlled features. Click the ellipsis [...] button to open a dialog for **Additional Job Options**.



The **Add...** button opens a New Environment Variable dialog in which you can include a vari-

able name and value.

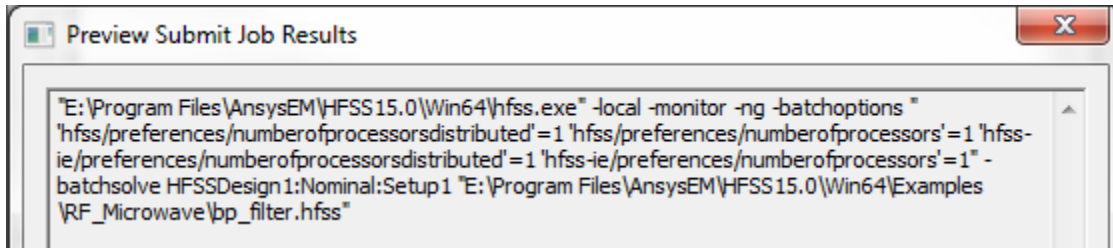


Clicking the **Add Debug Environment Variables** button automatically adds a set of debug variables. This can be useful in working with ANSYS Application Engineering support.

Variable	Value
ANSOFT_DEBUG_MODE	2
ANSOFT_DEBUG_LOG_SEPARATE	1
ANSOFT_DEBUG_LOG	$PROJECTFILEDIR\debug_log\log
ANSOFT_PASS_DEBUG_ENV_TO_REMOTE_EN...	1

Selecting a variable in the dialog enables the Remove and Edit buttons. The Edit button opens a dialog where you can change the variable and value.

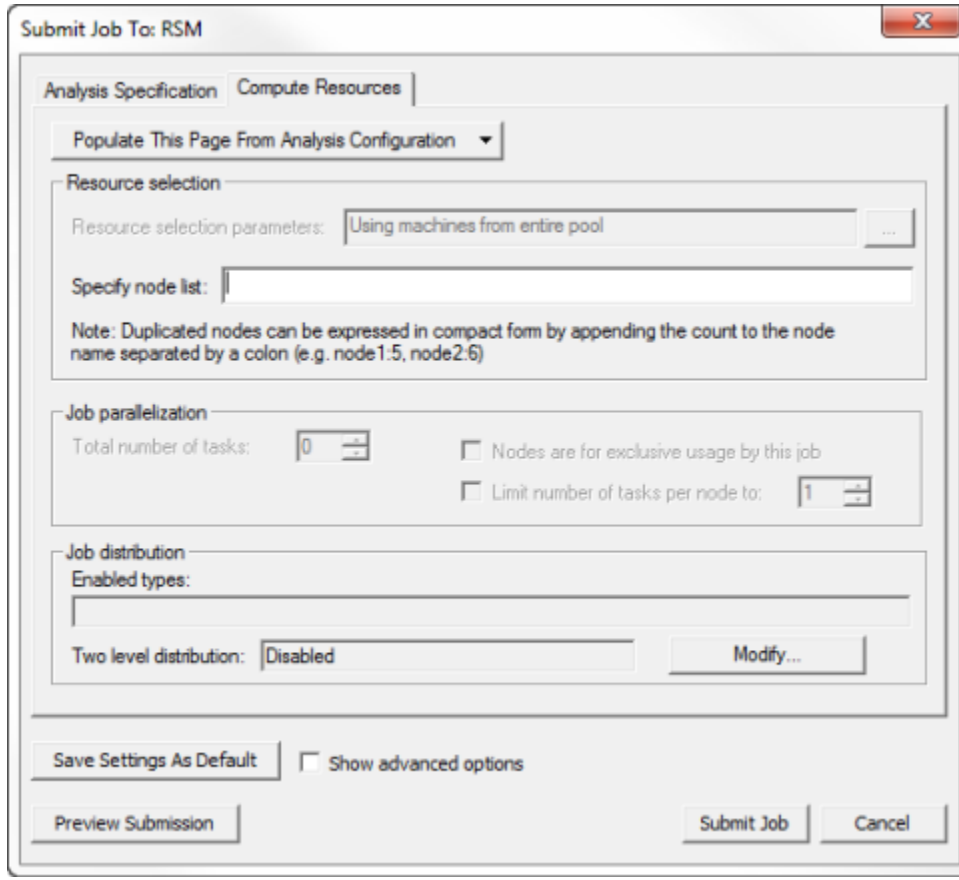
8. To see the command-line to be submitted to the scheduler, click **Preview Submission.** This opens a dialogue showing the command to be sent to the scheduler.



The text can be copied to the clipboard, if desired.

9. The **Compute Resources** tab of the **Submit Job to: RSM** dialog displays other parameters. Depending on the resources available for a scheduler environment, some of the fields may be

disabled.



- You can select **Populate This Page from Analysis Configuration**, which offers a drop-down menu of existing configurations. See Configuring Remote Simulations.

Otherwise:

- Specify node list

  Her you can specifying a node list. In a computing environment where the available cores are not uniform, you can use this to have control over which resources your job will use.

- Job parallelization:

  The values you specify here represent minimal requirements for each condition that can interact in leading to the total resources the Scheduler derives from them. If you specify a node list in the Resource selection area, that takes priority over any values specified in Job parallelization (which are then ignored). Depending on the resources available for a scheduler environment, some of the fields may be disabled.

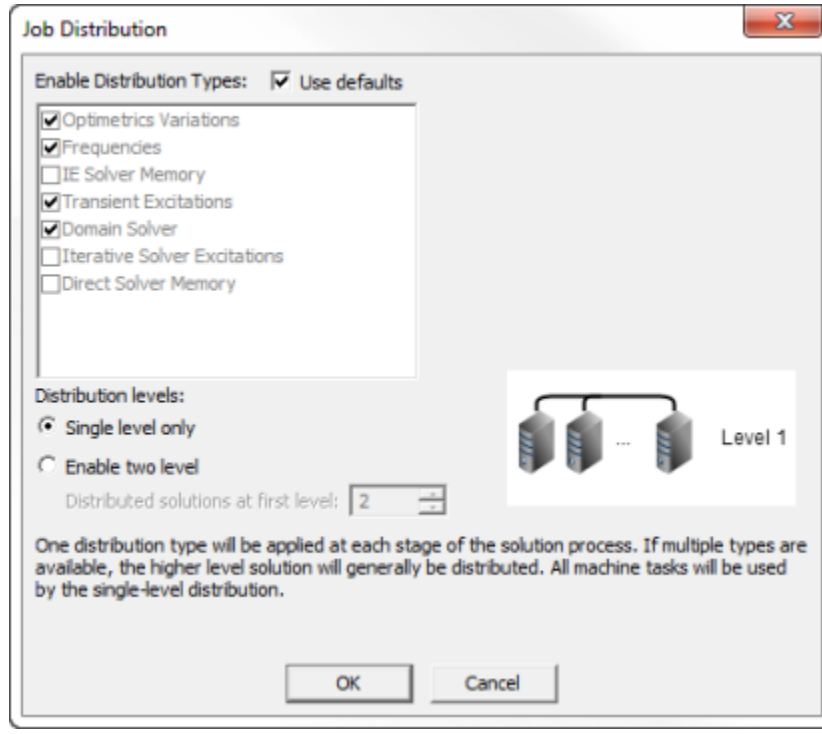
- Total number of tasks

- Whether Nodes are for exclusive usage by this job
- Whether to Limit number of tasks per node to a value

  This can be useful in situations where the amount of memory available for node is limited relative to the requirements for the project, and you want to ensure sufficient memory per process.

Job Distribution

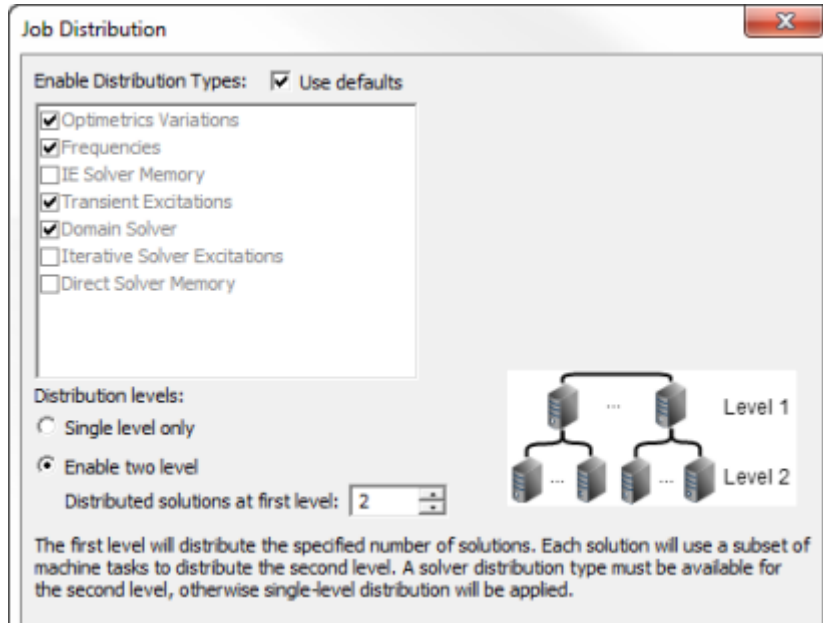
- Enabled types, such as Variations, Frequencies, Transient Excitations, Solver Domains, Direct Solver and Iterative Solver.
- Two level distribution, which may be disabled. Click the **Modify** button to display the **Job Distribution** dialog.



Enabled Distribution types can modified here.

Second level distribution operates within DSO. If available and enabled you can specify a

number of engines for level 1.

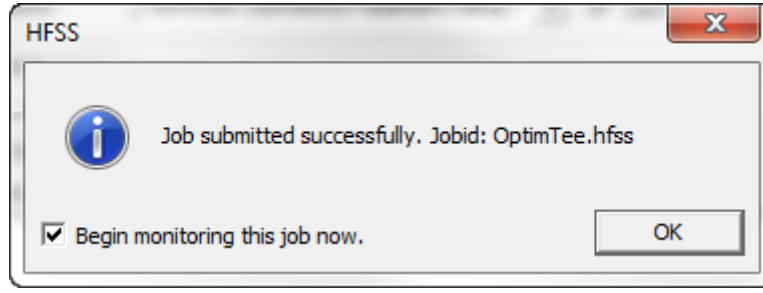


In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines, with two processors per engine, and also limit the number of engines per node to 4, the scheduler may increase the number of cores used in order to meet the limit specified for engines per node. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.

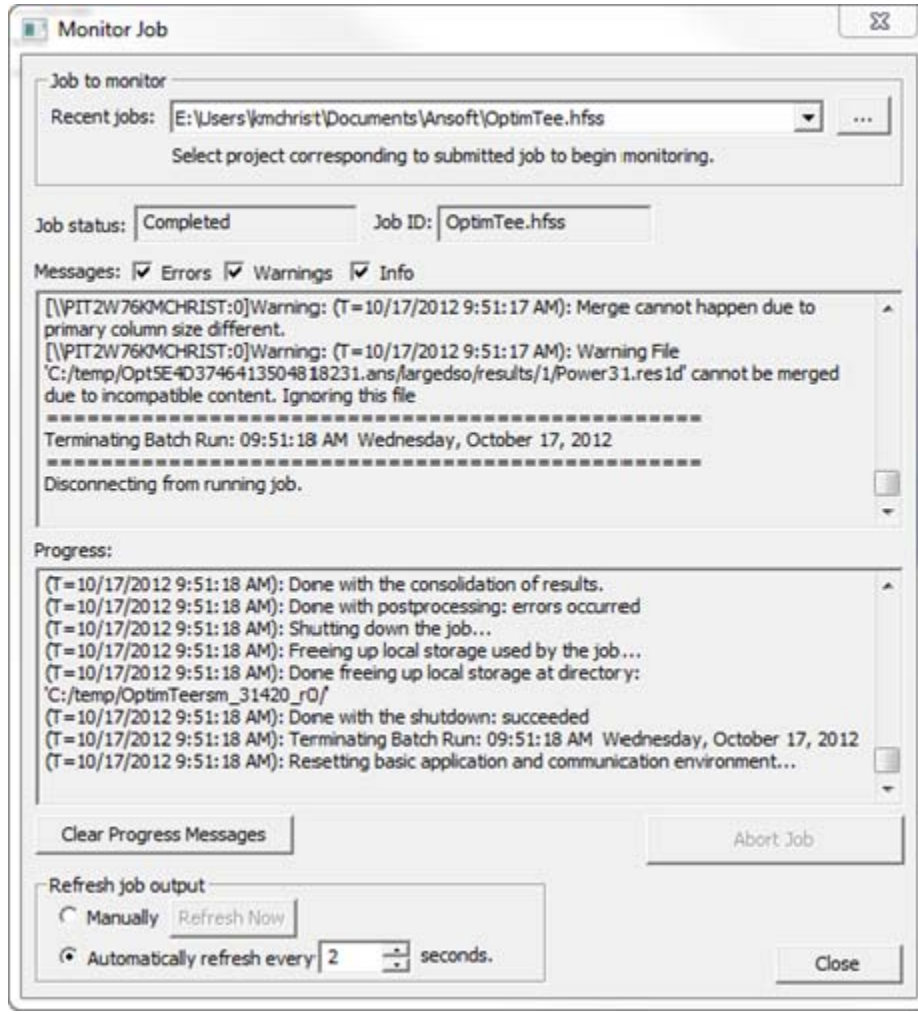
10. To submit the command with the specified parameters, click **Submit Job**.

Note The RSM environment does not support for queuing, so 'Submit Job' will immediately start running the job.

A dialog displays in which you can check "Begin monitoring this job now."



11. You can monitor this job either automatically (by checking the option) or through the **Tools>Job Management>Monitor Jobs...** command.



The dialog contains fields reporting the job status, job ID, messages issues, and progress. You can filter the messages for Errors, Warnings, and Info. By option you refresh the job manually or automatically at specified intervals.

Changing a Solution Priority for System Resources

You can modify the priority of HFSS simulations so that system resources are allocated to other computer processes before the solver. If you reduce the priority of HFSS simulations, your other software tools will respond as they normally would, but HFSS simulations may take longer.

Note The Windows Task Manager does not indicate a reduced priority for the HFSS solver. It only lists the priority of the engine manager, which appears normal, not the actual engine. The actual engine is in a separate thread, whose priority is not visible in the Windows Task Manager.

To change the priority of simulations for the system's resources:

1. While a solution is running, right-click the **Progress** window, and click **Change Priority** on the shortcut menu.
   - To affect priority for future simulation runs, click the **Tools>Options>HPC and Analysis** dialog box, and click the **Options** tab.
2. From the **Change Priority** menu (or the **Default Process Priority** pull-down menu), select one of the following priorities:

   Lowest Priority

   Below Normal

   Normal The default.

   Above Normal

   Highest

3. Click **OK**.

Aborting an Analysis

To end the solution process before it is complete:

- Right-click In the **Progress** window and click **Abort**.

  HFSS ends the analysis immediately.

The data for the currently solving pass or frequency point is deleted. All previously solved solutions are retained. For example, if you abort between the third and fourth adaptive pass, the solutions for the third pass will be available, and any solutions for the fourth pass are discarded.

To abort the solution process after the current adaptive pass or solved frequency point is complete:

- Right-click the **Progress** window, and click **Clean Stop** on the shortcut menu.

  HFSS ends the analysis after the next solved pass or frequency point.

If you request a clean stop between the third and fourth adaptive pass, the solutions for the third and fourth pass will be available once the fourth pass has finished solving.

ANSYS EM Application as an LSF Job

If you have an ANSYS EM application running as an LSF job, you can use the command "bkill -s SIGTERM *jobid*" to terminate that application. Here *jobid* is the LSF job id. The response will be "Job <jobid> is being signaled". The response is the same whether the job is actually being signaled or not.

This works correctly on a 32 bit version of Windows XP, using a 32 bit version of LSF.

In cases where the SIGTERM parameter is ignored, the command kills the LSF job, but does not clean the lock files, and other files may not be in a consistent state. See http://www.vital-it.ch/support/LSF/programmer/advanced.html for a detailed description under *Signal Handling in Windows*.

Unix/Linux

For UNIX/Linux, you can use TERM commands. Sigterm handling for Unix is done in Desktop library. You can abort a running batchsolve on Unix by sending a TERM signal to hfss.exe

Related Topics

Integration with Platform's Load Sharing Facility (LSF)

Re-solving after Modifying a Design

In some cases, if you modify a design after generating a solution, the solution in memory will no longer match the design. In such cases you receive a warning message that "Solutions have been invalidated. Undo to recover."

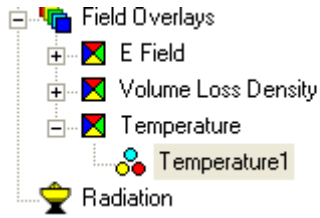
To generate a new solution after modifying a design, follow the procedure for running a simulation:

Also see Re-Solving with ANSYS Workbench Thermal Feedback.

Re-solving after ANSYS Workbench Thermal Feedback

With the Enable Feedback box in Setting the Temperature of Objects dialog is checked, you can manage analysis with feedback in ANSYS Workbench 14. After solving an HFFS 14 or Maxwell design, after performing the corresponding linked thermal analysis in ANSYS Workbench 14, you can receive a temperature distribution back from the thermal solution. ANSYS Workbench 14 will write the feedback files directly to the HFFS or Maxwell Project Solution directory.

After an analysis that includes thermal feedback from ANSYS Workbench 14, you can see temperature changes expressed in Temperature field overlays (both visually in the overlay and in the color key) as well as in the Solution data.



In the **Solution** data Profile tab you will see a new entry for Maximum Delta T, for the change in temperature from the previous simulation. The solver calculates delta in the first iteration by comparing the temperature distribution output from thermal with the initial temperature setting in HFSS/Maxwell. Subsequent simulation iterations provide a number for the temperature delta.

Solver MCS4	00:00:00	00:00:00	31.4 M	Disk = 0 KBytes, matrix size 3468 , matrix bandwidth
Field Recovery	00:00:00	00:00:00	31.4 M	Disk = 310 KBytes, 2 excitations
				Maximum Delta T = 7.0685

This simulation feedback loop from Ansoft to ANSYS Workbench 14 and back can continue until you decide that Temperature delta reported in the Solution Report low and stable for the designs.

Related Topics

Setting the Temperature of Objects

ANSYS Workbench Integration Overview

16

Optimetrics

Optimetrics enables you to determine the best design variation among a model's possible variations. You create the original model, the *nominal design*, and then define the design parameters that vary, which can be nearly any design parameter assigned a numeric value in HFSS. For example, you can parameterize the model geometry or material properties. You can then perform the following types of analyses on your nominal HFSS design:

Parametric In a parametric analysis, you define one or more *variable sweep definitions,* each specifying a series of variable values within a range. For example, you can parameterize component values. (See Variables in HFSS for more information.) Optimetrics solves the design at each variation. You can then compare the results to determine how each design variation affects the performance of the design. Parametric analyses are often used as precursors to optimization solutions because they help to determine a reasonable range of variable values for the optimization analysis.

Optimization For an optimization analysis, you identify the cost function and the optimization goal. Optimetrics changes the design parameter values to meet that goal. The cost function can be based on any solution quantity that HFSS can compute.

Sensitivity In a sensitivity analysis, you use Optimetrics to explore the vicinity of the design point to determine the sensitivity of the design to small changes in variables.

Tuning Tuning allows you to change variable values interactively while monitoring the performance of the design. If you want to ensure that tuning does not resolve variations already solved by parametric setup, you must check **Save Fields Mesh** in the **Options** tab of the optimetrics setup.

Statistical In a statistical analysis, you use Optimetrics to determine the distribution of a design's performance, which is caused by a statistical distribution of variable values.

Note Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

The HPC and Analysis Options dialog can be accessed from the setup dialog for each type of Optimetrics analysis.

You can also Link to ANSYS Design Xplorer. This permits you to manage an HFSS Optimetrics simulation from the ANSYS Workbench.

Related Topics

Getting Started Guide: Optimizing Waveguide T-Junction

Setting up a Parametric Analysis

Setting up an Optimization Analysis

Setting up a Sensitivity Analysis

Tuning a Variable

Setting up a Statistical Analysis

Parametric Overview

Optimization Overview

Sensitivity Analysis Overview

Statistical Analysis Overview

Tuning Overview

Using Distributed Analysis

Parametric Overview

Running a parametric analysis enables you to simulate several design variations using a single model. You define a series of variable values within a range, or a variable sweep definition, and HFSS generates a solution for each design variation. You can then compare the results to determine how each design variation affects the performance of the design.

You can vary design parameters that are assigned a quantity, such as geometry dimensions, material properties, and boundary and excitation properties. (See the online help topic for the specific parameter you want to vary.) The number of variations that can be defined in a parametric sweep setup is limited only by your computing resources.

To perform a parametric analysis, you first create a nominal design. A nominal design is created like any other design, except that variables are assigned to those aspects of the model you want to change. You can create a parametric setup before defining variables but all variables must be defined before you start the parametric analysis. Although you are not required to solve the nominal design before performing a parametric analysis, doing so helps ensure that the model is set up and operates as intended. Alternatively, you can perform a validation check on the nominal design before performing a parametric analysis.

Parametric analyses are often used as precursors to optimization analyses because they enable you to determine a reasonable range of variable values for an optimization analysis.

Related Topics

Setting up a Parametric Analysis

Determining Phase Center using Optimetrics

Large Scale DSO for Parametric Analysis

Setting Up a Parametric Analysis

A *parametric setup* specifies all of the design variations that Optimetrics drives HFSS to solve. A parametric setup is made up of one or more *variable sweep definitions*, which are a set of variable values within a range that you want HFSS to solve when you run the parametric setup.

You can define more than one parametric setup per design.

Note Once you have created a parametric setup, you can copy and paste it, and then make changes to the copy, rather than redoing the whole process for minor changes.

To add a parametric setup to a design:

1. Click **HFSS** or **HFSS**>**Optimetrics Analysis**> **Add Parametric**  .
   - Alternatively, right-click **Optimetrics** in the project tree, and then click **Add**>**Parametric** on the shortcut menu.

   The **Setup Sweep Analysis** dialog box appears.
2. Add a variable sweep definition.

After you define a parametric sweep, a shortcut menu becomes available when you right-click the setup name.

Note Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

Related Topics

Adding a Variable Sweep Definition

Specifying a Solution Setup for a Parametric Setup

Using Distributed Analysis

Parametric Overview

Adding a Parametric Sweep from a File

Large Scale DSO for Parametric Analysis

Adding a Variable Sweep Definition

A parametric setup is made up of one or more *variable sweep definitions*. A variable sweep definition is a set of variable values within a range that Optimetrics drives to solve when the parametric setup is analyzed. You can add one or more sweep definitions to a parametric setup.

Note Sweeping a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

1. Click **HFSS** or **HFSS**>**Optimetrics Analysis**>**Add Parametric**  .
   - Alternatively, right-click **Optimetrics** in the project tree, and then click **Add**>**Parametric** on the shortcut menu.

The **Setup Sweep Analysis** dialog box appears.

2. Under the **Sweep Definitions** tab, click **Add**.

   The **Add/Edit Sweep** dialog box appears.

   All the independent variables associated with the design are listed in the **Variable** pull-down list.

3. Click the variable for which you are defining the sweep definition from the **Variable** pull-down list.

   If you do not define a sweep definition for a variable in the list, the variable's current value in the nominal design is used in the parametric analysis.

4. Specify the variable values to be included in the sweep.
5. Click **Add**, and then click **OK**.

   You return to the **Setup Sweep Analysis** dialog box. The variable sweep is listed in the top half of the window.

6. View the design variations that are to be solved in table format under the **Table** tab. Viewing the sweep definition in table format enables you to visualize the design variations that are to be solved and manually adjust sweep points if necessary.
7. Click **OK**.

Related Topics

Specifying Variable Values for a Sweep Definitions

Synchronizing Variable Sweep Definitions

Modifying a Variable Sweep Definition Manually

Overriding a Variable's Current Value in a Parametric Setup

Adding a Parametric Sweep from a File

Specifying Variable Values for a Sweep Definition

To specify the variable values to include in a sweep definition:

1. Select one of the following in the **Add/Edit Sweep** dialog box:

Single value	Specify a single value for the sweep definition.
Linear step	Specify a linear range of values with a constant step size.
Linear count	Specify a linear range of values and the number, or count of points within this range.
Decade count	Specify a logarithmic (base 10) series of values, and the number of values to calculate in each decade.
Octave count	Specify a logarithmic (base 2) series of values, and the number of values to calculate in each octave.
Exponential count	Specify an exponential (base e) series of values, and the number of values to calculate.

2. If you selected **Single value**, type the value of the sweep definition in the **Value** box.

   If you selected another sweep type, do the following:

   a. Type the starting value of the variable range in the **Start** text box.

   b. Type the final value of the variable range in the **Stop** text box.

Warning Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

3. If you selected **Linear step** as the sweep type, type the step size in the **Step** box.

   The step size is the difference between variable values in the sweep definition. The step size determines the number of design variations between the start and stop values. HFSS will solve the model at each step in the specified range, including the start and stop values. The step size can be negative, when the **Stop** value is less than the **Start** value

   If you selected another sweep type, type the number of points, or variable values, in the sweep definition in the **Count** text box. For **Decade count** and **Octave count**, the **Count** value specifies the number of points to calculate in every decade or octave. For **Exponential count**, the **Count** value is the total number of points. The total number of points includes the start and stop values.

Related Topics

Synchronizing Variable Sweep Definitions

Synchronizing Variable Sweep Definitions

By default, variable sweep definitions are nested. Alternatively, you can synchronize the variable sweep definitions if they have the same number of sweep points.

For example, if you synchronize a sweep definition that includes values of 1, 2, and 3 inches with a second sweep definition that includes values of 4, 5, and 6 inches, HFSS will solve 3 design variations. The first variation is solved at the variable values of 1 and 4; the second variation is solved at the variable values 2 and 5; and the third variation is solved at the final variable values 3 and 6.

To synchronize variable sweep definitions:

1. Under the **Sweep Definitions** tab of the **Setup Sweep Analysis** dialog box, select the rows containing the sweep definitions you want to synchronize.
2. Click **Sync**.

   The synchronized sweeps are given a group number, which is listed in the **Sync #** column.

Optionally, view the design variations that are to be solved in table format under the **Table** tab.

Related Topics

Specifying Variable Values for a Sweep Definitions

Modifying a Variable Sweep Definition Manually

You can manually modify the variable values that are solved for a parametric setup by explicitly changing, adding, or deleting existing points in a variable sweep definition under the **Table** tab of the **Setup Sweep Analysis** dialog box.

To manually modify a variable sweep definition:

1. Click the **Table** tab of the **Setup Sweep Analysis** dialog box.

   The design variations HFSS solves for the parametric setup listed in table format.
2. Do one of the following:
   - To modify a variable value, click a value text box in the table and type a new value.
   - To delete a variable value from the sweep definition, click the row you want to delete, and then click **Delete**.
   - To add a new variable value to the sweep definition, click **Add**. Then click in the value text box and type a new value.

Warning Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

Your modifications are tracked and available for viewing at the bottom of the **Setup Sweep Analysis** dialog box under the **Sweep Definitions** tab. The operations you performed are listed with descriptions.

Warning If you modify an original sweep definition using the **Add/Edit Sweep** dialog box after you have manually modified its table of design variations, your manual modifications become invalid and are removed. A warning is displayed to inform you that your manual values are about to become invalid, so you can decide whether or not to proceed.

Related Topics

Adding a Variable Sweep Definition

Overriding a Variable's Current Value in a Parametric Setup

Adding a Parametric Sweep from a File

Overriding a Variable's Current Value in a Parametric Setup

If you choose not to sweep a variable HFSS uses the variable's current value set for the nominal design when it solves the parametric setup. To override the current variable value for a parametric setup:

1. In the **Setup Sweep Analysis** dialog box, click the **General** tab.

   Under **Starting Point**, all of the current independent design variable values are listed.
2. Click the **Override** box of the design variable with the value you want to override for the parametric setup.
3. Type a new value in the **Value** box, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered will be used for the parametric setup. For this parametric setup, the new value will override the current value in the nominal design.

Note Alternatively, you can select the **Override** option first, and then type a new variable value in the **Value** box.

4. Optionally, click a new unit in the **Units** box.

To revert to the current variable value, clear the **Override** option.

Warning Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

Related Topics

Adding a Variable Sweep Definition

Modifying a Variable Sweep Definition Manually

Specifying a Solution Setup for a Parametric Setup

To specify the solution setup that HFSS analyzes when it solves a parametric setup:

1. In the **Setup Sweep Analysis** dialog box, click the **General** tab.
2. Select the solution setup you want HFSS to use when it solves the parametric setup.

   HFSS solves the parametric setup using the solution setup you select. If you select more than one, results are generated for all selected solution setups.

Related Topics

Specifying the Solution Quantity to Evaluate for Parametric Analysis

Specifying a Solution Quantity's Calculation Range

Specifying the Solution Quantity to Evaluate for Parametric Analysis

When you add a parametric setup, you can identify one or more solution quantities to be presented in the **Post Analysis Display** dialog box. The solution quantities are specified by mathematical expressions that are composed of basic quantities, such as output variables. When you view the results, HFSS extracts the solution quantities and lists them in the results table.

1. In the **Setup Sweep Analysis** dialog box, click the **Calculations** tab.

   This displays a table that will show Solutions and associated Calculations. Below the table, are control buttons to **Setup Calculations...** and **Delete**.

2. Click **Setup Calculations**.

   This displays the **Add/Edit Calculation** dialog. The dialog contains panes to set the **Context**, the **Trace** tab for the **Calculation Expression**, and the **Calculation Range** tab for the **Calculation Range**.

   Follow the procedure to Setup Calculations for Optimetrics.

3. Click **Add Calculation** to add the expression in the **Add/Edit Calculation** dialog **Calculation Expression** field to the Calculations tab of the **Setup Sweep Analysis** dialog.

4. Click **Done** to close the **Add/Edit Calculation** dialog.

Related Topics

Specifying a Solution Quantity's Calculation Range

Specifying Output Variables

Setup Calculations for Optimetrics.

Setup Calculations for Optimetrics

The **Setup** dialogs for each of the Optimetrics types include a **Setup Calculations** button. Clicking this displays the **Add/Edit Calculation** dialog box. The dialog box contains distinct panes and tabs to set the **Context**, the **Calculation Expression**, and the **Calculation Range**.

The **Context** pane contains fields for the Report Type to use, the Solution, and depending on the Report Type selection, the Geometry.

The **Trace** tab contains fields for the Calculation expression, and, to build the expression, a Category list, a Quantity list with a Text Filter field, and a list of Functions available for the selected Category. The Range function button opens a dialog in which you can define a range function to apply a function to the expression.

The Category list for the **Trace** tab includes Variables and Output Variables. An Output Variables... button lets you open a dialog box to define and edit the Output Variables.

To setup an Optimetrics calculation:

1. Click the **Setup Calculations** button to open the **Add/Edit/Calculation** dialog.
2. In the **Report Type** text field in the **Context** pane, select from the drop down list of available types.

   Selecting Fields as the Report type causes the **Geometry** field to display.
3. In the **Solution** text box, select from the drop down list of available solutions.
4. If the **Geometry** field is available, select from the drop down list.
5. In the **Trace** tab, specify the solution Category, a Quantity, and Functions. The resulting expression will be displayed in the **Calculation Expression** field.
   a. Select the **Category** from the list.

      The selection appears in the **Calculation Expression** field, and the Quantity and Function fields list what is available for the corresponding selection.
   b. Select the **Quantity** from the list.

      The selected quantity appears in the **Calculation Expression** field.

      If the **Quantity** list is long, you can filter it for easier selection by typing in the text filter field. Only quantities that contain those alphanumeric characters anywhere in their name will remain visible in the list.

      If you want to create an output variable that represents the solution quantity, do the following:

      - Click the **Output Variables** button.

        The **Output Variables** dialog box appears.
      - Add the expression you want to evaluate, and then click **Done**.

        The recently created output variable appears in the **Quantity** list.
      - Click a new output variable in the Quantity pull-down list.

   Note The calculation you specify must be able to be evaluated into a single, real number.

      The selected Quantity appears in the **Calculation Expression** field.
   c. Select the **Function** from the list.

      The selected function is applied to the **Quantity** in the **Calculation Expression** field.
6. To apply a **Range function** to the **Calculation Expression**, see Setting a Range function.
7. Click **Add Calculation** to add the expression in the **Add/Edit Calculation** dialog **Calculation Expression** field to the Calculations tab of the **Setup Sweep Analysis** dialog.
8. Click **Done** to close the **Add/Edit Calculation** dialog box.

Related Topics

Specifying a Solution Quantity to Evaluate

Setting a Range function

Specifying a Solution Quantity's Calculation Range

The calculation range of a solution quantity determines the value of intrinsic variables such as frequency (F) at which the solution quantity will be extracted. For a parametric setup, the calculation range must be a single value. For a Driven Modal or Driven Terminal design, if you selected to extract the solution from the last adaptive solution, Optimetrics uses the adaptive frequency defined in the solution setup. If you selected to extract the solution quantity from a frequency sweep solution, Optimetrics by default will use the starting frequency in the sweep.

1. In the **Setup Sweep Analysis** dialog box, click the **Calculations** tab.
2. Click the **Setup Calculations** button.

   The **Add/Edit/Calculation** dialog box appears.
3. Select the **Calculation Range** tab.
4. In the **Variable** list, click an intrinsic variable.

   Single Value is selected by default.
5. In the **Value** box, click a value at which the solution quantity will be extracted.
6. Click **Update**, and then click **Edit**.

Viewing Results for Parametric Solution Quantities

1. In the project tree, right-click the parametric setup for which you want to view the results calculated for the solution quantities, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Select the parametric setup with the results you want to view from the pull-down list at the top of the dialog box.
3. Make sure the **Result** tab is selected on the dialog.
4. To view the results in tabular form, select **Table** as the view type.

   The results for the selected solution quantities are listed in table format for each solved design variation.
5. Optionally, select **Show complete output name**.

   The complete name of the solution for which the results are being displayed will be listed in the column headings.
6. Optionally, click a design variation in the table, and then click **Apply** (at the far right side of the dialog box).

   HFSS now points to the selected design variation as the nominal solution and as a result, the design displayed in the **Modeler** window is changed to represent the selected design variation.

   Click **Revert** to return the design in the view window to the original value.
7. To view the results in graphic form, select **Plot** as the view type.
8. Select the variable with the swept values you want to plot on the x-axis from the **X** pull-down list.
9. Only one sweep variable at a time can be plotted against solution quantity results. Any other

variables that were swept during the parametric analysis remain constant.

Optionally, to modify the constant values of other swept variables, do the following:

a. Click **Set Other Sweep Variables Value**.

The **Setup Plot** dialog box appears. All of the other solved variable values are listed.

b. Click the row with the variable value you want to use as the constant value in the plot, and then click **OK**.

10. Select the solution quantity results you want to plot on the y-axis from the **Y** pull-down list.

The x -y plot appears in the view window.

You can modify the display by right-clicking in the graph area. See Creating Reports for details on these operations.

11. To view profile information about the analysis, click the **Profile** tab on the **Post Analysis Display** dialog.
12. When more than one parametric analysis has been run, use the left and right arrows to select a profile.
13. Click **Close** to close the **Post Analysis Display** dialog.

Related Topics

Plotting Solution Quantity Results vs. a Swept Variable

Viewing Solution Data for an Optimetrics Design Variation

Using Distributed Analysis

If you have purchased the appropriate license, HFSS supports distributed solve, which involves distributing rows of a parametric table during Optimetrics solve.

If you do a distributed solve, HFSS launches solver engines on multiple machines, assuming that you have configured your HPC and Analysis Options correctly. Also see Large Scale DSO for Parametric Analysis.

To run a distributed analysis:

1. Under **Optimetrics** in the project tree, right-click the specific parametric setup.
A shortcut menu appears.
2. Select **Analyze** from the shortcut menu.

Note After you define a parametric sweep, a shortcut menu becomes available when you right-click the setup name.

While the analysis is running, you can access parent and child progress bars. By default, only the main progress bar is displayed, while the child progress bars (or subtasks) remain hidden. You can toggle between showing and hiding the child progress bars.

To show the child progress bars:

- Right-click the progress window, and select **Show Subtask Progress Bars**.

To hide the child progress bars:

- Right-click the progress window, and select **Hide Subtask Progress Bars**.

Related Topics

Setting HPC and Analysis Options

Setting Up Distributed Analysis with Licensing

Adding a Parametric Sweep from a File

You can specify the parameters for a parametric sweep in a spreadsheet that uses either a .csv (comma delimited) or .txt (tab delimited) format. You can then import the parametric sweep using the **HFSS>Optimetrics>Add Parametric from File** command.

For example, a .txt spreadsheet file could resemble the following:

```
a          $b        $c[in]    d[m]      $e        $f
0.1 mil    2mm       11        21        0.6in     8
0.2mil     3mm       1.3       2.6mm     3         9cm
...
```

The first row lists the Project and Design Variable names, and when followed by parentheses, the units. The following rows provide the variable values and units. Project or Design variables must be defined before they are accepted from a file. The characters in variable names are not case sensitive. Consecutive separators are treated as one separator.

The header row also takes units in ( ) as well as the conventional [ ].

Related Topics

Setting up a Parametric Analysis

Optimization Overview

Optimetrics interfaces with ANSYS Electromagnetics products to help you optimize a wide variety of design parameters based on variable geometry, materials, excitations, component values, etc. Optimization is the process of locating the minimum of a user-defined cost function. Optimetrics modifies the variable values until the minimum is reached with acceptable accuracy.

Related Topics

Setting Up an Optimization Analysis

Choosing an Optimizer

Choosing an Optimizer

Conducting an optimization analysis allows you to determine an optimum solution for your problem. In HFSS optimization analyses, you have choices of optimizer, though in most cases, the Sequential Nonlinear Programming (Gradient) optimizer is recommended.

- Sequential Nonlinear Programming (Gradient)
- Sequential Mixed Integer Non Linear Programming (Gradient and Discrete)
- Quasi Newton (Gradient)
- Pattern Search (Search-based)
- Genetic Algorithm (Random Search)
- MATLAB

All optimizers assume that the nominal problem you are analyzing is close to the optimal solution; therefore, you must specify a domain that contains the region in which you expect to reach the optimum value.

All optimizers allow you to define a maximum limit to the number of iterations to be executed. This prevents you from consuming your remaining computing resources and allows you to analyze the obtained solutions. From this reduced range, you can further narrow the domain of the problem and regenerate the solutions.

All optimizers also allow you to enter a coefficient in the **Add Constraints** window to define the linear relationship between the selected variables and the entered constraint value. For the SNLP and SMINLP optimizers, the relationship can be linear or nonlinear. For the Quasi Newton and Pattern Search optimizers, the relationship must be linear.

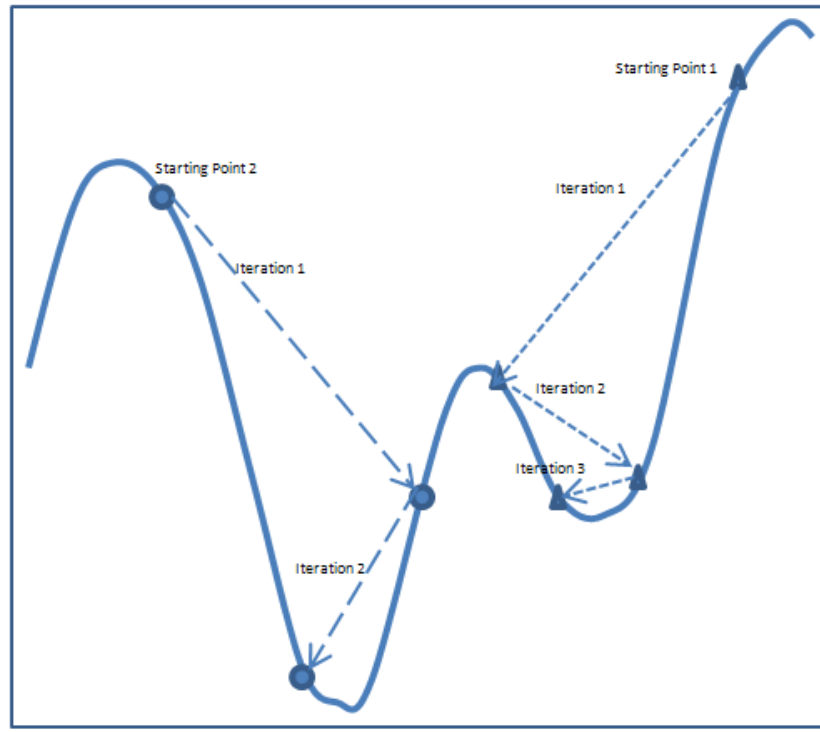
Quasi Newton (Gradient)

If the Sequential Non Linear Programming Optimizer has difficulty, and if the numerical noise is insignificant during the solution process, use the Quasi Newton optimizer to obtain the results. The Quasi Newton optimizer works on the basis of finding a minimum or maximum of a cost function which relates variables in the model or circuit to overall simulation goals. The user marks one or more variables in the project and defines a cost function in the optimization setup. The cost function relates the variable values to field quantities, design parameters like force or torque, power loss, etc. The optimizer can then maximize or minimize the value of the design parameter by varying the problem variables.

Sir Isaac Newton first showed that the maximum or minimum of any function can be determined by setting the derivative of a function with respect to a variable (x) to zero and solving for the variable. This approach leads to the exact solution for quadratic functions. However, for higher order functions or numerical analysis, an iterative approach is commonly taken. The function is approximated locally by a quadratic and the approximation is solved for the value of x. This value is placed back into the original function and used to calculate a gradient which provides a step direction and size for determining the next best value of x in the iteration process.

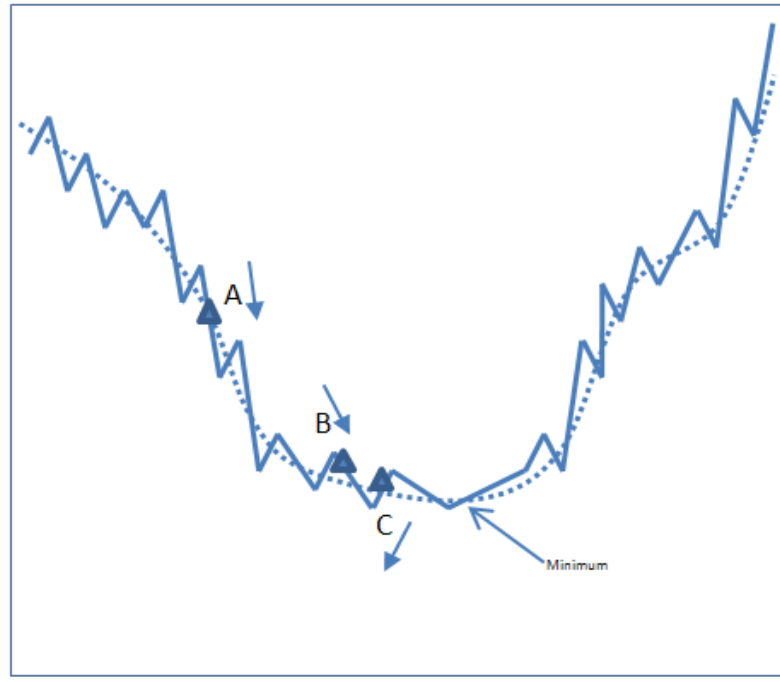
In the Quasi-Newton optimization procedure, the gradients and Hessian are calculated numerically. Essentially, the change in x and the change in the gradient are used to estimate the Hessian for the next iteration. The ratio of the change in cost to the change in the values of x provides the gradient, whereas, the ratio of the change in the gradients to the change in the values of x provides the Hessian for the next step and is know as the quasi-Newton condition. In order to perform the Quasi-Newton optimization, at least three solutions are required for each parameter being varied. This can have a significant computational cost depending upon the type of analysis being performed.

There are numerous methods described in the literature for solving for the Hessian and the details of the method used by Optimetrics are beyond the scope of this document. However, as the Quasi-Newton method is, at its heart, a gradient method, it suffers from two fundamental problems common to optimization. The first is the possible presence of local minima. The figure below illustrates the problem of local minima. In this scenario, you can see that in order to find the minimum of the



function over the domain, a number of factors will determine the overall success including the initial starting point, the initial set of gradients calculated, the allowable step size, etc. Once the optimizer has located a minimum, the Quasi-Newton approach will locate the bottom and will not search further for other possible minima. In the example shown, when the optimizer begins at the point labeled "Starting Point 1" the minima it finds is a local minima and not a good global solution to the problem.

The second basic issue with Quasi-Newton optimization is numerical noise. In gradient optimization, the derivatives are assumed to be smooth, well behaved functions. However, when the gradients are calculated numerically, the calculation involves taking the differences of numbers that get progressively smaller. At some point, the numerical imprecision in the parameter calculations becomes greater than the differences calculated in the gradients and the solution will oscillate and may never reach convergence. To illustrate this, consider the figure shown below. In this scenario,



the optimizer is looking for the point labeled "minimum". Three possible solutions are labeled A, B and C, with each arrow indicating the direction of the derivative of the function at that point. If points A and B represent the last two solution points for the parameter, then it is easy to see that the changes in the magnitude and the consistent direction of the derivatives will serve to push the solution closer to the desired minimum. If, however, points A and C are the last two solution points respectively, the magnitude indicates the proper direction of movement, but the derivatives are opposite, possibly causing the solution to move away from the minimum, back in the direction of point A.

In order to use the Quasi-Newton optimizer effectively, the cost function should be based on parameters that exhibit a smooth characteristic (little numerical noise) and a starting point of the optimization should be chosen somewhat close to the expected minimum based on an understanding of the physical problem being optimized. This becomes increasingly difficult, however, when multiple parameters are being varied or when multiple parameters are to be optimized. In addition, the computational burden of multivariate optimization with Quasi-Newton increases geometrically with the number of variables being optimized. As a result, this method should only be attempted when 1 or 2 variables are being optimized as a time.

For more information regarding Quasi-Newton optimization methods, see the following reference:

Schoenberg, Ronald. *Optimization with the Quasi-Newton Method*. Aptech Systems, Inc. 2001.

Related Topics

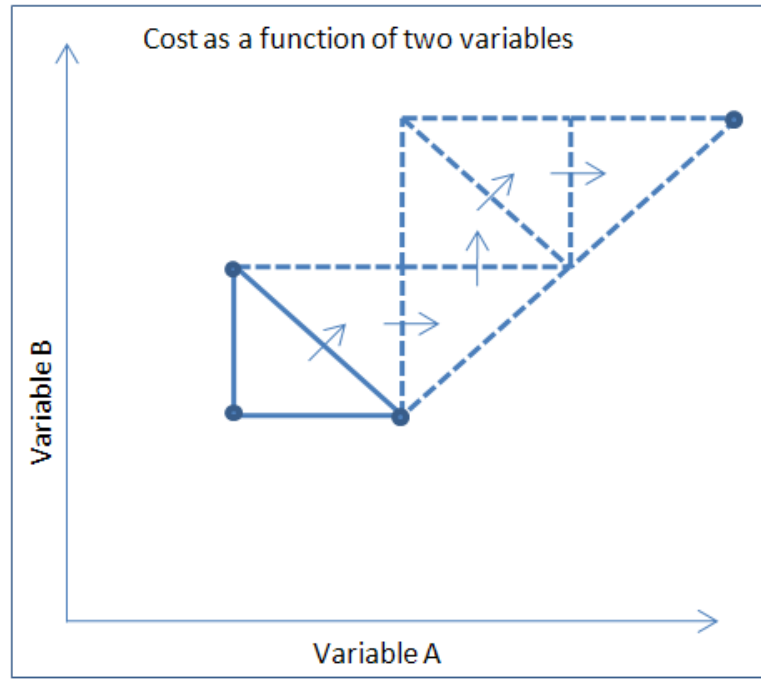
Optimization Setup for Quasi Newton Optimizer

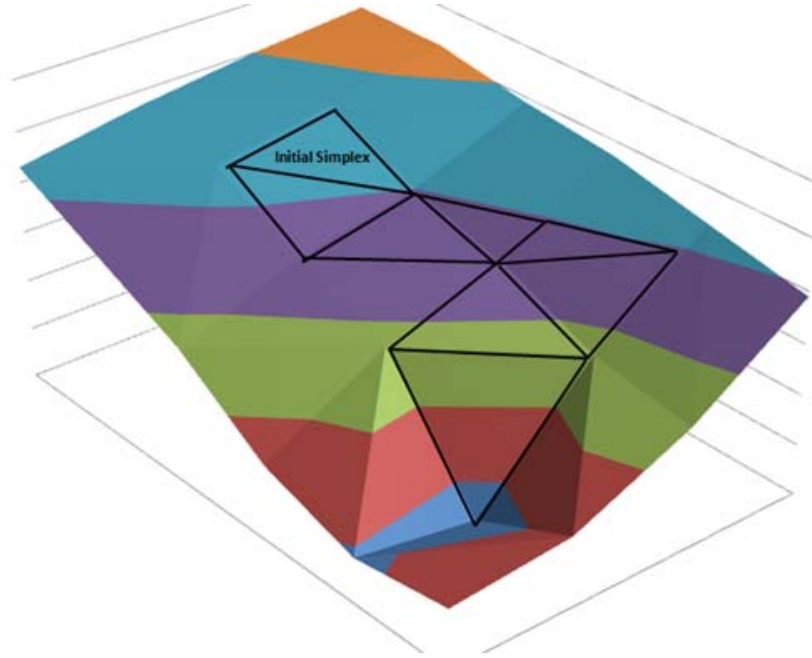
Pattern Search (Search-Based)

If the noise is significant in the nominal project, use the Pattern Search optimizer to obtain the results. It performs a grid-based simplex search, which makes use of simplices: triangles in 2D space or tetrahedra in 3D space. A simplex is a Euclidean geometric spatial element having the minimum number of boundary points, such as a line segment in one-dimensional space, a triangle in two-dimensional space, or a tetrahedron in three-dimensional space.

The cost value is calculated at the vertices of the simplex. The optimizer mirrors the simplex across one of its faces based on mathematical guidelines and determines if the new simplex provides better results. If it does not produce a better result, the next face is used for mirroring and the pattern continues. If no improvement occurs, the grid is refined. If improvement occurs, the step is accepted and the new simplex is generated to replace the original one. The figures below illustrate a triangular simplex mirrored several times to demonstrate the pattern search approach in two vari-

ables and the simplices superimposed on a 2D cost function to demonstrate the convergence toward a minimum in the cost function.





The Pattern Search algorithms are extensible to three variable optimization by using tetrahedral simplices, however, they are not easily represented in graphical form. Generally, Pattern Search algorithms are not used when more than three variables are used in the optimization.

When there is no improvement in the cost function regardless of the direction the simplex is mirrored, then the simplex is subdivided into smaller simplices and the process restarted.

Pattern Search algorithms have several advantages over Quasi-Newton algorithms. First, they are less sensitive to noise because the cost function is evaluated at all node points on the simplex and the numerical noise averages out over the simplex. The second advantage is that the number of initial solutions is generally smaller. However, since the pattern search does not use gradient information to locate the minimum the process converges more slowly toward the true minimum, taking more steps to successively divide the simplices as the minimum is approached.

Related Topics

Optimization Setup for Pattern Search Optimizer

Sequential Non linear Programming (Gradient)

The main advantage of SNLP (Gradient) over Quasi Newton (Gradient) is that it handles the optimization problem in more depth. This optimizer assumes that the optimization variables span a continuous space. As a result, there is no Minimum Step Size specified in this optimizer and the variables may take any value within the allowable constraints and within the numerical precision

limits of the simulator. Like Quasi Newton, the SNLP optimizer assumes that the noise is not significant. It does reduce the effect of the noise, but the noise filtering is not strong.

The SNLP optimizer approximates the FEA characterization with Response Surfaces (RS). With the FEA-approximation and with light evaluation of the cost function, SNLP has a good approximation of the cost function in terms of the optimization variables. This approximation allows the SNLP optimizer to estimate the location of improving points. The overall cost approximations are more accurate. This allows the SNLP optimizer a faster practical convergence speed than that of quasi Newton.

The SNLP Optimizer creates the response surface using a polynomial approximation from the FEA simulation results available from past solutions. The response surface is most accurate in the local vicinity. The response surface is used in the optimization loop to determine the gradients and calculate the next step direction and distance. The response surface acts as a surrogate for the FEA simulation, reducing the number of FEA simulations required and greatly speeding the problem. Convergence improves as more FEA solutions are created and the response surface approximation improves.

The SNLP method is similar to the Sequential Quadratic Programming (SQP) method in two ways: Both are sequential, updating the optimizer state to the current optimal values and iterating. Sequential optimization can be thought of as walking a path, step by step, toward an optimal goal. SNLP and SQP optimizers are also similar in that both use local and inexpensive surrogates. However, in the SNLP case, the surrogate can be of a higher order and is more generally constrained. The goal is to achieve a surrogate model that is accurate enough on a wider scale, so that the search procedures are well lead by the surrogate, even for relatively large steps. All functions calculated by the supporting finite element product (for example, Maxwell 3D or HFSS) is assumed to be expensive, while the rest of the cost calculation (for example, an extra user-defined expression) — which is implemented in Optimetrics — is assumed to be inexpensive. For this reason, it makes sense to remove inexpensive evaluations from the finite element problem and, instead, implement them in Optimetrics. This optimizer holds several advantages over the Quasi Newton and Pattern Search optimizers.

Most importantly, due to the separation of expensive and inexpensive evaluations in the cost calculation, the SNLP optimizer is more tightly integrated with the supporting FEA tools. This tight integration provides more insight into the optimization problem, resulting in a significantly faster optimization process. A second advantage is that the SNLP optimizer does not require cost-derivatives to be approximated, protecting against uncertainties (noise) in cost evaluations. In addition to derivative-free state of the RS-based SNLP, the RS technique also proves to have noise suppression properties. Finally, this optimizer allows you to use nonlinear constraints, making this approach much more general than either of the other two optimizers.

Related Topics

Optimization Setup for SNLP Optimizer

Sequential Mixed Integer NonLinear Programming (Gradient and Discrete)

The Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete) optimizer is equivalent to the SNLP (Gradient) optimizer with only one difference. Many problems require variables

take only discrete values. One example might be to optimize on the number of turns in a coil. To be able to optimize on number of turns or quarter turns, the optimizer must handle discrete optimization variables. The SMINLP optimizer can mix continuous variables among the integers, or can have only integers, and works if all variables are continuous. The setup resembles the setup for SNLP, except that you must flag the integer variables.supporting integer variables. You can set up internal variables based on the integer optimization variable.

For example, consider N to be an integer optimization variable. By definition it can only assume integer values. You can establish another variable, which further depends on this one: K = 2.345 * N, or K = sin(30 * N). This way K has a discrete value, but is not necessarily integer. Or, one can use N directly as a design parameter.

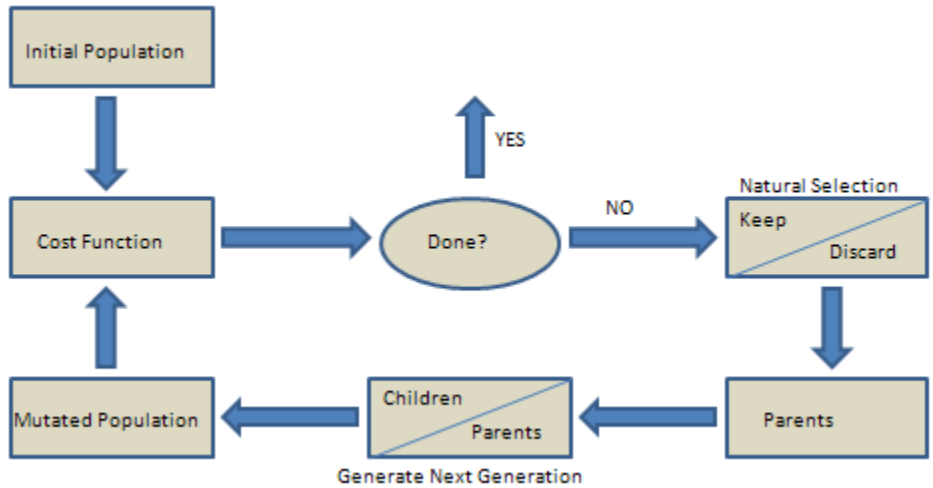
Related Topics

Optimization Setup for SMINLP Optimizer

Genetic Algorithm (Random Search)

Genetic Algorithm (Random Search) optimizers are part of a class of optimization techniques called stochastic optimizers. They do not use the information from the experiment or the cost function to determine where to further explore the design space. Instead, they use a type of random selection and apply it in a structured manner. The random selection of evaluations to proceed to the next generation has the advantage of allowing the optimizer to jump out of a local minima at the expense of many random solutions which do not provide improvement toward the optimization goal. As a result, the GA optimizer will run many more iterations and may be prohibitively slow.

The Genetic Algorithm search is an iterative process that goes through a number of generations (see picture below). In each generation some new individuals (Children / Number of Individuals) are created and the grown population participates in a selection (natural-selection) process that in turn reduces the size of the population to a desired level (Next Generation / Number of Individuals).



When a smaller set of individuals must be created from a bigger set, the GA selects individuals from the original set. During this process, better fit (in relation to the cost function) individuals are preferred. In the elitist selection, simply the best so many individuals are selected, but if you turn on the roulette selection, then the selection process gets relaxed. An iterative process starts selecting the individuals and fills up the resulting set, but instead of selecting the best so many, we use a roulette wheel that has for each selection-candidate divisions made proportional to the fitness level (relative to the cost function) of the candidate. This means that the fitter the individual is, the larger the probability of his survival will be.

Related Topics

Optimization Setup for Genetic Algorithm Optimizer

Optimization Variables in Design Space

Cost Function

Advanced Genetic Algorithm Optimizer Options

MATLAB optimizer

The MATLAB optimizer option lets you pass a script to MATLAB to perform the optimization. When the optimization is analyzed, MATLAB is launched and a script is passed in to MATLAB to perform the optimization. During the optimization, MATLAB will call back into our application to perform the solve and compute the cost. The cost will be reported back to MATLAB, and MATLAB's optimization will determine the next step in the optimization.

The optimization script is specified as part of the optimization setup. By modifying the optimization script, users can change the optimization parameters and optimization method as well as use the full power of MATLAB in their optimization.

Running the Optimization

The MATLAB optimization is launched just like any other optimization. The Message Window will display status messages when MATLAB is being launched, and status messages will be generated for each solve that is being performed.

In most cases, MATLAB will terminate when the optimization has been completed. Some reasons why MATLAB would not terminate are:

- The user has modified the MATLAB script to not terminate MATLAB after the optimization.
- A syntax error or some other has occurred.
- The user has added some other code which runs after the optimization has completed.

System Requirements

In order to use MATLAB to perform optimizations from your application:

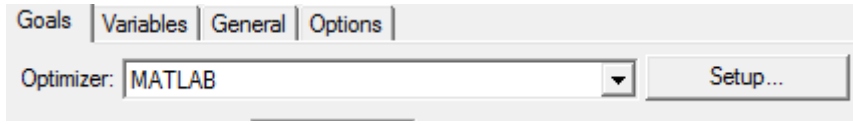
- A version of MATLAB must be installed on your system.
- The computing platform (i.e. 32/64 bit) of MATLAB MUST match the platform of the ANSYS application you are using it with.
- You must have the MATLAB Optimization Toolkit installed.

Specifying the MATLAB Location

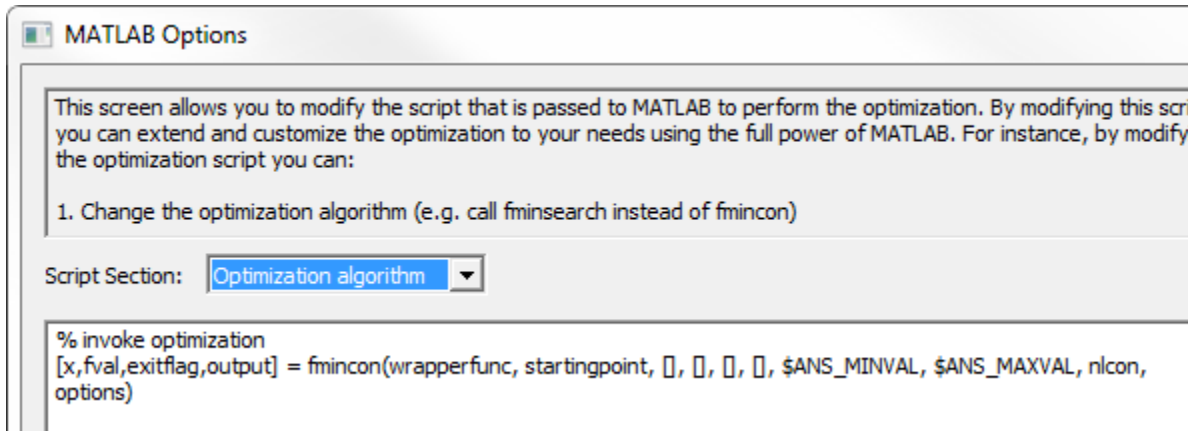
The Tools>General Options:Miscellaneous tab contains a setting for the MATLAB location. This setting must to point to the version of MATLAB to be used for performing the optimization. The platform (32/64 bit of the specified version of MATLAB must match the platform of this application). \

MATLAB Optimization Setup

MATLAB optimization starts by creating an optimization and selecting MATLAB from the optimizer dropdown list. If you select MATLAB as the optimizer, the **Setup Optimization** dialog displays a **Setup...** button.



Select **Setup...** to open the **MATLAB Options** dialog.



Thee upper text panel is informative. The Script section drop down lets you select a lower panel display for Optimization algorithm, Options, or the Full script template.

This screen allows you to modify the script that is passed to MATLAB to perform the optimization. The complete script contains all the instructions necessary for MATLAB to connect to our application and perform the optimization, and a lot of that code is unimportant to users. We have addressed this issue by displaying a dropdown to let you view only the portion of code they are interested in without having to view the full script. The choices are:

- **Optimization algorithm**: displays only the line of code invoking the actual optimization function. By changing this line, the user can use a different MATLAB function for optimization. By default we use fmincon() which is a derivative based constrained optimization. By modifying this line, the user could replace the fmincon() call with fminsearch() to use an unconstrained pattern searching optimizer or another optimization function. See the MATLAB documentation for details about available optimization functions.
- **Options**: Each optimization function contains a multitude of options and parameters which are

set in the MATLAB script prior to actually calling the optimization function. By modifying these options, the optimization can be customized as desired. For instance, options can be set for fmincon() to specify the algorithm that it uses internally. See the MATLAB documentation for details about options available for each optimization function.

- **Full script template**: This choice displays the full optimization script that is passed to MATLAB.

The initial Script Section display for the Optimization algorithm shows the following:

```
% invoke optimization
[x,fval,exitflag,output] = fmincon(wrapperfunc, startingpoint,
[], [], [], [], $ANS_MINVAL, $ANS_MAXVAL, nlcon, options)
```

The initial Script Section Options display shows the following:

```
% customers can add their own options below
options = optimset(options, 'display', 'iter')
options = optimset(options, 'Algorithm', 'interior-point')
% options = optimset(options, 'PlotFcns', @optimplotfval)
```

You can modify the script to extend and customize the optimization to your needs. You must ensure that the script follows MATLAB syntax. For instance, by modifying the optimization script you can:

- Change the optimization algorithm (e.g. call fminsearch instead of fmincon)
- Change the parameters/options of the optimization algorithm (see the MATLAB documentation for details).
- Specify a plot function to provide graphical output during optimization.
- Specify a user defined output function to be called at completion or per iteration.

Symbols:

When modifying the MATLAB code, users can use symbols to represent values from the optimization setup. The symbols and their definitions are listed below.

$ANS_VARIABLE_LIST:	list of variables we are optimizing
$ANS_STARTING_POINT:	vector of starting values of variables used in the optimization
$ANS_MAXITERATIONS:	maximum number of iterations specified in optimization setup
$ANS_MINVAL:	vector of minimum values from optimization setup
$ANS_MAXVAL:	vector of maximum values from optimization setup
$ANS_MINSTEP:	vector of minimum step sizes from optimization setup
$ANS_MAXSTEP:	vector of maximum step sizes from optimization setup

Note While modifying the script, please ensure that the script follows MATLAB syntax.

MATLAB Optimization Script Template

The script template shown in the Script Section is as follows:

```
% make sure platform matches
if strcmp(computer, '$ANS_EXPECTED_PLATFORM') ~= 1
  h = msgbox('32/64 platform does not match calling application,
exiting')
  uiwait(h)
  exit
end

% add installation dir to search path so .mex file can be found
originalpath = addpath('$ANS_EXEDIR')

% connect back to opticomengine
callbackinterface = optimex('connect', '$ANS_CONNECTIONSTRING')

% set up optimization
% variables are: $ANS_VARIABLELIST
startingpoint = $ANS_STARTINGPOINT
options = optimset('MaxIter', $ANS_MAXITERATIONS)
iterationCallbackWrapper = @(x, optimValues, state) optimex('noti-
fyiterationcomplete', callbackinterface, x, optimValues.fval,
state)
options = optimset(options, 'OutputFcn', iterationCallbackWrapper)

% halt execution so debugger can be attached
% h = msgbox('attach debugger if desired')
% uiwait(h)

% attributes that user can pass to optimization algorithm
% variables are: $ANS_VARIABLELIST

% this is the objective function which returns cost
```

```
wrapperfunc = @(x)optimex('eval', callbackinterface, x)

% this is our non linear constraint function, returns no con-
straints
returnempty = @(x)[];
nlcon = @(x) deal(returnempty(x), returnempty(x));

% DO NOT EDIT THIS LINE - START OPTIONS SECTION

% customers can add their own options below
options = optimset(options, 'display', 'iter')
options = optimset(options, 'Algorithm', 'interior-point')
% options = optimset(options, 'PlotFcns', @optimplotfval)

% DO NOT EDIT THIS LINE - END OPTIONS SECTION
% DO NOT EDIT THIS LINE - START OPTIMIZATION ALGO SECTION
% invoke optimization
[x,fval,exitflag,output] = fmincon(wrapperfunc, startingpoint, [],
[], [], [], $ANS_MINVAL, $ANS_MAXVAL, nlcon,
options)

% DO NOT EDIT THIS LINE - END OPTIMIZATION ALGO SECTION

% write exit message to Ansoft message window (warn-
ing=0,error=1,info=2)
optimex('postansoftmessage', callbackinterface, 2, output.message)

% notify opticomengine that optimization is finished
optimex('optimizationfinished', callbackinterface, exitflag)

% restore original path
path = originalpath

% note: comment below line if you want MATLAB to remain
% running after optimization
exit
```

Related Topics

Tools>General Options:Miscellaneous tab

Optimization Variables and the Design Space

Once the optimization variables are specified, the optimizer handles each of them as an *n*-dimensional vector *x*. Any point in the design space corresponds to a particular *x*-vector and to a design instance. Each design instance may be evaluated via FEA and assigned a cost value; therefore, the cost function is defined over the design space (cost(*x*): $R^n \rightarrow R$, where *n* is the number of optimization variables.

In practice, a solution of the minimization problem is sought only on a bounded subset of the Rn space. This subset is called the feasible domain and is defined via linear constraints.

Setting Up an Optimization Analysis

Optimization allows you to vary predefined variables in the nominal design to search for the solution that best satisfies a set of user defined goals or cost functions. Optimetrics modifies the variable values until the minimum is reached with acceptable accuracy.

Note

- You can define more than one optimization analysis setup per design.
- Once you have created an optimization analysis setup, you can copy and paste it, and then make changes to the copy, rather than redoing the whole process for minor changes.

To provide a broad range of capability, Optimetrics incorporates the following types of numerical optimizers:

- Sequential Nonlinear Programming (Gradient)
- Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete)
- Quasi Newton (Gradient)
- Pattern Search (Search-based)
- Genetic Algorithm (Random search)
- MATLAB

Click on the links above to view the setup procedure for each optimizer. Options for the analysis are listed in the table.

The following ***optional*** optimization solution setup options can also be used:

- Modify the starting variable value.
- Edit the Calc. Range text field or use the Edit Calculation Range dialog.
- Modify the minimum and maximum values of variables that will be optimized.
- Exclude variables from optimization.
- Modify the values of fixed variables that are not being optimized.
- Set the minimum and maximum step size between solved design variations For the Quasi Newton (Gradient) and Pattern Search (Search based optimizers), **Variables** tab).
- Set the minimum and maximum focus size. (For the SNLP Gradient and SMINLP Gradient and Discrete optimizers, **Variables** tab).
- Set Linear constraints.
- Request that Optimetrics solve a parametric sweep before an optimization analysis.
- Request that Optimetrics solve a parametric sweep during an optimization analysis.
- Automatically update optimized variables to the optimal values during an optimization or after an optimization analysis is completed.
- Change the norm used for the cost function calculation (Advanced Option)
- Open the HPC and Analysis Options dialog.

Note Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

Related Topics

Optimization Overview

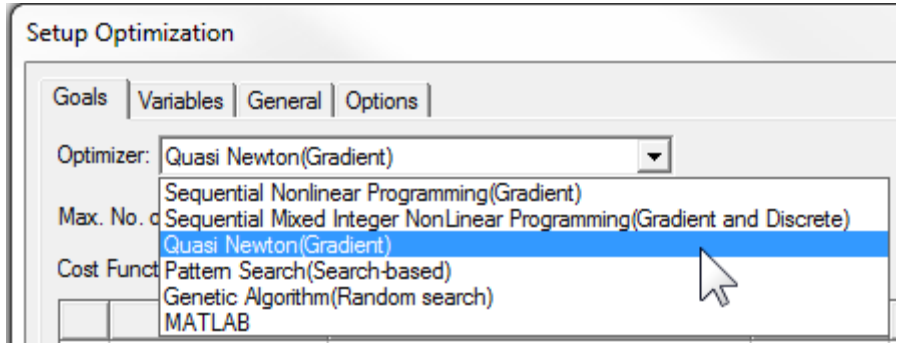
Choosing an Optimizer

Optimization Setup for the Quasi Newton Optimizer

Following is the procedure for setting up an optimization analysis using the Quasi Newton (Gradient) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. Click **HFSS** >**Optimetrics Analysis**>**Add Optimization** .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Quasi Newton (Gradient)** from the **Optimizer** pull-down list. Selecting Quasi Newton (Gradient) enables the **Acceptable Cost** and **Noise** fields.



4. Type the maximum number of iterations you want Optimetrics to perform during the optimization analysis in the **Max. No. of Iterations** text box.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. Type the value of the cost function at which the optimization process should stop in the **Acceptable Cost** text box. Note that for Quasi Newton, if you specify 0 as the acceptable cost, the simulation stops after the first analysis.
7. Type the cost function noise in the **Noise** text box.
8. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

A norm is a function that assigns a positive value to the cost function.

For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

9. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Step Size** for the analysis.

   You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.

10. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** checkbox will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.

11. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

Optimization Setup for the Pattern Search Optimizer

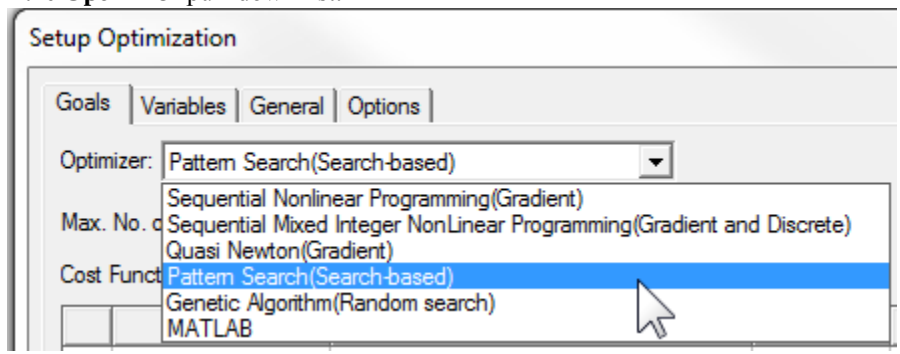
Following is the procedure for setting up an optimization analysis using the Pattern Search (Search-based) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. Click **HFSS** >**Optimetrics Analysis**>**Add Optimization**  .

   The **Setup Optimization** dialog box appears.

3. Under the **Goal**s tab, select the optimizer by selecting **Pattern Search (Search-based)** from

the **Optimizer** pull-down list.



Selecting Pattern Search enables the **Acceptable Cost** and **Noise** fields.

4. Type the maximum number of iterations you want Optimetrics to perform during the optimization analysis in the **Max. No. of Iterations** text box.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. Type the value of the cost function at which the optimization process should stop in the **Acceptable Cost** text box.
7. Type the cost function noise in the **Noise** text box.
8. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

     A norm is a function that assigns a positive value to the cost function.

     For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

     The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.
9. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Step Size** for the analysis.

   You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.
10. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** checkbox will cause Optimetrics to modify the variable values in the nominal design to match the final values from the

optimization analysis.

11. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

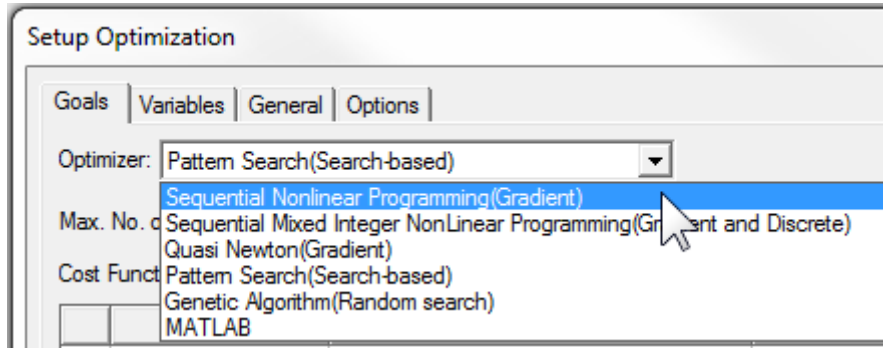
You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

Optimization Setup for the SNLP Optimizer

Following is the procedure for setting up an optimization analysis using the Sequential Nonlinear Programming (SNLP) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. Click **HFSS** >**Optimetrics Analysis**>**Add Optimization** .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Sequential Nonlinear Programming (Gradient)** from the **Optimizer** pull-down list.



4. Type the maximum number of iterations you want Optimetrics to perform during the optimization analysis in the **Max. No. of Iterations** text box.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

     A norm is a function that assigns a positive value to the cost function.

For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors, which means that it is always less than zero. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

7. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Focus** for the analysis.

   You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.

8. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

   Enabling the **Update design parameters' value after optimization** checkbox will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.

9. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

Optimization Setup for the SMINLP Optimizer

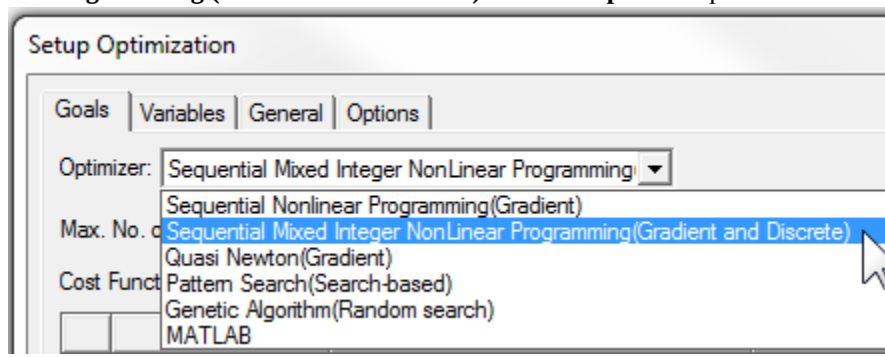
Following is the procedure for setting up an optimization analysis using the Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. Click **HFSS** >**Optimetrics Analysis**> **Add Optimization**  .

   The **Setup Optimization** dialog box appears.

3. Under the **Goals** tab, select the optimizer by selecting **Sequential Mixed Integer Nonlinear**

Programming (Gradient and Discrete) from the **Optimizer** pull-down list.



4. Type the maximum number of iterations you want Optimetrics to perform during the optimization analysis in the **Max. No. of Iterations** text box.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

     A norm is a function that assigns a positive value to the cost function.

     For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

     The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.
7. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Focus** for the analysis.

   Check the Integer box for integer variables.

   You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.
8. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

   Enabling the **Update design parameters' value after optimization** checkbox will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
9. Under the **Options** tab, if you want to save the field solution data for every solved design vari-

ations in the optimization analysis, select **Save Fields And Mesh**.

Note Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

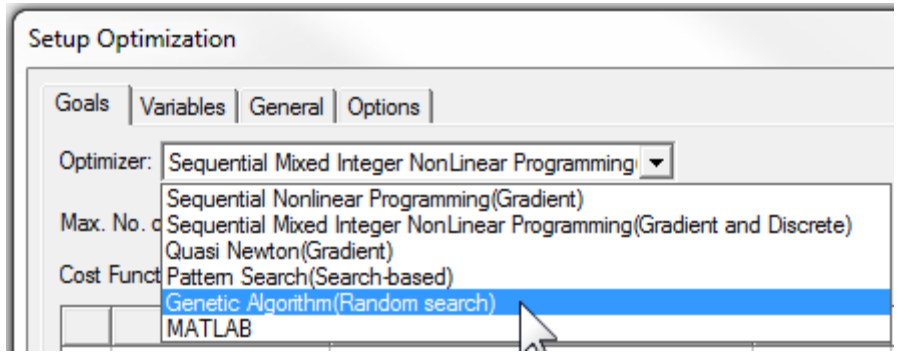
You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

Optimization Setup for the Genetic Algorithm Optimizer

Following is the procedure for setting up an optimization analysis using the Genetic Algorithm (Random search) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. Click **HFSS** >**Optimetrics Analysis**>**Add Optimization** .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Genetic Algorithm(Random search)** from the **Optimizer** pull-down list.



4. Click the **Setup...** button to modify the Advanced Genetic Algorithm Optimizer Options.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

     A norm is a function that assigns a positive value to the cost function.

     For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost func-

tion uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

7. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Focus** for the analysis.

   You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.

8. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

   Enabling the **Update design parameters' value after optimization** checkbox will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.

9. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

Setting the Maximum Iterations for an Optimization Analysis

The **Max. No. of Iterations** value is the maximum number of design variations that you want Optimetrics to solve during an optimization when using the SNLP, SMINLP, Quasi Newton, or Pattern Search Optimizer. This value is a stopping criterion; if the maximum number of iterations has been completed, the optimization analysis stops. If the maximum number of iterations has not been completed, the optimization continues by performing another iteration, that is, by solving another design variation.

If the maximum number of iterations has not been reached, the optimizer performs iterations until the acceptable cost function is reached or until the optimizer cannot proceed as a result of other optimization setup constraints, such as when it searches for a variable value with a step size smaller than the minimum step size.

Note The **Genetic Algorithm** optimizer does not use the **Max. No. of Iterations** criteria.

To set the maximum number of iterations for an optimization analysis:

- Under the **Goals** tab of the **Setup Optimization** dialog box, type a value in the **Max. No. of Iterations** text box.

Related Topics

Adding a Cost Function

Cost Function

Optimetrics manipulates the model's design variable values to find the minimum location of the cost function; therefore, you should define the cost function so that a minimum location is also the optimum location. For example, if you vary a design to find the maximum transmission from *Wave Port 1* to *Wave Port 2* (S_{21}=>1), define the cost function to be -mag(S(WavePort2,WavePort1)).

When using the Quasi Newton optimizer, which is appropriate for designs that are not sensitive to noise, the best cost function is a smooth, second-order function that can be approximated well by quadratics in the vicinity of the minimum; the slope of the cost function should decrease as Optimetrics approaches the optimum value. The preferred cost function takes values between 0 and 1. In practice, most functions that are smooth around the minimum are acceptable as cost functions. Most importantly, the cost function should not have a sharp dip or pole at the minimum. A well designed cost function can significantly reduce the optimization process time.

The cost function is defined in the **Setup Optimization** dialog box when you set up an optimization analysis. If you know the exact syntax of the solution quantity on which you want to base the cost function, you can type it directly in the **Calculation** text box. You can also use **Setup Calculations** to add a solution quantity via the **Add/Edit Calculation** dialog box, or to create an output variable that represents the solution quantity in the Output Variables dialog box.

Related Topics

Adding a Cost Function

Acceptable Cost

Cost Function Noise

Linear Constraints

Goal Weight

Step Size

Explanation of L1, L1, Norm Costs in Optimization

Example of a More Complex Cost Function

Acceptable Cost

The acceptable cost is the value of the cost function at which the optimization process should stop; otherwise known as the *stopping criterion*.The cost function value must be equal to or below the acceptable cost value for the optimization analysis to stop. The acceptable cost may be a negative value.

Related Topics

Cost Function

Adding a Cost Function

Cost Function Noise

The numerical calculation of the electromagnetic field introduces various sources of noise to the cost function, particularly because of changes in the finite element mesh. You must provide the optimizer with an estimate of the noise. The noise indicates whether a change during the solution process is significant enough to support achievement of the cost function.

For example, if the cost function, c, is

$$c = 10000 \cdot |S_{11}|^2$$

where $|S_{11}|$ is the magnitude of the reflection coefficient, at the minimum, $|S_{11}|$ is expected to be very small, $|S_{11}| \approx 0$.

From the solution setup, the error in $|S_{11}|$ is expected to be $E_{S11} \approx 0.01$. The perturbed cost function is therefore

$$c_{perturbed} = 10000 \cdot (|S_{11}|_{min} + E_{S11})^2$$

Near the minimum, the error in the cost function E_c is given by

$$E_c = c_{perturbed} - c_{min} = 10000 \cdot (0.0 + 0.01)^2 - (10000 \cdot 0.0) = 1.0$$

Therefore, the cost function noise would be 1.0.

Related Topics

Cost Function

Adding a Cost Function

A cost function can include one or more goals for an optimization analysis. Optimetrics manipulates the model's design variable values to fulfill the cost function.The optimization will stop when the solution quantity meets the acceptable cost criterion.

Following is the general procedure for adding a cost function with a single goal:

1. Under the **Goals** tab of the **Setup Optimization** dialog box, click **Setup Calculations...**

   The Add/Edit Calculation dialog box is displayed.
2. In the **Add/Edit Calculation** dialog box, follow these general steps to set up a cost function.
   a. Set the **Context** for the calculation.
   b. Choose the **Category** of available data type depending upon the Solution type of the design being optimized.
   c. Select the **Quantity** to add to the **Calculated Expression** field. Available quantities depend upon the **Category** selection.

d. You may optionally make a selection from the function list to apply to the calculated expression.

e. When the **Calculation Expression** has the desired equation, click **Add Calculation** to add the expression to the cost function table.

f. Repeat to add additional calculations to the cost function or click **Done** to exit the **Add/Edit Calculation** dialog box and return to **Setup Optimization**.

3. To modify the **Solution** on which the calculation is based, click in the **Solution** column and select the solution from which the cost function is to be extracted from the pull-down list.
4. To edit the calculation on which to base the cost function goal, select **Edit** from the pull-down list.
5. In the **Condition** text box, click one of the following conditions from the pull-down list:

<=	Less than or equal to
=	Equal to
>=	Greater than or equal to
Minimize	Reduce the cost function to a minimum value
Maximize	Identify a maximized condition

6. In the **Goal** text box, type the value of the solution quantity that you want to be achieved during the optimization analysis. If the solution quantity is a complex calculation, the goal value must be complex; two goal values must be specified. The **Minimize** and **Maximize** options do not require you to specify a **Goal** value.

   When Minimize is used as an optimization condition, the value of calculation is used as the cost (there is no target value to compare to). For maximize, the negative of calculation value is used as cost.
7. Optionally, if you have multiple goals and want to assign higher or lower priority to a goal, type a different value for the goal's weight in the **Weight** text box. The goal with the greater weight is given more importance. If the goal is a complex value, the weight value must be complex; two weight values must be specified. The weight value cannot be variable dependent.

Note Click the **Edit Goal/Weight** button to open the **Edit Goal Value/Weight** dialog box where you can modify weights for all goals simultaneously; as well as, set the **Goal Values** to expressions.

8. Specify other options (such as acceptable cost, noise, and number of passes), and then click **OK**.

The optimization stops when the solution quantity meets the acceptable cost criterion.

Related Topics

Setting a Goal Value

Cost Function

Acceptable Cost

Goal Weight

Example of a More Complex Cost Function

Adding/Editing a Cost Function Calculation

The **Add/Edit Calculation** dialog box allows you to define the mathematical equation for one or multiple cost functions. It represents the calculation to be performed on the optimization variables to compare to the goal values. To set up a calculation for a cost function:

1. In the **Context** section of the dialog:
   - Select the **Report Type** with a pull-down selection list containing the available types for this design.
   - Select the **Solution** from the drop down selection list. This lists the available setups and sweeps. As a minimum, the **LastAdaptive** solution is available.
   - Select the Geometry from the drop down selection list or select none (the default). This modifies the list of quantities available to the ones that apply to the specific geometry.
2. The **Output Variables** button opens the Output Variables dialog box allowing you to create special output variables to be used in the cost function.
3. The **Calculated Expression** field in the **Trace** tab is used to enter the equation to be used for the cost function. To enter an expression, you may type it directly into the field or use the **Category**, **Quantity**, and **Function** lists as follows:
   - Select the **Category**, these depend on the Solution type and the design. This lets you specify the category of information to be used in the cost function.
   - Select a **Quantity** from the list. Available quantities depend upon the Solution type, as well as the Geometry and Category selection. Selecting a Quantity automatically enters it into the Calculated Expression field.
   - Select a **Function** to apply to the value in the calculated expression.
   - For swept variables, the Range**Function** button opens the **Set Range Function** dialog to apply functions to the expression that apply over the sweep range.
4. The **Calculation Range** tab applies to swept variables and allows you to specify the range of the sweep over which to apply the calculation.
5. When the desired **Calculated Expression** has been obtained, click the **Add Calculation** button to add the entry to the cost function table. You may add multiple entries to the table simply by changing the **Calculated Expression** and using the **Add Calculation** button.
6. To update or edit a selected cost function, enter the desired Calculated Expression and click the **Update Calculation** button.
7. Click **Done** to return to the **Setup Calculations** dialog box.

Related Topics

Example of a More Complex Cost Function

Specifying a Solution Quantity for a Cost Function Goal

When setting up a cost function, you must identify the solution quantity on which to base each goal. Solution quantities are specified by mathematical expressions that are composed of basic quantities, such as matrix parameters, and output variables.

1. Add a row (a goal) to the cost function table:
   a. Under the **Goals** tab of the **Setup Optimization** dialog box, click **Add**.

      A new row is added to the **Cost Function** table.

   b. In the **Solution** column, click the solution from which the cost function is to be extracted.
2. In the **Solution** text box, click the solution from which the solution quantity is to be extracted.
3. In the **Calculation** text box, specify the solution quantity in one of the following ways:
   - If you know the syntax of the mathematical expression or the output variable's name, type it in the **Calculation** text box.
   - If you want to create an output variable that represents the solution quantity, do the following:
     a. Click **Edit Calculation**.

        The **Output Variables** dialog box appears.

     b. Add the expression you want to evaluate, click **Done**.
     c. Click **Done** to close the **Output Variables** dialog box.

        In the **Setup Optimization** dialog box, the most recently created output variable appears in the **Calculation** text box.

     d. To specify a different defined output variable, click the **Calculation** text box. It becomes a pull-down list that displays all of the defined output variables. Click an output variable from the pull-down list.

Setting the Calculation Range of a Cost Function Goal

The calculation range is the range within which you want a cost function goal to be calculated. It can be a single value or a range of values, depending on the solution or solution quantity selected for the goal.

1. Under the **Goals** tab in the **Setup Optimization** dialog box, click **Edit Cal. Range**.
2. In the **Variable** pull-down list, click a variable.

   If you chose to solve a parametric setup during the optimization analysis, the variables swept in that parametric setup are available in the **Variable** pull-down list. If you sweep a variable in the parametric setup that is also being optimized, that variable is excluded from the optimization.

   Other examples of available variables include frequency, if the solution quantity is an S-parameter quantity, and phi or theta, if the solution quantity is a radiated field quantity.
3. After you select a variable from the **Variable** pull-down list, you can select a range of values for the calculation range as follows:
   a. Select **Range**.

b. In the **Start** text box, type the starting value of the range.

c. In the **Stop** text box, type the final value of the range.

4. To select a single value for the calculation range:

   a. Select **Single Value**.

   b. In the **Value** text box, type the value of the variable at which the cost function goal is to be extracted.

5. Click **Update**, and then click **OK**.

Setting a Goal Value

A goal is the value you want a solution quantity to reach during an optimization analysis. It can be a real value or a complex value. If the solution quantity is a complex calculation, the goal value must be complex. You can type the goal value in the **Goal** text box. Alternatively, you can use the **Edit Goal/Value Weight** dialog box to specify the goal value as a single value, a mathematical expression, or a value dependent on a variable such as frequency.

Related Topics

Specify a single goal value.

Specify an expression as the goal value.

Specify a variable-dependent goal value.

Example of a More Complex Cost Function

Specifying a Single Goal Value

1. Under the **Goals** tab in the **Setup Optimization** dialog box, click **Edit Goal/Weight**.

   The **Edit Goal/Weight** dialog box appears.

2. Under the **Goal Value** tab, click **Simple Numeric Value** from the **Type** list.

3. If the goal value is complex, click **real/imag** in the pull-down list to the right if you want to specify the real and imaginary parts of the goal value.

   Alternatively, click **mag/ang** if you want to specify the magnitude and angle of the goal value.

4. Type the goal value in the **Goal Value** table.

   If the goal value is complex, type both parts of the goal value in the text box below the **Goal Value** heading. For example, type **1, 1** to specify the real part of the goal value as 1 and the imaginary part as 1.

   If the goal value is real, type a real goal value in the text box below the **Goal Value** heading.

5. Click **OK**.

   The goal value you specified appears in the **Goal** text box.

Specifying an Expression as a Goal Value

1. Under the **Goals** tab in the **Setup Optimization** dialog box, click **Edit Goal/Weight**.

The **Edit Goal/Weight** dialog box appears.

2. Under the **Goal Value** tab, click **Expression** from the **Type** list.
3. If you know the syntax of the mathematical expression or the existing output variable's name, type it in the text box below the **Goal Value** heading.

   Alternatively, if you want to create an output variable that represents the goal value, do the following:

   a. Click **Edit Expression**.

      The **Output Variables** dialog box appears.

   b. Add the expression you want to be the goal value, and then click **Done**.

      HFSSenters the most recently created output variable in the text box below the **Goal Value** heading.
4. Click **OK**.

   The goal value you specified appears in the **Goal** text box.

Specifying a Variable-Dependent Goal Value

1. Under the **Goals** tab in the **Setup Optimization** dialog box, click **Edit Goal/Weight**.

   The **Edit Goal/Weight** dialog box appears.
2. Under the **Goal Value** tab, click **Variable Dependent** from the **Type** list.
3. Click a variable from the pull-down list to the left of the table.
4. Type the value of that variable in the first column of the table.

Warning Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

5. Type a corresponding goal value for that variable value in the text box below the **Goal Value** heading.
6. Click **Add** to add another row to the reference curve.
7. Repeat steps 4, 5, and 6 until you have specified the reference curve.
8. Click **OK**.

   The goal value is listed as being variable dependent in the **Goal** text box.

Goal Weight

If an optimization setup has a cost function made up of multiple goals, you can assign a different weight to each goal. The goal with the greater weight is given more importance during the cost calculation.

The error function value is a weighted sum of the sub-goal errors. Each sub-goal, at each frequency at which it is evaluated, gives rise to a (positive) error value that represents the discrepancy between the simulated response and the goal value limit. If the response satisfies the goal value

limit, then the error value is 0. Otherwise, the error value depends on the differences between the simulated response and the respective goal limit. The error function may be defined as follows:

$$\sum_{j}^{G} \frac{W_j}{N_j} \sum_{i}^{N_j} e_i$$

where

- G is the number of sub-goals.
- W_j is the weight factor associated with the j[th] sub-goal.
- N_j is the number of frequencies for the j[th] sub-goal.
- e_i is the error contribution from the j[th] sub-goal at the i[th] frequency.

The value of e_i is determined by the band characteristics, target value, and the simulated response value. The choices for band characteristics are <=, =, and >=.

Band Characteristics (Condition)	**e_i evaluation where s_i is the simulated response and g_i is the desired limit.**
<=	$e_i = \begin{cases} 0 & s_i \le g_i \\ s_i - g_i & s_i > g_i \end{cases}$
=	$e_i = \lvert s_i - g_i \rvert$
>=	$e_i = \begin{cases} 0 & s_i \ge g_i \\ g_i - s_i & s_i < g_i \end{cases}$

If the total error value is within the acceptable cost, the optimization stops.

Related Topics

Adding a Cost Function

Cost Function

Example of a More Complex Cost Function

Modifying the Starting Variable Value for Optimization

A variable's starting value is the first value to be solved during the optimization analysis. Optimetrics automatically sets the starting value of a variable to be the current value set for the nominal design. You can modify this value for each optimization setup.

Note If you choose to solve a parametric setup before an optimization analysis, a variable's starting value is ignored if a more appropriate starting value is calculated for it during the parametric analysis.

1. In the **Setup Optimization** dialog box, click the **Variables** tab.

   All of the variables that were selected for the optimization analysis are listed.

2. Type a new value in the **Starting Value** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is used for this optimization analysis, and the current value set for the nominal model is ignored.

   - Alternatively, you can select the **Override** option first, and then type a new variable value in the **Starting Value** text box.

3. Optionally, click a new unit system in one of the **Units** text boxes.

Note To revert to the default starting value, clear the **Override** checkbox.

Related Topics

Setting the Min. and Max. Variable Values for Optimization

Step Size

Setting the Min and Max Focus

Modifying the Starting Variable Value for Sensitivity Analysis

Modifying the Starting Variable Value for Statistical Analysis

Setting the Min. and Max. Variable Values for Optimization

For every optimization setup, Optimetrics automatically sets the minimum and maximum values it will consider for a variable being optimized. Optimetrics sets a variable's minimum value equal to approximately 50% of its starting value. (The starting value is the variable's current value set for the nominal design.) Optimetrics sets the variable's maximum value equal to approximately 150% of the starting value. During the optimization analysis, variable values that lie outside of this range are not considered.

Warning Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any Optimetrics analysis.

Related Topics

Text Entry for Calc. Range or Edit Calculation Range Dialog

Override the default min and max variable values for a single optimization setup.

Change the default min and max variable values for every optimization setup.

Text Entry for Calc. Range or Edit Calculation Range Dialog

In the **Setup Optimization** dialog, you can enter the Calc. Range Sweep Min/Max by directly editing the Calc. Range field or by accessing an **Edit Calculation Range** dialog.

The edit field accepts the following forms of text:

- sweep that allows your to select different discrete values:

  discrete values, for example, F(10GHz, 11GHz)

  min/max range, for example, F([10GHz, 11GHz])
- editable sweep, which allows you to customize values (that is a sweep that has an enabled "edited" radio button in sweep selection dialog):

  The min/max is used on top of selected values. For example, if you use the sweep dialog and choose "0 deg, 60 deg, 180 deg, 240 deg", then [60deg, 240deg] will select values "60 deg, 180 deg, 240 deg".
- sweep that uses a full range:

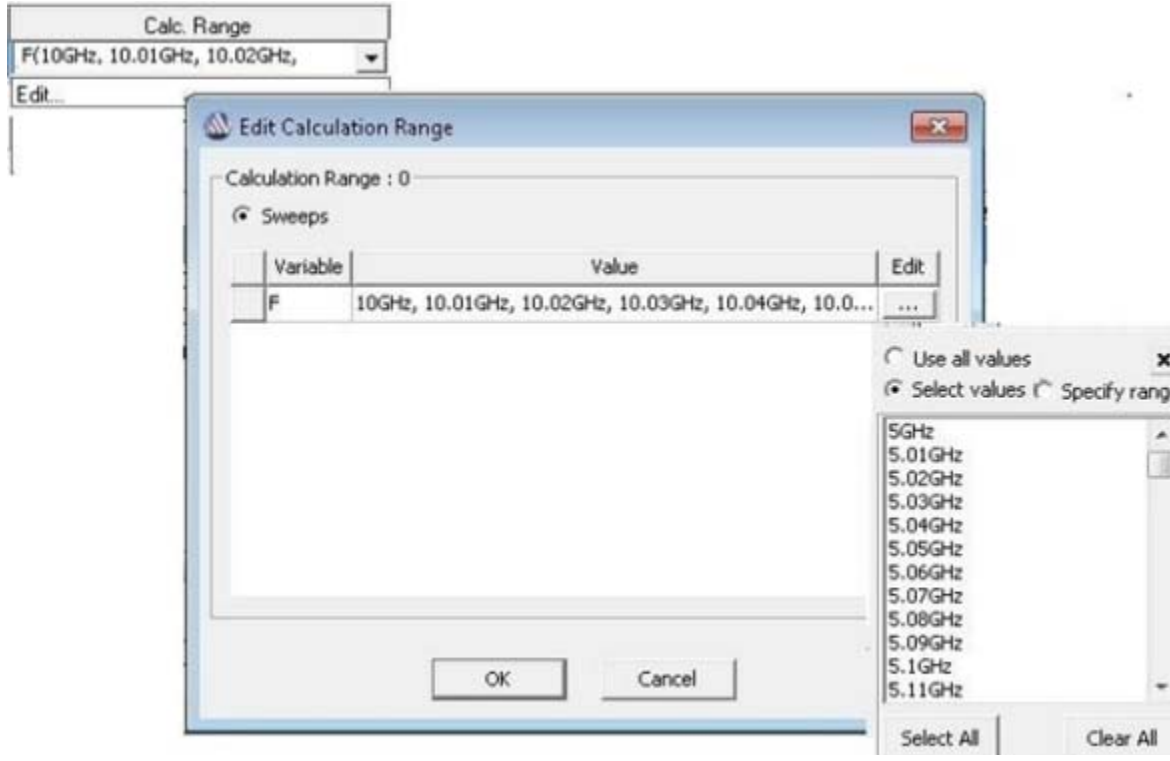
  all values, for example, Time(All)

  min/max range, for example, Time([1ms, 2ms]) HFSS/MAXWELL/SIMPLORER
- You solve 1 to 20 GHz step .1 and specify F[10.381GHz, 11.381GHz], it is equivalent to selecting values between 10.4GHz and 11.3GHz.
- You can specify multiple sweep values by separating those with comma ","

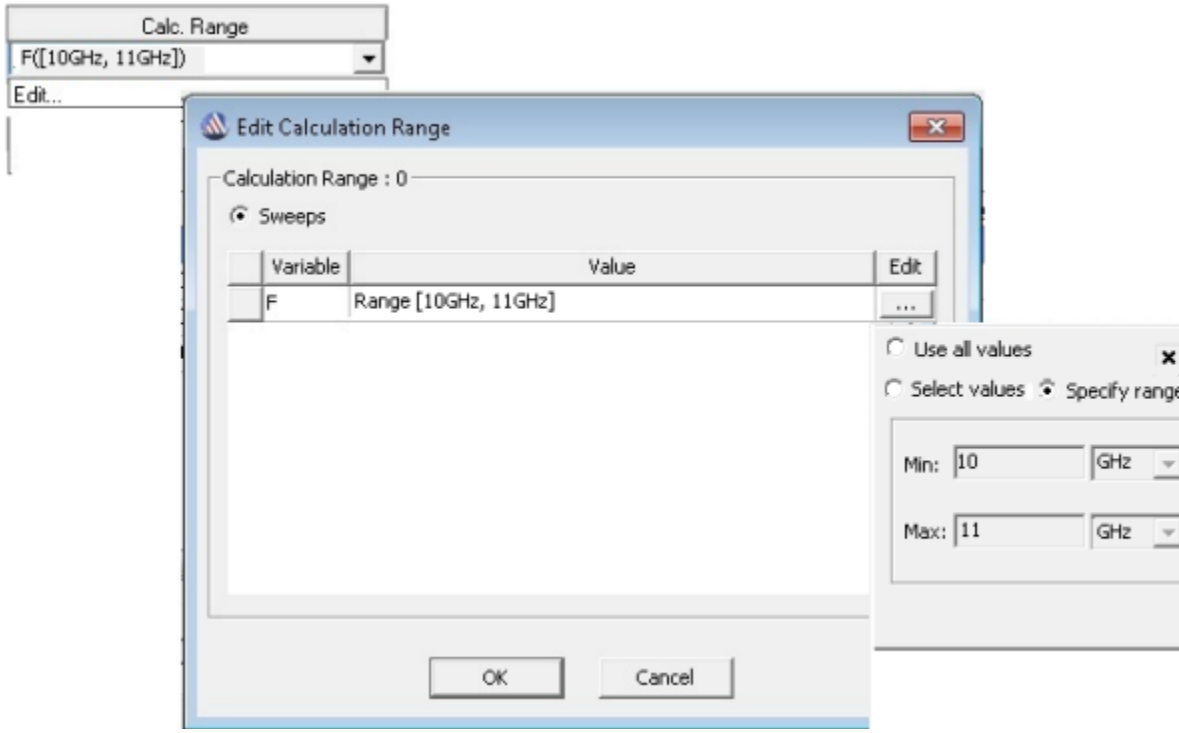
  For example, F(1GHz), cap(1pf, 1.2pf)

  For example, Distance(All), Freq([1ghz,2ghz]), Phase(0 deg)

If you click Edit.. on the menu, you see the **Edit Calculation Range** dialog. Click on the ellipsis [...] button to select radio buttons for Use all values, Select Values or Specify range.



This example shows that when you specify a range, how the range appears in the Calc. Range field..



You could also enter the range directly in the Cal. Range field.

Overriding the Min. and Max. Variable Values for a Single Optimization Setup

1. In the **Setup Optimization** dialog box, click the **Variables** tab.

   All of the variables that were selected for optimization analysis are listed.

2. Type a new value in the **Min** or **Max** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is used for this optimization analysis; the variable's current **Min** or **Max** value in the nominal design is ignored.

   - Alternatively, you can select the **Override** option first, and then type a new value in the **Min** or **Max** text box.

3. Optionally, click a new unit system in one of the **Units** text boxes.

To revert to the default minimum and maximum values, clear the **Override** option.

Changing the Min. and Max. Variable Values for Every Optimization Setup

1. Make sure that the variable's minimum and maximum values are not being overridden in any single optimization setup.
2. If the variable is a design variable, do the following: Click **HFSS>Design Properties**.

   If the variable is a project variable, do the following: Click **Project>Project Variables**.

   The **Properties** dialog box appears.
3. Select **Optimization**.
4. Type a new value in the **Min** or **Max** text box for the value you want to override, and then press **Enter**.
5. Click **OK**.

   When Optimetrics solves an optimization setup, it does not consider variable values that lie outside of this range.

Step Size

To make the search for the minimum cost value reasonable, the search algorithm is limited in two ways. First, you do not want the optimizer to continue the search if the step size becomes irrelevant or small. This limitation impacts the accuracy of the final optimum. Second, in some cases you do not want the optimizer to take large steps either. In case the cost function is suspected to possess large variations in a relatively small vicinity of the design space, large steps may result in too many trial steps, which do not improve the cost value. In these cases, it is safer to proceed with limited size steps and have more frequent improvements.

For these two limitations, the optimizer uses two independent distance measures. Both are based on user-defined quantities: the minimum and maximum step limits for individual optimization variables. Since the particular step is in a general direction, these measures are combined together in order to derive the limitation for that particular direction.

The step vector between the i^{th} and $(i+1)^{th}$ iterate is as follows:

$$s_i = x_{i+1} - x_i$$

The natural distance measure is,

$$\|s_i\| = \sqrt{s_i^T s_i}$$

which is the Euclidean norm.

A more general distance measure incorporates some "stretching" of the design space:,

$$\|s_i\|_D = \sqrt{s_i^T D^T D s_i}$$

where the matrix *D* incorporates the linear operation of the stretching of design space. The simplest case is when the *D* matrix is diagonal, meaning that the design space is stretched along the orthogonal direction of the base vectors.

The optimizer stops the search if,

$$\|s_i\|_{D_{min}} < 1$$

where D_{min} consists of diagonal elements

equal to the inverse of the **Min. Step** value assigned to the corresponding optimization variable. Similarly the optimizer truncates steps for which

$$\|s_i\|_{D_{max}} > 1$$

where D_{max} has diagonal elements equal to the inverse of **Max. Step** values of the corresponding optimization variables.

Related Topics

Setting the Min. and Max. Step Sizes

Cost Function

Adding a Cost Function

Setting the Min. and Max. Step Sizes

For the Quasi Newton and Pattern Search optimizers, the step size is the difference in a variable's value between one solved design variation and the next. The step size is determined when Optimetrics locates the next design variation that should be solved in an effort to meet the cost function.

1. In the **Setup Optimization** dialog box, click the **Variables** tab.
2. Optimetrics displays **Min Step** and **Max Step** columns, with default values for each variable to be optimized.
3. In the **Min Step** text box, type the minimum step size value. Optionally, modify the unit system in the **Units** text box.
4. In the **Max Step** text box, type the maximum step size value. Optionally, modify the unit system in the **Units** text box.

5. Click **OK**.

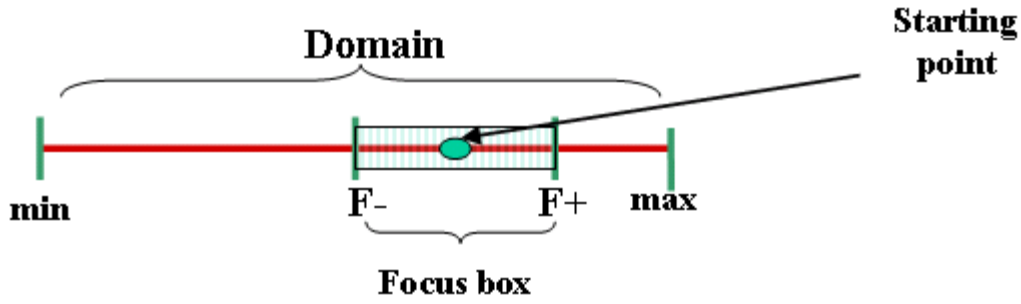
Hint A value of zero is recommended for the minimum step size.

Related Topics

Step Size

Setting the Min and Max Focus

For the SNLP, SMINLP and Genetic Algorithm optimizers, the min focus and max focus criteria allow you to specify a sub-range of parameter values where the optimizer should look when performing the optimization. This focus box is where you suspect the optimal solution will be, so it is a hint for the optimizer.



- The domain limits the search. The domain = physical limits.
- The focus box does not limit the search. Rather, the Focus box = an initial guess of optimum search domain. The starting point is the center of the focus box, but the search does extend beyond the box.
- This focus must be inside the domain limits. Consequently, it has to be equal or smaller size. An error message is generated if you specify a focus outside the domain.
- The focus box must be at least one hundredth of the domain size. Otherwise, an error message is sent.

Equalizing the influence of different optimization variables.

The optimizer seeks optimal values for the optimization variables. These variables are usually quantities with specified units. The change in one variable could be measured in [mm] and the change in other variable could be measured in [mA]. Instead of those units, the optimizer uses internal abstract units, so that a change in one variable changes the design behavior about as much as the same change in another variable, where changes are measured in the respective internal abstract units. When you define the focus box, the unit of the abstract internal unit is defined as the difference of the upper and lower focus limits. This way you can use the focus box to equalize the influence of different optimization variables on the design behavior.

To set the Min and Max Focus values:

1. In the **Setup Optimization** dialog box, click the **Variables** tab.
2. Optimetrics displays **Min. Focus** and **Max. Focus** columns, with default values for each variable to be optimized.

   If you do not have an initial guess based on your knowledge of the problem, make the focus box equal to the domain; that is, the physical limits. This tells SNLP to search the entire decision space.

   - In the **Min. Focus** text box, type the minimum value of the focus range. Optionally, modify the unit system in the **Units** text box.
   - In the **Max. Focus** text box, type the maximum value of the focus range. Optionally, modify the unit system in the **Units** text box.
   - Click **OK**.

Solving a Parametric Setup Before an Optimization

Solving a parametric setup before an optimization setup is useful for guiding Optimetrics during an optimization.

To solve a parametric setup before an optimization setup:

1. In the **Setup Optimization** dialog box, click the **General** tab.
2. In the **Parametric Analysis** pull-down list, click the parametric setup you want Optimetrics to solve before optimization.

Note The parametric setup must include sweep definitions for the variables you are optimizing.

3. Select **Solve the parametric sweep before optimization**.

If the parametric setup has not yet been solved, Optimetrics solves it. Optimetrics uses the cost value evaluated at each parametric design variation to determine the next step in the optimization analysis. This enables you to guide the direction in which the optimizer searches for the optimal design variation.

Related Topics

Solving a Parametric Setup During an Optimization

Solving a Parametric Setup During an Optimization

Solving a parametric setup during an optimization analysis is useful when you want Optimetrics to solve every design variation specified in the parametric setup at each optimization iteration. A cost function goal could then depend on the value of the variable swept in the parametric setup.

To solve a parametric setup during an optimization analysis:

1. In the **Setup Optimization** dialog box, click the **General** tab.
2. In the **Parametric Analysis** pull-down list, click the parametric setup you want Optimetrics to solve during an optimization.

3. Select **Solve the parametric sweep during optimization**.
4. Optionally, you can adjust the sweep values to be used during the optimization.
   a. Click on the **Goal** tab, click **Setup Calculations** to specify a calculation.
      The **Add/Edit Calculation** dialog box is displayed.
   b. Click the **Calculation Range** tab.
   c. Click the **Edit** button for the sweep to be modified.
   d. In the pop-up dialog box, select the sweep values to use.
   e. Close the pup-up dialog box. Click **Done** to close the **Add/Edit Calculation** dialog.

Automatically Updating a Variable's Value After Optimization

When Optimetrics finds an optimal variable value by solving an optimization setup, it can automatically update that variable's current value set for the nominal model to the optimal value.

1. In the **Setup Optimization** dialog box, click the **General** tab.
2. Select **Update design parameters' values after optimization**.
   When optimization is complete, the current variable value for each optimized variable is changed to the optimal value.

Changing the Cost Function Norm

You can select the norm to be used in the calculation of the cost goal.

1. In the **Setup Optimization** dialog box, click the **Goals** tab.
2. Select **Show Advanced Options**.
3. Select a norm from the pull-down in the **Cost Function Norm Type** field. The options are **L1**, **L2**, and **Maximum**. **L2** is the default.

Related Topics

Explanation of L1, L2 and Max Norms in Optimization

Cost Function

Explanation of L1, L2 and Max norms in Optimization

When you set multiple goals for an optimization, the question arises as to what is actually going to drive the optimizer which is not a multi-objective one. The cost function will have a lot to do with it. The following discussion explains how the cost function is put together when there are multiple goals.

The general goal setting structure in Optimetrics is a logical sentence with the format:

$$Calculation_{(i)}\, Condition_{(i)}\, Goal_{(i)}\, Weight_{(i)}$$

The cost function that the optimizer uses is built based on the norm setting as long as there are multiple goals and none of those use the "minimize" or "maximize" conditions. Thus, in this case the error associated with each individual goal (weighted) is combined in a way that is specific for each norm type chosen.

For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors:

$$Cost = \sum_{1}^{N} \left| w_i \varepsilon_i \right|$$

For **L2** norm the actual cost function uses the weighted sum of absolute values of the individual goal errors.

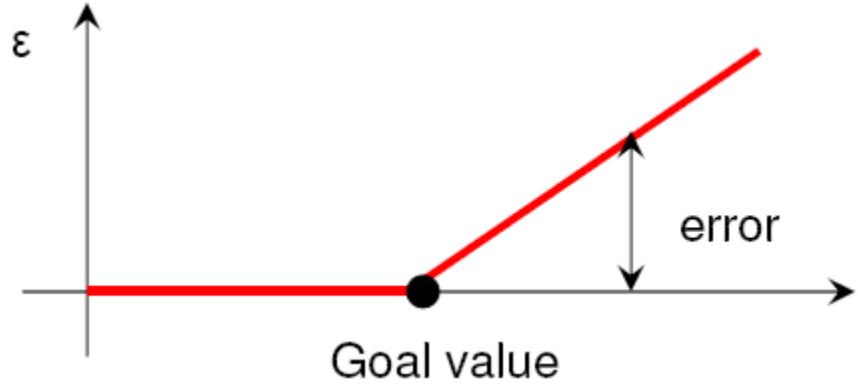
$$Cost = \sum_{1}^{N} w_i \varepsilon_i^2$$

For the **Maximum** norm the cost function uses the maximum among all the weighted goal errors, which means that cost is always less than zero:

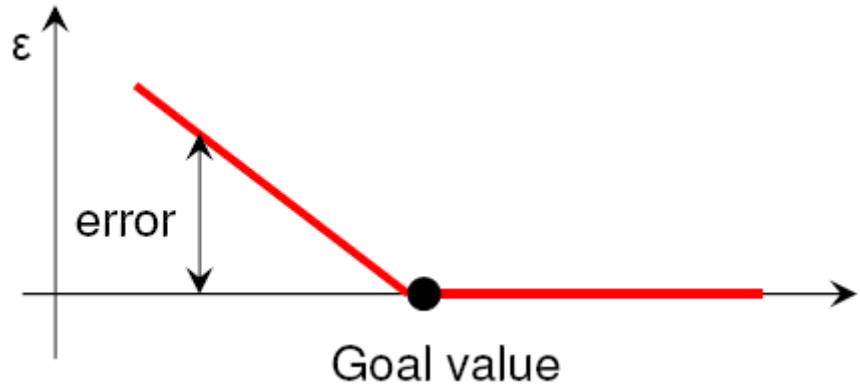
$$Cost = \underset{1}{\overset{N}{Max}} W_i \cdot \varepsilon_i$$

For all the above situations N is the number of individual goals w_i ε_i are individual weighting factors and residual error respectively. A minimization of the cost function is performed during optimization since it makes sense to minimize the error in the sense of the chosen norm type.

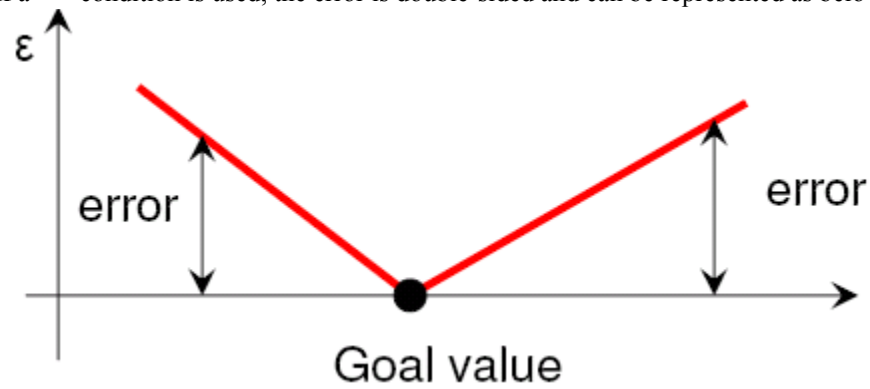
The graphical representation of the error is possible and depends upon the actual condition being used. If a "<" condition is used, the error can be represented as below:



If a ">" condition is used, the error can be represented as below:



If a "=" condition is used, the error is double-sided and can be represented as below:



The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios. Note that when using "minimize" or "maximize" settings for the condition there should be a single goal setting which in this case coincides with the cost function.

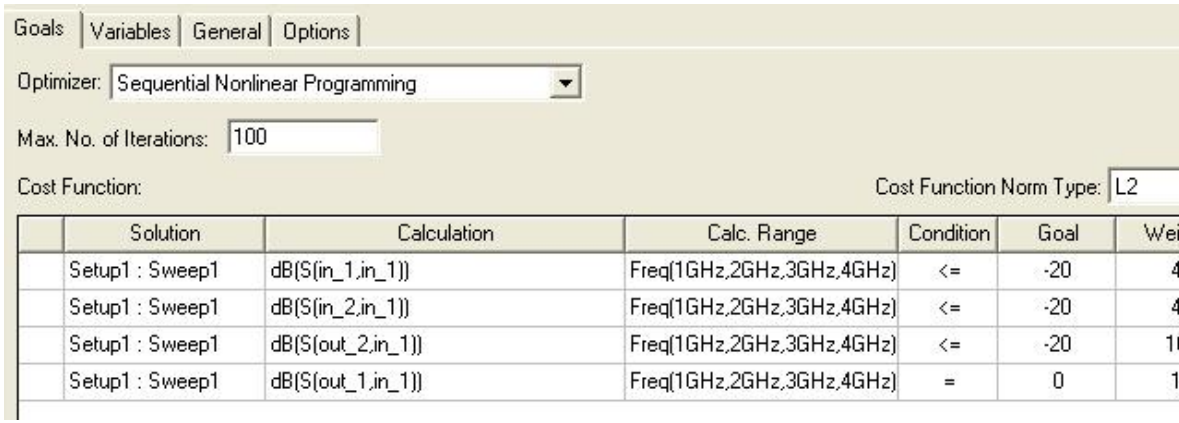
Related Topics

Cost Function

Example of a More Complex Cost Function

Example of a More Complex Cost Function

As an example of a more sophisticated cost function, consider the figure. It belongs to a connector simulated in HFSS with more than four ports.



	Solution	Calculation	Calc. Range	Condition	Goal	Weigh
	Setup1 : Sweep1	dB(S(in_1,in_1))	Freq(1GHz,2GHz,3GHz,4GHz)	<=	-20	4
	Setup1 : Sweep1	dB(S(in_2,in_1))	Freq(1GHz,2GHz,3GHz,4GHz)	<=	-20	4
	Setup1 : Sweep1	dB(S(out_2,in_1))	Freq(1GHz,2GHz,3GHz,4GHz)	<=	-20	10
	Setup1 : Sweep1	dB(S(out_1,in_1))	Freq(1GHz,2GHz,3GHz,4GHz)	=	0	1

The cost function given here concentrates only on a signal sent into port in_1. Suppose the specifications to be met are: reflection, backward cross talk and forward cross talk all smaller than or equal to -20 dB, of which the forward cross talk is the most important.

The first three entries in the cost function enforce those specifications, with the weight for the forward cross talk being a larger number than the other weights. The actual values for the weights are somewhat arbitrary and serve as examples only. For this cost function, as long as specifications are not met, the optimizer puts the most effort in getting the forward cross talk close to its specification. Once the three specifications have been satisfied, their contributions to the cost function become zero, and only the fourth entry remains. Remember that the connector has more than four ports, so satisfying the given specs does not guarantee maximum transmission.

The fourth line tries to maximize the transmission by asking for S(out_1, in_1) to be 0 dB. That will never be reached, but its presence forces the optimizer to improve the connector a bit beyond the specifications.

The cost function norm type specifies how the four lines are combined into one cost function with one value. With L1 and L2, all four contribute simultaneously, rather than only the largest of the four at any one time.

Related Topics

Cost Function

Adding a Cost Function

Advanced Genetic Algorithm Optimizer Options

The Genetic Algorithm (GA) search for Optimization analysis is an iterative process that goes through a number of generations. In each generation some new individuals (Children / Number of Individuals) are created and the so grown population participates in a selection (natural-selection)

process that in turn reduces the size of the population to a desired level (Next Generation / Number of Individuals).

If you select the Genetic Algorithm for an Optimization analysis, a **Setup** button is enabled on the **Setup Optimization** page.

1. Click the **Setup** button to open the **Advanced Genetic Algorithm Optimizer Options** dialog.
2. Select the Stopping Criteria. Any of the three following, or any combination of these can be selected.
   - **Maximum number of generations**. If checked, this enables a value field.
   - **Elapsed time**. If checked, this enables a drop down menu with times ranging from five minutes to two weeks.
   - **Slow convergence**.
3. Specify the Parents.

   The first step towards mating is a selection process that determines the participating individuals. Potential parents are selected from the Current Generation. This is a set of individuals that is always a subset of the current generation.
   - **Number of individuals** value field -- specify the number of parents for the optimizer to use. You can set the Number of Individuals to less than or equal to the size of the "Current Generation". One reason to consider fewer parents than the possible maximum is to steer the GA towards improvement by selecting the better portion of the current generation to be able to mate.
   - **Roulette selection** checkbox -- if checked, this enables the **Selection pressure** value field. This number defines how many times more probable is the selection of the best individual over the worst individual in an elementary spin of the roulette wheel.
4. Specify the Mating pool.

   The Mating pool is created by selecting randomly from the parents, but with each selection, the parent gets "cloned" so it can be selected again and again.
   - **Number of individuals** field -- specify the number individuals to include in the mating pool.
   - **Reproduction setup**-- this button opens the **Genetic Algorithm Optimizer Reproduction Setup** dialog.
5. Click the **Reproduction setup** button for the dialog to specify the Crossover setup, and the Mutation setup.

   The crossover and mutation operator have different roles: *Crossover* mixes "features" of the parents in a new combination, while *mutation* slightly alters the "features" of the individuals. Both need to be present in a GA. The crossover is a way to discover new combinations while the mutation acts as a local search or fine-tuning step. Mutation also keeps diversity in a population, which is a must for GA.

   The crossover operator has two steps. It first alters the variable values of the parents according to a distribution. This tends to produce one child that looks a lot like one parent, and one child that looks a lot like the other parent. Next, some of the variable values of the two children can

be exchanged in order to achieve more variation.

For crossover there are four possible parameters.

a. **Individual Crossover Probability** determines, for each pair in the mating pool, the probability that their features will be mixed. Usually, this probability should be close or equal to one. If you set it to less than one, some parents will produce two children which are exact clones of the parents. This means that some children inherit all the features of their parents unchanged.

b. Parents often have multiple variables. If the parent is a candidate for mixing, the **Variable Crossover Probability** determines, for each variable, the probability of mixing. This is usually set high to ensure that most or all variables mix.

c. **Variable Exchange Probability**: After the slight change in the variable values has been made, the crossover operation is also able to exchange the values of the variables between the two children that are being constructed. The Variable Exchange Probability governs the likelihood of exchange of any variable.

d. **Mu** is a general parameter defining the sharpness of the distribution that might be used for the **Variable Crossover Probability**. Mu should be greater than one. There is no theoretical upper limit, but we recommend not exceeding 30.

6. Select one of the four **Crossover types** from the drop-down menu.

   The crossover type selected affects the options available.

Uniform	Individual crossover probability Variable crossover probability
One point	Individual crossover probability
Two point	Individual crossover probability
Simulated binary crossover	Individual crossover probability Variable crossover probability Variable exchange probability Mu

7. Select the **Mutation type**--this can be one of three types, which you select from a drop-down menu.
   - **Uniform Distribution**
   - **Gaussian Distribution**
   - **Polynomial Mutation.**
8. For the selected mutation type, set the following parameters:
   - **Uniform Mutation Probability**: If this is more than zero (recommendation is to have still a small probability here), then there will be some children whose features are simply a completely random design (design variables randomly selected over the domain).
   - **Individual Mutation Probability** controls, for each child, the likelihood of a mild muta-

tion.

- **Variable Mutation Probability**. If the child will be mutated, this probability controls at the variable level the likelihood of a mutation of the variables.
- **Standard Deviation** is the standard deviation of the selected distribution that is being used for the mutation and it is measured relatively to the optimization-domain.

9. When you have completed the Reproduction setup in the **Genetic Algorithm Optimizer Reproduction Setup** dialog, click **OK** to close it and return to the **Advanced Genetic Algorithm Optimizer Options** dialog.
10. In the **Advanced Genetic Algorithm Optimizer Options** dialog, specify the children as a Number of Individuals.
11. Set the **Pareto Front** value.

    This is the number of the very best individuals (identified relative to the cost function) to keep for future generations.
12. Set the Next Generation parameters. The Next Generation is selected from the Parents, the children, and the Pareto front.
    - **Number of individuals** value field -- specify the number of individuals to survive to form the next generation for the optimizer to use.
    - **Roulette selection** checkbox -- if checked, this enables the **Selection pressure** value field. This number defines how many times more probable is the selection of the best individual over the worst individual in an elementary spin of the roulette wheel.
13. Click **OK** to accept the settings for the Genetic Algorithm and to close the dialog.

Related Topics

Setting up an Optimization Analysis

Adding a cost function

Optimization Overview

Acceptable Cost

Explanation of L1, L2, and Max Norms in Optimization

Choosing an Optimizer

Sensitivity Analysis Overview

During a sensitivity analysis, Optimetrics explores the vicinity of the design point to determine the sensitivity of the design to small changes in variables. The variables and their attributes define the design point, the problem around which the sensitivity analysis is performed.

When Optimetrics performs a sensitivity analysis, its goal is to calculate the second-order regression polynomials for all of the design's output parameters. The algorithm first determines an appropriate interval for each variable. The intervals are further sub-divided according to the available number of iterations and variables. If the master output is not used, the specified initial displacement values define those intervals.

When all of the design calculations are complete, the second-order polynomials are fitted for all the output parameters. Optimetrics then reports the following quantities:

- Regression value at the current variable value.
- First derivative of the regression.
- Second derivative of the regression.

Related Topics

Setting Up a Sensitivity Analysis

Selecting a Master Output

Selecting a Master Output

During a sensitivity analysis, the design variations that Optimetrics selects to solve are close to the design point, but not so close that numerical noise (from the finite element mesh) affects the analysis. The algorithm that Optimetrics uses to determine the design variations to solve must be based on only one output parameter and that output parameter's numerical noise. Therefore, if you have defined more than one output parameter, be sure to select **Master Output** for the output variable on which you want the selection of design variations to be based.

Related Topics

Setting Up an Output Parameter

Setting Up a Sensitivity Analysis

Setting Up a Sensitivity Analysis

Following is the general procedure for setting up a sensitivity analysis. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Before a variable can be included in a sensitivity analysis, you must specify that you intend for it to be used during a sensitivity analysis in the **Design Properties** dialog box.
2. Click **HFSS** >**Optimetrics Analysis**>**Add Sensitivity**  .

   The **Setup Sensitivity Analysis** dialog box appears.
3. Under the **Calculations** tab, type the maximum number of iterations per variable value that you want HFSS to perform in the **Max. No. of Iterations/Sensitivity Variable** text box.
4. Set up an output parameter calculation and select a Master Output
5. Specify the value of the design point at which the sensitivity analysis should stop in the **Approximate Error in Master Output** text box.
6. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Initial Displacement (Initial Disp.)** for the analysis.

   You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.
7. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

   Checking the Optional **Worst Case Analysis** option does a Monte Carlo Analysis that focuses on the upper and lower boundaries of all the analyzed parameters.
8. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

The following ***optional*** sensitivity analysis setup options can also be used:

- Modify the starting variable value.
- Modify the minimum and maximum values of variables that will be solved.
- Exclude variables from the sensitivity analysis.
- Set the initial displacement.
- Modify the values of fixed variables that are not being modified during the sensitivity analysis.

- Set linear constraints.
- Request that Optimetrics solve a parametric sweep **before** a sensitivity analysis.
- You can also request that Optimetrics solve a parametric sweep **during** a sensitivity analysis.

Note Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

Related Topics

Sensitivity Analysis Overview

Setting the Maximum Iteration Per Variable

Setting the Maximum Iterations Per Variable

The **Max. No. of Iterations/Sensitivity Variable** value is the maximum number of design variations that Optimetrics solves per variable during a sensitivity analysis. This value is a stopping criterion; if the maximum number of iterations has been completed, the sensitivity analysis stops. If the maximum number of iterations has not been completed, the sensitivity analysis continues by performing another iteration, that is, by solving another design variation. It performs iterations until the approximate error in master output value is reached or until Optimetrics cannot proceed as a result of other sensitivity setup constraints, such as when it searches for a variable value that is larger than the maximum value.

To set the maximum number of iterations for a sensitivity analysis:

- Under the **Calculations** tab of the **Setup Sensitivity Analysis** dialog box, type a value in the **Max. No. of Iterations/Sensitivity Variable** text box.

Related Topics

Setting Up an Output Parameter

Setting Up an Output Parameter

Following is the general procedure for adding an output parameter to a sensitivity setup:

1. Under the **Calculations** tab of the **Setup Sensitivity Analysis** dialog box, click **Setup Calculations** to open the **Add/Edit Calculations** dialog box.
2. In the **Add/Edit Calculations** dialog box, set up output parameter calculations to be evaluated for sensitivity.
3. To modify the solution from which the output parameter is to be extracted, click in the **Solution** column and select from the options in the pop-up list.
4. You can modify the Calculation specified by clicking on the output parameter in the table and selecting **Edit**.
5. For output parameters based on swept variable, you must choose a single value in the **Calculation Range** at which to evaluate the output parameter.

6. If the output parameter is based on a swept variable, in the **Calculation Range** column, set the value of the variable at which the output parameter is to be computed.
7. If you have more than one output parameter, select **Master Output** if you want Optimetrics to use the output parameter to base its selection of solved design variations.

Note During a sensitivity analysis, the design variations that Optimetrics selects to solve are close to the design point, but not so close that numerical noise (from the finite element mesh) affects the analysis. The algorithm that Optimetrics uses to determine the design variations to solve must be based on only one output parameter and that output parameter's numerical noise. If you have defined more than one output parameter, be sure to select **Master Output** for the output variable on which you want the selection of design variations to be based.

Related Topics

Selecting a Master Output

Specifying a Solution Quantity for an Output Parameter

When setting up an output parameter, you must identify the solution quantity on which to base the output parameter. Solution quantities are specified by mathematical expressions that are composed of basic quantities, such as matrix parameters; and output variables.

The **Add/Edit Calculation** dialog box allows you to define the mathematical equation for one or multiple output parameters. To set up an output parameter:

1. In the **Context** section of the dialog:
   - Select the **Report Type** with a pull-down selection list containing the available types for this design.
   - Select the **Solution** from the drop down selection list. This lists the available setups and sweeps. As a minimum, the **LastAdaptive** solution is available.
   - Select the **Geometry** from the drop down selection list or select none (the default). This modifies the list of quantities available to the ones that apply to the specific geometry.
   - When selecting a geometry, you may also be required to specify a point within the geometry where the calculation is to be performed.
2. The **Output Variables** button opens the Output Variables dialog box allowing you to create special output variables to be used in the output parameter.
3. The **Calculation Expression** field in the **Trace** tab is used to enter the equation to be used for the output parameter. To enter an expression, you may type it directly into the field or use the **Category**, **Quantity**, and **Function** lists as follows:
   - Select the **Category**, these depend on the Solution type and the design. This lets you specify the category of information to be used in the output parameter.
   - Select a **Quantity** from the list. Available quantities depend upon the Solution type, as well as the Geometry and Category selection. Selecting a Quantity automatically enters it into the Calculation Expression field.

- Select a **Function** to apply to the value in the calculated expression.
- For swept variables, the Range**Function** button opens the **Set Range Function** dialog to apply functions to the expression that apply over the sweep range.

4. The **Calculation Range** tab applies to swept variables and allows you to specify the range of the sweep over which to apply the calculation.
5. When the desired **Calculation Expression** has been obtained, click the **Add Calculation** button to add the entry to the calculation table in the Setup Sensitivity Analysis dialog box. You may add multiple entries to the table simply by changing the **Calculation Expression** and using the **Add Calculation** button.
6. To update or edit a selected cost function, enter the desired Calculation Expression and click the **Update Calculation** button.
7. Click **Done** to return to the Setup Sensitivity Analysis dialog box.

Note The solution quantity you specify must be able to be evaluated to a single, real number.

Related Topics

Setting the Calculation Range of an Output Parameter

Setting the Calculation Range of an Output Parameter

The calculation range of a solution quantity determines the intrinsic variable value at which the solution quantity is to be extracted. For a sensitivity setup, the calculation range must be a single value. If you specified that the solution quantity be extracted from a frequency sweep solution, by default, Optimetrics uses the starting frequency in the sweep. If you specified that the solution be extracted from the last adaptive solution, Optimetrics uses the adaptive frequency defined in the solution setup.

1. Under the **Calculations** tab of the **Setup Sensitivity Analysis** dialog box, click in the **Calculation Range** column of the table for the calculation to be modified.

   The **Edit Calculation Range** dialog box appears.
2. In the table, click the **Edit** button in the row to be modified.

   If you choose to solve a parametric setup during the sensitivity analysis, the variables swept in that parametric setup are available in the pop-up list dialog box. If you sweep a variable in the parametric setup that is also a sensitivity variable, that variable is excluded from the sensitivity analysis.

   Other examples of available variables include frequency, if you selected an S-parameter solution quantity; and phi or theta, if the solution quantity is a radiated field quantity.
3. Click on the value for the calculation range in the list and dismiss the pop-up dialog box.
4. Click **OK** in the **Edit Calculation Range** dialog box to accept the new value for the intrinsic variable, and return to the **Setup Sensitivity Analysis** dialog box.

Related Topics

Setting Up an Output Parameter

Modifying the Starting Variable Value for Sensitivity Analysis

The design point of the sensitivity analysis is the starting value of the sensitivity variable and is usually the first variation to be solved. Optimetrics automatically sets the starting value of a variable to be the current value set for the nominal design. You can modify the design point for each sensitivity setup.

Warning Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

1. In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.

   All of the variables that were selected for the sensitivity analysis are listed.

2. Type a new value in the **Starting Value** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is to be used for this sensitivity analysis; the current value set for the nominal model will be ignored.

   - Alternatively, you can select the **Override** option first, and then type a new variable value in the **Starting Value** text box.

3. Optionally, click a new unit system in one of the **Units** text boxes.

To revert to the default starting value, clear the **Override** option.

Related Topics

Setting Up a Sensitivity Analysis

Setting the Min. and Max. Variable Values

For every sensitivity setup, Optimetrics automatically sets the minimum and maximum values that it will consider for a sensitivity variable. Optimetrics sets a variable's minimum value equal to approximately one-half its starting value. (The starting value is the variable's current value set for the nominal design.) Optimetrics sets the variable's maximum value equal to approximately 1.5 times the starting value. During sensitivity analysis, variable values outside this range are not considered.

Warning Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

Related Topics

Override the default minimum and maximum variable values for a single sensitivity setup.

Change the default minimum and maximum variable values for every sensitivity setup.

Overriding the Min. and Max. Variable Values for a Single Sensitivity Setup

1. In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.

   All of the variables that were selected for sensitivity analysis are listed.

2. Type a new value in the **Min** or **Max** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is to be used for this sensitivity analysis; the variable's current **Min** or **Max** value set in the nominal design is ignored.

   - Alternatively, you can select the **Override** option first, and then type a new value in the **Min** or **Max** text box.

3. Optionally, click a new unit system in one of the **Units** text boxes.

To revert to the default minimum and maximum values, clear the **Override** option.

Related Topics

Setting Up a Sensitivity Analysis

Changing the Min. and Max. Variable Values for Every Sensitivity Setup

1. Make sure the variable's minimum and maximum values are not being overridden in any sensitivity setup.
2. If the variable is a design variable, do the following: Click **HFSS>Design Properties**.

   If the variable is a project variable, do the following: Click **Project**>**Project Variables**.

   The **Properties** dialog box appears.

3. Select **Sensitivity**.
4. Type a new value in the **Min** or **Max** text box for the value you want to override, and then press **Enter**.

   When Optimetrics solves a sensitivity setup, it does not consider variable values that lie outside of this range.

Related Topics

Setting Up a Sensitivity Analysis

Setting the Initial Displacement

The initial displacement is the difference in a variable's starting value and the next solved design variation. During the sensitivity analysis, Optimetrics does not consider an initial variable value that is greater than this step size away from the starting variable value.

1. In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
2. Optimetrics displays the **Initial Disp.** column, with default values for each sensitivity variable.
3. In the **Initial Disp.** text box, type the initial displacement value. Optionally, modify the unit system in the **Units** text box.

Related Topics

Setting Up a Sensitivity Analysis

Solving a Parametric Setup Before a Sensitivity Analysis

Solving a parametric setup before a sensitivity setup is useful for guiding Optimetrics in a sensitivity analysis.

To solve a parametric setup before a sensitivity setup:

1. In the **Setup Sensitivity Analysis** dialog box, click the **General** tab.
2. Click the parametric setup you want Optimetrics to solve before the sensitivity setup from the **Parametric Analysis** pull-down list.

Note The parametric setup must include sweep definitions for the sensitivity variables.

3. Select **Solve the parametric sweep before analysis**.

If the parametric setup has not yet been solved, Optimetrics solves it. Optimetrics uses the results (of the solution calculation you requested under the **Goals** tab of the **Setup Sensitivity** dialog box) to determine the next design variation to solve for the sensitivity analysis.

Related Topics

Setting Up a Sensitivity Analysis

Solving a Parametric Setup During a Sensitivity Analysis

Solving a parametric setup during a sensitivity analysis is useful when you want Optimetrics to solve every design variation in the parametric setup at each sensitivity analysis iteration. An output parameter goal could then depend on the value of the variable swept in the parametric setup.

To solve a parametric setup during a sensitivity analysis:

1. In the **Setup Sensitivity Analysis** dialog box, click the **General** tab.
2. Click the parametric setup you want Optimetrics to solve during the sensitivity analysis from the **Parametric Analysis** pull-down list.
3. Select **Solve the parametric sweep during analysis**.

Related Topics

Setting Up a Sensitivity Analysis

Statistical Analysis Overview

Statistical analysis allows you to explore the effects of random combinations of values of selected variables on selected global or local available analysis results. Therefore, before a variable can be included in a statistical analysis, you must specify that you intend for it to be used during a statistical analysis. For each variable you must specify the type of distribution (Uniform, Gaussian, Log-normal or User Defined) and the corresponding parameters of the selected distribution.

In addition to specifying the variables to be used in the statistical analysis and the parameters of the chosen distribution, the output quantities of interest also need to be specified. These quantities can be global ones such as previously defined parameters (Force/torque, inductance / capacitance, etc), other named quantities, quantities defined in the field calculator as global (such a domain integral of a certain field quantity) or local (such as field value at a certain location). The calculations to be performed during the statistical analysis are specified during setup, in a manner similar to other types of analysis in Optimetrics.

Following the analysis the statistical distribution of the output quantities can be visualized in histogram format. To access available reports, after the statistical analysis is complete, right click the respective Statistical analysis setup and select **View Analysis Result**.

Related Topics

Setting Up a Statistical Analysis

Setting Up a Statistical Analysis

Following is the general procedure for setting up a statistical analysis. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Before a variable can be included in a statistical analysis, you must specify that you intend for it to be used during a statistical analysis in the **Properties** dialog box.
2. On the **HFSS** >**Optimetrics Analysis>Add Statistical**  .
3. The **Setup Statistical Analysis** dialog box appears.
4. Under the **Calculations** tab, type the maximum number of iterations you want HFSS to perform in the **Maximum Iterations** text box.
5. Specify a solution quantity to evaluate.
6. In the **Calculation** text box, set the value at which the solution quantity is to be computed.
7. Optionally, modify the distribution criteria to be used.
8. The following ***optional*** statistical analysis setup options can also be used:
   - Modify the starting variable value.
   - Exclude variables from the statistical analysis.
   - Modify the values of fixed variables that are not being modified during the statistical analysis.
   - Request that Optimetrics solve a parametric sweep during a statistical analysis.

Note Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

9. If you want to save the field solution data for the design variations solved during analysis, select **Save Fields**.

Related Topics

Statistical Analysis Overview

Setting the Maximum Iterations for a Statistical Analysis

The **Maximum Iterations** value is the maximum number of design variations Optimetrics solves during a statistical analysis. This value is a stopping criterion; if the maximum number of iterations has been completed, the analysis stops. If the maximum number of iterations has not been completed, Optimetrics continues by performing another iteration, that is, by solving another design variation.

To set the maximum number of iterations for a statistical analysis:

- Under the **Calculations** tab of the **Setup Sensitivity Analysis** dialog box, type a value in the **Maximum Iterations** text box.

Related Topics

Setting up a Statistical Analysis

Specifying the Solution Quantity to Evaluate for Statistical Analysis

When you add a statistical setup, you can identify one or more solution quantities to evaluate. The solution quantities are specified by mathematical expressions that are composed of basic quantities. When you view the results, HFSS displays the distribution of the solution quantities.

1. In the **Calculations** tab of the **Setup Statistical Analysis** dialog box, click **Setup Calculations**.

   The **Add/Edit Calculations** dialog box is displayed, allowing you to define one or more mathematical expressions for statistical evaluation.

2. In the **Context** section of the dialog:
   - Select the **Report Type** with a pull-down selection list containing the available types for this design.
   - Select the **Solution** from the drop down selection list. This lists the available setups and sweeps. As a minimum, the **LastAdaptive** solution is available.
   - Select the **Geometry** from the drop down selection list or select none (the default). This modifies the list of quantities available to the ones that apply to the specific geometry.
   - When selecting a geometry, you may also be required to specify a point within the geometry where the calculation is to be performed.
3. The **Output Variables** button opens the Output Variables dialog box allowing you to create special output variables to be used in the output parameter.
4. The **Calculation Expression** field in the **Trace** tab is used to enter the equation to be used for the solution quantities. To enter an expression, you may type it directly into the field or use the **Category**, **Quantity**, and **Function** lists as follows:
   - Select the **Category**, these depend on the Solution type and the design. This lets you specify the category of information to be used in the output parameter.
   - Select a **Quantity** from the list. Available quantities depend upon the Solution type, as well as the Geometry and Category selection. Selecting a Quantity automatically enters it into the Calculation Expression field.
   - Select a **Function** to apply to the value in the calculated expression.
   - For swept variables, the Range**Function** button opens the **Set Range Function** dialog to apply functions to the expression that apply over the sweep range.
5. The **Calculation Range** tab applies to swept variables and allows you to specify the range of the sweep over which to apply the calculation.
6. When the desired **Calculation Expression** has been obtained, click the **Add Calculation** button to add the entry to the calculation table in the Setup Statistical Analysis dialog box. You may add multiple entries to the table simply by changing the **Calculated Expression** and using the **Add Calculation** button.
7. To update or edit a selected cost function, enter the desired Calculation Expression and click

the **Update Calculation** button.

8. Click **Done** to return to the Setup Statistical Analysis dialog box.

Note The solution quantity you specify must be able to be evaluated to a single, real number.

Related Topics

Setting up a Statistical Analysis

Setting the Maximum Iterations for a Statistical Analysis

Setting the Solution Quantity's Calculation Range

The calculation range of a solution quantity determines the intrinsic variable value at which the solution quantity is extracted. For a statistical setup, the calculation range must be a single value.For a Driven Modal or Driven Terminal design, if you specified that the solution be extracted from the last adaptive solution, Optimetrics uses the adaptive frequency defined in the solution setup. If you specified that the solution quantity be extracted from a frequency sweep solution, Optimetrics will use the starting frequency in the sweep by default.The calculation range should be set during the setup of the solution quantity for statistical evaluation. In order to modify the calculation range, do the following:

1. Under the **Calculations** tab of the **Setup Statistical Analysis** dialog box, click in the **Calculation Range** column of the table for the calculation to be modified.

   The **Edit Calculation Range** dialog box appears.

2. In the table, click the **Edit** button in the row to be modified.

   If you choose to solve a parametric setup during the statistical analysis, the variables swept in that parametric setup are available in the pop-up list dialog box. If you sweep a variable in the parametric setup that is also a statistics variable, that variable is excluded from the statistics analysis.

   Other examples of available variables include frequency, if you selected an S-parameter solution quantity; and phi or theta, if the solution quantity is a radiated field quantity.

3. Click on the value for the calculation range in the list and dismiss the pop-up dialog box.
4. Click **OK** in the **Edit Calculation Range** dialog box to accept the new value for the intrinsic variable, and return to the **Setup Statistical Analysis** dialog box.

Related Topics

Setting up a Statistical Analysis

Setting the Distribution Criteria

For every statistical setup, Optimetrics automatically sets the distribution criteria to be uniform within a 10% tolerance of the variable's starting value. You can modify the distribution type and criteria for a single statistical setup or for every statistical setup.

Related Topics

Override the default distribution criteria for a single statistical setup.

Change the default distribution criteria for every statistical setup.

Overriding the Distribution Criteria for a Single Statistical Setup

To override the default distribution criteria for a single statistical setup:

1. In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.

   All of the variables that were selected for statistical analysis are listed.

2. Check or clear the **Include** checkbox for each variable to define the specific variables to be varied in the statistical analysis setup.

3. For each included variable, select **Uniform**, **Gaussian**, **Lognormal**, or **User Defined** in the **Distribution** column for the variable you want to override.

   If you changed the distribution type, the **Override** option is now selected. This indicates that the distribution type you selected is to be used for this optimization analysis; the current distribution type selected for the variable in the nominal design is ignored in this statistical analysis.

   - Alternatively, you can select the **Override** option first, and then select a different distribution type in the **Distribution** text box.

4. Optionally, if you want to change the distribution criteria, click in **Distribution Criteria** column for the variable you want to override.

   The **Edit Distribution** dialog box appears.

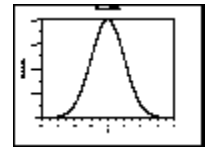
5. If the distribution type is **Gaussian**, do the following:

   a. Type the lower limit of the distribution in the **Cutoff Probability** text box. This is a value =>0 and < 0.1.

   b. Type the mean value of the distribution in the **Mean** text box.

   c. Type the standard deviation of the distribution in the **Std Dev** text box.

      HFSS solves design variations using a Gaussian distribution within the specified mean and standard deviation values.



6. If the distribution type is **Uniform**, do the following:

   - Enter a tolerance value in the text box.

     HFSS will solve design variations within the tolerance range of the starting value, using an even distribution.

7. If the distribution type is **Lognormal**, do the following:

   a. Enter the cutoff probability in the **Cutoff Probability** text box.

   b. Enter the sigma value of the distribution in the **Sigma** text box and select a unit from the pull-down.

c. Enter the m value of the distribution in the **M** text box.

d. Enter the theta value in the Theta text box and select a unit from the pull-down.

8. If the distribution type is User Defined, do the following:
   a. Enter the cutoff probability in the **Cutoff Probability** text box.
   b. Click **Edit XY Data** to open the **Edit Datasets** dialog box in which you can select an existing dataset, or create a new one.
9. By default, all variables are set to sample using **Latin Hypercube** sampling. This sampling method provides for greater variability than random sampling by keeping track of chosen samples and guaranteeing that samples cannot be repeated. You may revert to random sampling by clearing the checkbox in the **Latin Hypercube** column for any desired variable.
10. Click **OK**.

To revert to the default distribution settings, clear the **Override** option.

Related Topics

Statistical Cutoffs

Changing the Distribution Criteria for Every Statistical Setup

To change the default distribution criteria for every statistical setup:

1. Make sure that the variable's distribution criteria are not being overridden in any statistical setup.
2. If the variable is a design variable, do the following: On the **HFSS** >**Design Properties**.

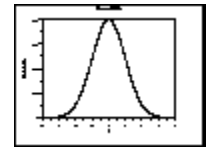
   If the variable is a project variable, do the following: Click **Project**>**Project Variables**.

   The **Properties** dialog box appears.
3. Select **Statistics**.
4. Click in the **Distribution** column for the variable you want to change, and then select **Uniform**, **Gaussian**, **Lognormal**, or **User Defined**.
5. Optionally, if you want to change the distribution criteria, click in the **Distribution Criteria** column for the variable you want to change.

   If the distribution type is **Gaussian**, the **Gaussian Distribution** dialog box appears. If the distribution type is **Uniform**, the **Uniform Distribution** dialog box appears.
6. If the distribution type is **Gaussian**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box. This is a value =>0 and < 0.1.
   b. Type the mean value of the distribution in the **Mean** text box.
   c. Type the standard deviation of the distribution in the **Std Dev** text box.

      HFSS solves design variations using a Gaussian distribution within the specified mean

and standard deviation values.



7. If the distribution type is **Uniform**, do the following:
   a. Type a cutoff probability value in the **Cutoff Probability** text box.
   b. Type mean and tolerance values in the corresponding text boxes.

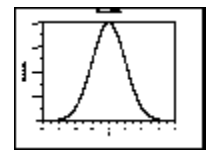
      HFSS will solve design variations within the tolerance range of the starting value, using an even distribution.
8. If the distribution type is **Lognormal**, do the following:
   a. Type a cutoff probability value in the **Cutoff Probability** text box.
   b. Type values for Sigma, M, and Theta in the corresponding text boxes.
9. If the distribution type is **User Defined**, do the following:
   a. Type a cutoff probability value in the **Cutoff Probability** text box.
   b. Click **Edit XY Data** to open the **Edit Dataset** dialog.
   c. Either type or import the X and Y data values for the distribution in the **Edit Dataset** dialog.
10. Click **OK**.

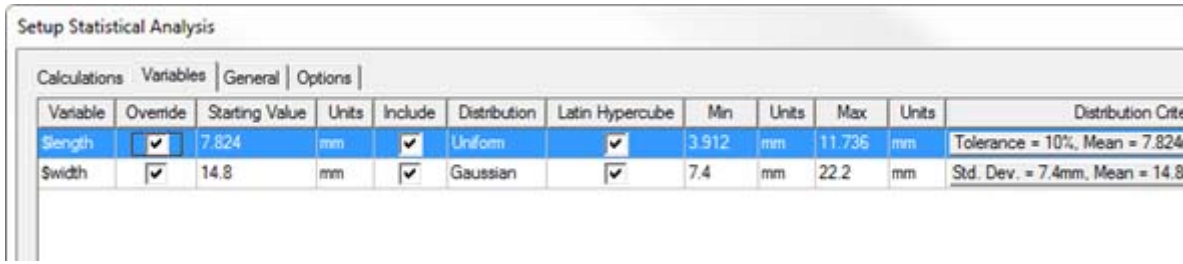
Related Topic

Statistical Cutoff Probability

Statistical Cutoff Probability

The cutoff probability values affects the Gaussian distribution criteria. This is a value =>0 and < 0.1. HFSS solves design variations using a Gaussian distribution using a lower limit cutoff probability and specified mean and standard deviation values.





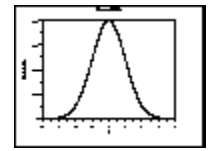
Uniform distributions such as variable "length" above use only the Tolerance value, and do not have a cutoff probability.

Edit Distribution

When setting the distribution type for a variable, you have the option of changing the distribution parameters from the default values.

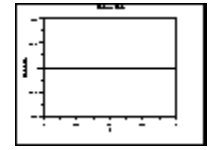
1. If the distribution type is **Gaussian**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box. This is a value =>0 and < 0.1.
   b. Type the mean value of the distribution in the **Mean** text box.
   c. Type the standard deviation of the distribution in the **Std Dev** text box.

      HFSS solves design variations using a Gaussian distribution within the specified mean and standard deviation values.



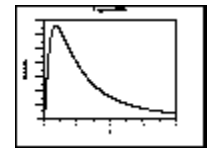
2. If the distribution type is **Uniform**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box.
   b. Type the mean value of the distribution in the **Mean** text box.
   c. Enter the tolerance in the **Tolerance** text box.

      HFSS solves design variations within the tolerance range of the starting value, using an even distribution.



3. If the distribution type is **Lognormal**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box.
   b. Enter the shape parameter of the distribution in the **Sigma** text box.
   c. Enter the scale parameter in the **M** text box. The scale parameter should be set to 1 for the standard lognormal distribution.
   d. Enter the location parameter value for **Theta** in the text box. The value for a standard log-normal distribution is 0.

      HFSS solves design variations with a logarithmic distribution using the shape, scale and location parameters provided..



4. If the distribution type is **User Defined**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box.
   b. Select the **Edit XY Data** button to manually define the data distribution using datasets.

Related Topics

Adding Datasets

Changing the Distribution Criteria for Every Statistical Setup

Overriding the Distribution Criteria for a Single Statistical Setup

Modifying the Starting Variable Value for Statistical Analysis

A variable's starting value is the first value that is solved during the statistical analysis. Optimetrics automatically sets the starting value of a variable to be the current value set for the nominal design. You can modify this value for each statistical setup.

Warning Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

1. In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.

   All of the variables selected for the statistical analysis are listed.
2. Type a new value in the **Starting Value** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is to be used for this statistical analysis; the current value set for the nominal model will be ignored.
   - Alternatively, you can select the **Override** option first, and then type a new variable value

in the **Starting Value** text box.

3. Optionally, click a new unit system in one of the **Units** text boxes.

To revert to the default starting value, clear the **Override** option.

Related Topics

Setting up a Statistical Analysis

Solving a Parametric Setup During a Statistical Analysis

Solving a parametric setup during a statistical analysis is useful when you want Optimetrics to solve every design variation in the parametric setup at each statistical analysis iteration.

To solve a parametric setup during a statistical analysis:

1. In the **Setup Statistical Analysis** dialog box, click the **General** tab.
2. Click the parametric setup you want Optimetrics to solve during the statistical analysis from the **Parametric Analysis** pull-down list.
3. Select **Solve the parametric sweep during analysis**.

Related Topics

Setting up a Statistical Analysis

Tuning Overview

Tuning a variable is useful when you want to manually modify its value and immediately perform an analysis of the design. For example, it is useful after performing an optimization analysis, in which Optimetrics has determined an optimal variable value, and you want to fine tune the value to see how the design results (for example, traces in a report) are affected.

A design can be updated after a tuning analysis to reflect a design variation solved during a tuning analysis and the results, including field solutions if you select **Save Fields and Mesh** on the **Options tab** of the setup.

Related Topics

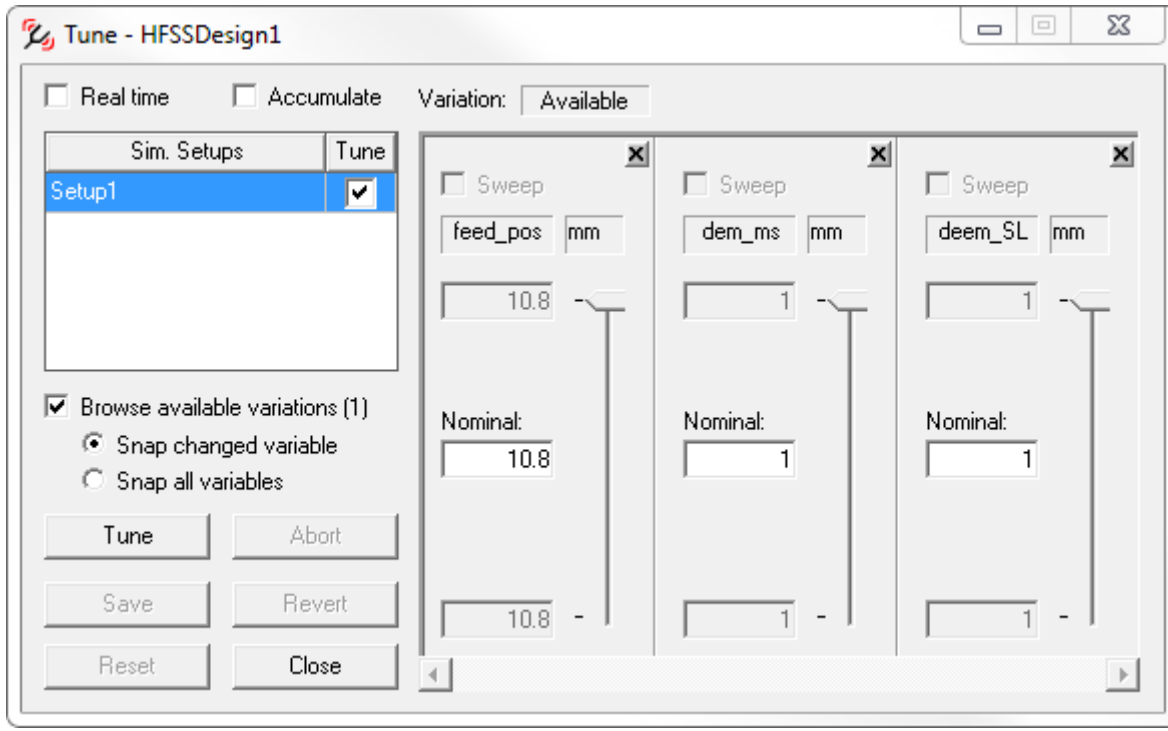
Tuning a Variable

Tuning a Variable

If you want to ensure that tuning does not resolve variations already solved by an optimization setup, you must check **Save Fields and Mesh** in the **Options** tab of that setup.

1. Before a variable can be tuned, you must specify that you intend for it to be used during a tuning analysis in a Project or Design **Properties** dialog box.
2. After running the simulation, click **HFSS >Optimetrics>Tune** .

   The **Tune** dialog box appears., listing the variables which have been included for tuning.



3. Clear the **Real Time** option.

   Clearing the **Real Time** option enables the **Tune** button. If this option is selected, a simulation begins immediately after you move the slider. Otherwise, you use the **Tune** button to apply the current values to a simulation.
4. If you want to see updates to an open Report plot while tuning a post processing variable, you must select the **Browse available variations** checkbox. Selecting **Browse available variations** disables the sweep checkbox, and the fields for minimum and maximum variable values. This feature lets you see the effect of changes to the post processing variables on plotted results.

   Clearing **Browse available variations** enables the Sweep checkbox, the minimum and maximum fields, and changes the Nominal field to Step. See step 6.

5. In the **Sim. Setups** column, select the solution setup you want HFSS to use when it solves the specified design variation.

   HFSS solves the analysis using the solution setup you select. If you select more than one, results are generated for all selected solution setups.

   Checking the Tune box for a Sim Setup enables the Real Time checkbox, the Browse available variations checkbox, and the Snap radio buttons. Clearing the Tune box disables those selections.

6. In the **Nominal** text box for the variable you want to tune, type the value of the variable you want HFSS to solve, or drag the slider to increase or decrease its value.

   Warning Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

   Alternatively, if you want HFSS to solve a range of values, specify a linear range of values with a constant step size:

   a. Select the **Sweep** check box. (You must have cleared the **Browse available variations** checkbox).

   b. In the text box below the **Step** value, type the starting value in the variable range.

   c. Type the step size, or difference between variable values in the sweep definition, in the **Step** text box. The step size determines the number of design variations between the start and stop values. HFSS solves the model at each step in the specified range, including the start and stop values.

   d. In the text box just below the variable name, type a stopping value in the variable range.

7. If you have cleared the Real Time checkbox, click **Tune** to apply the changes you have made to the variable values.

   Note Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

8. Changing a variable value with the sliders of by typing in the text field enables the **Save** and **Reset** buttons.

   Clicking **Save** opens a **Save As** dialog with a name field, and an **Apply tuned values to design** checkbox.

   Clicking **Reset** changes the variable values back to what they were originally.

9. If you have changed one or more included variables, clicking **Close** on the **Tuning** dialog opens the **Apply Tuned Variation** dialog. This lists the included variables and the values for each tuning. If you have tried multiple values, they are listed, and the current value is highlighted. Select another value to change the highlight. Click **OK** to apply the highlighted values to the design, or **Don't Apply** to ignore the changes from the original variable values.

   If you have applied variant values, you should see the new values listed in the relevant Design

or Project Properties lists of variables and values, and if the changes affect plots or physical features of a model, those changes should also appear.

Click **Cancel** to close the dialog and go back to the **Tune** dialog.

Related Topics

Applying a Tuned State to a Design

Tuning Overview

Resetting Variable Values after Tuning

Applying a Tuned State to a Design

You can apply the variable values solved during a tuning analysis to the nominal design in one of the following three ways:

- When closing the **Tune** dialog box:
  1. Click **Close** to exit the **Tune** dialog box.

     The **Apply Tuned Variation** dialog box appears.
  2. Click the design variation you want to apply, and then click **OK**.

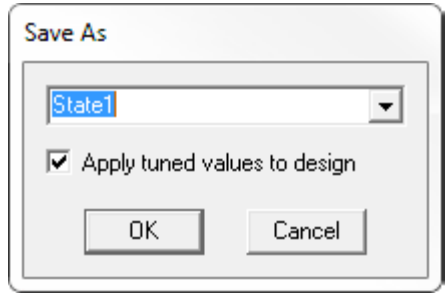
     The variable values from the solved design variation become the current variable values for the nominal design. If you have applied variant values, you should see the new values listed in the relevant Design or Project Properties lists of variables and values, and if the changes affect plots or physical features of a model, those changes should also be apparent.
     - When saving a tuned state.
     - When reverting to a tuned state.

Saving a Tuned State

You can save the settings in the **Tune** dialog box, including the variable values you specified for a tuning analysis. Saved states are only available during the current session of the **Tune** dialog box; they are not stored for the next session.

1. After tuning a variable, click **Save** in the **Tune** dialog box.

   A **Save As** dialog box appears.



2. Type a name for the tuned state in the text box.
3. Select **Apply tuned values to design** if you want to update the model to the new variable val-

ues.

4. Click **OK** to return to the **Tune** dialog box.

Related Topics

Reverting to a Saved Tuned State

Reverting to a Saved Tuned State

You can revert to a group of saved settings in the **Tune** dialog box, including the variable values you specified for a specific tuning analysis. Saved states are only available during the current session of the **Tune** dialog box; they are not stored for the next session.

1. In the **Tune** dialog box, click **Revert**.

   The **Revert** dialog box appears.

2. Type the name of the tuned state you want to apply or click a name in the pull-down list.
3. Select **Apply tuned values to design** if you want to update the model to the selected tuned state's variable values.
4. Click **OK** to return to the **Tune** dialog box.

Related Topics

Saving a Tuned State

Resetting Variable Values after Tuning

If you want to reset variable values to the values they were set to when you started the current session of the **Tune** dialog box:

- After tuning a variable, click **Reset** in the **Tune** dialog box.

  Solutions for the design variations solved during tuning analyses remain available for post processing.

Saving Field Solutions for Optimetrics Analyses

In order to preserve disk space, by default HFSS does not save field solution data for every solved design variation in an optimization analysis. It only saves the field solutions for the nominal design when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the optimization analysis, all field solutions are deleted.

To save the fields for all design variations, change the default setting for all projects:

1. Select **Tools**>**Options**, and then select either **HFSS Options**.

   The appropriate **Options** dialog box appears.

2. Under the **General** tab, select **Save Optimetrics field solutions**.

   Save Fields is selected by default when you create a new Optimetrics setup.

Related Topics

Saving Field Solutions for a Parametric Setup

Saving Field Solutions for an Optimization Setup

Saving Field Solutions for a Sensitivity Setup

Saving Field Solutions for a Tuning Analysis

Saving Field Solutions for a Statistical Setup

Copy Geometrically Equivalent Meshes

Saving Field Solutions for a Parametric Setup

In order to preserve disk space, by default HFSS does not save field solution data for every solved design variation in a parametric setup. It only saves the field solutions for the nominal design. If the nominal design is not included in the parametric setup, by default field solutions will not be available.

To save the fields for all design variations solved during a parametric analysis:

1. Either **Add Sweep** or right click on an existing sweep to open the **Setup Sweep Analysis** dialog box.
2. Select the **Options** tab.
3. Click the **Save Fields And Mesh** check box. Optionally, select **Copy geometrically equivalent meshes**.

   HFSS will save the field solution data for every solved design variation in the parametric setup.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Saving Field Solutions for an Optimization Setup

In order to preserve disk space, by default HFSS does not save field solution data for every solved design variation in an optimization analysis. It only saves the field solutions for the nominal design

when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the optimization analysis, all field solutions are deleted.

To save the fields for all design variations solved during an optimization analysis:

1. Open an **Edit Sweep** dialog by either adding a sweep or right-click on a an existing sweep to view the short cut menu and selecting Properties.
2. Select the **Options** tab.
3. Click the **Save Fields And Mesh** check box. Optionally, select **Copy geometrically equivalent mashes**.

   HFSS will save the field solution data for every solved design variation in the optimization setup.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Saving Field Solutions for a Sensitivity Setup

In order to preserve disk space, by default HFSS does not save field solution data for every solved design variation in a sensitivity analysis. It only saves the field solutions for the nominal design when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the sensitivity analysis, all field solutions are deleted.

To save the fields for all design variations solved during a sensitivity analysis:

1. Open the **Setup Sensitivity Analysis** dialog box.
2. Select the **Options** tab.
3. Click the **Save Fields And Mesh** check box. Optionally, select **Copy geometrically equivalent mashes**.

   HFSS will save the field solution data for every solved design variation in the sensitivity analysis.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Saving Field Solutions for a Tuning Analysis

In order to preserve disk space, by default HFSS does not save field solution data for every design variation solved in a tuning analysis. It only saves the field solutions for the nominal design when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the tuning analysis, all field solutions are deleted.

To save the fields for all design variations solved during a tuning analysis:

- In the **Tuning** dialog box, select **Save Fields**.

  HFSS will save the field solution data for every solved design variation in a tuning analysis.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Saving Field Solutions for a Statistical Setup

In order to preserve disk space, by default HFSS does not save field solution data for every design variation solved in a statistical analysis. It only saves the field solutions for the nominal design when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the statistical analysis, all field solutions are deleted.

To save the fields for all design variations solved during a statistical analysis:

1. Open the **Setup Statistical Analysis** dialog box.
2. Select the **Options** tab.
3. Click the **Save Fields And Mesh** check box. Optionally, select **Copy geometrically equivalent mashes**.

   HFSS will save the field solution data for every solved design variation in the statistical setup.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Copying Meshes in Optimetrics Sweeps

An option in the Optimetrics Analysis setups allows you to request HFSS to copy a mesh that was calculated for one sweep variation for reuse on a geometrically-equivalent sweep variation. For example, with this option selected a sweep on a scan angle would not need to generate meshes for each solution. The option is available on the setups for sweeps on parametrics, optimization, sensitivity, and statistics.

To copy and reuse meshes on geometrically-equivalent parametric variations:

1. Define a variable for the kind of Optimetrics sweep you intent to setup.
2. Select **HFSS** and then select the appropriate **Optimetrics>Add** command to display a **Setup** dialog box.
3. Click the **Options** tab in the Setup dialog box.
4. Select **Copy geometrically equivalent meshes**.

   HFSS will copy the mesh solution calculated for a particular parametric sweep for reuse on each geometrically-equivalent sweep variation.

Note This option is available with all Optimetrics setups, and is applied when these analyses generate geometrically-equivalent values. However, it is most relevant to parametric sweep, where such equivalences are more likely to occur.

The **Copy geometrically equivalent mesh** option is not recommended for use when the frequency is varying, since meshing is frequency-dependent. You may wish to turn this option off when the first geometrically equivalent variation requires numerous passes after the initial mesh, but the other geometrically-equivalent variations require fewer additional passes, so that it is cheaper to start with the initial mesh each time.

Adding an Expression in the Output Variables Window

When you are in the **Output Variables** window (after clicking **Edit Calculation** from the one of the setup analysis windows), do the following to specify an expression:

1. Type a name for the expression in the **Name** text box.
2. Do the following in the **Calculation** section of the window to insert a quantity into the expression:
   a. Select the **Report Type** and **Solution** from the pull-down lists.
   b. Select a **Category**, **Quantity**, and **Function** from the lists, and click **Insert Quantity Into Expression**.
   c. If you want to insert a specific pre-defined function, select one from the **Function** pull-down list, and click **Insert Function**.
3. You can also type numbers or expression by hand directly into the **Expression** area.

Excluding a Variable from an Optimetrics Analysis

To exclude a variable from being optimized or included in a sensitivity or statistical analysis:

1. Do one of the following:
   - In the **Setup Optimization** dialog box, click the **Variables** tab.
   - In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
   - In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.

   All of the independent variables that were selected for the optimization analysis are listed.
2. Clear the **Include** option for the variable you want to exclude from the analysis.

   The **Override** option is now selected. This indicates that, for this optimization analysis, the variable is not included.

Note Alternatively, you can select the **Override** option first, and then clear the **Include** option for the variable you want to exclude.

3. Click **OK**.

Modifying the Value of a Fixed Variable

If you are not including a variable in an optimization, sensitivity, or statistical analysis, Optimetrics uses that variable's current value during the analysis.

To override the current value of a fixed variable for an Optimetrics setup:

1. Do one of the following:
   - In the **Setup Optimization** dialog box, click the **Variables** tab.
   - In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
   - In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.
2. Click **Set Fixed Variables**.

   The **Setup Fixed Variables** dialog box appears. Under **Fixed Variables**, all of the current independent variable values are listed.
3. Click the **Value** text box of the variable with the value you want to override.
4. Type a new value in the **Value** text box, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is used for this Optimetrics setup; the current variable value set for the nominal design is ignored.

Note Alternatively, you can select the **Override** option first, and then type a new value in the **Value** text box.

5. Optionally, click a new unit system in the **Units** text box.
6. Click **OK**.

To revert to a default variable value, clear the **Override** option.

Linear Constraints

Once the optimization variables are specified, the optimizer handles each of them as an *n*-dimensional vector *x*. Any point in the design space corresponds to a particular *x*-vector and to a design instance. Each design instance may be evaluated via Finite Element Analysis and assigned a cost value; therefore, the cost function is defined over the design space $\mathrm{cost}(x): R^n \rightarrow R$, where *n* is the number of optimization variables.

In practice, a solution of the minimization problem is sought only on a bounded subset of the R^n space. This subset is called the feasible domain and is defined via linear constraints.

You may constrain the feasible domain of a design variable by defining linear constraints for the optimization process. The feasible domain is defined as the domain of all design variables that satisfy all upper and lower bounds and constraints. Linear constraints are defined by the following inequalities:

$$\sum_i \alpha_{ij} x_i < c_j \forall j$$

where

- α_{ij} are coefficients.
- c_j is a comparison value for the j^{th} linear constraint.
- x_i is the i^{th} designer parameter.

Related Topics

Setting a Linear Constraint

Setting a Linear Constraint

A linear constraint defines the linear relationship between variables. Setting linear constraints in Optimetrics is useful for establishing limitations involving linear combinations of variable values.

1. Do one of the following:
   - If you are setting up an optimization analysis: In the **Setup Optimization** dialog box, click the **Variables** tab.
   - If you are setting up a sensitivity analysis: In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
2. Click **Linear Constraint**.

   The **Linear Constraint** dialog box appears.
3. Click **Add**.

   The **Edit Linear Constraint** dialog box appears.
4. Click a **Coeff** text box and type a positive or negative coefficient value.
5. Click a condition, < (less than) or > (greater than), from the pull-down list.

6. Type the inequality value, which should be a constant value, in the text box to the right of the condition.
7. Click **OK**.

   You return to the **Linear Constraint** dialog box. The left-hand side of the constraint appears in the **LHS** (left-hand side) column. The condition is listed in the **Condition** column, and the inequality value is listed in the **RHS** (right-hand side) column.

Related Topics

Modifying a Linear Constraint

Deleting a Linear Constraint

Linear Constraints

Modifying a Linear Constraint

1. Do one of the following:
   - If you are setting up an optimization analysis: In the **Setup Optimization** dialog box, click the **Variables** tab.
   - If you are setting up a sensitivity analysis: In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
2. Click **Linear Constraint**.

   The **Linear Constraint** dialog box appears.
3. Click the row listing the constraint you want to modify, and then click **Edit**.

   The **Edit Linear Constraint** dialog box appears.
4. Optionally, click a **Coeff** text box and type a new coefficient value.
5. Optionally, click a different condition, < (less than) or > (greater than), in the pull-down list.
6. Optionally, type a different inequality value in the text box to the right of the condition, and then click **OK**.

   You return to the **Linear Constraint** dialog box. The new coefficient value, the condition, and the inequality value appear in the **LHS** (left-hand side), **Condition**, and **RHS** (right-hand side) columns, respectively.

Deleting a Linear Constraint

1. Do one of the following:
   - If you are setting up an optimization analysis: In the **Setup Optimization** dialog box, click the **Variables** tab.
   - If you are setting up a sensitivity analysis: In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
2. Click **Linear Constraint**.

   The **Linear Constraint** dialog box appears.
3. Click the row listing the constraint you want to delete, and then click **Delete**.

The constraint is deleted.

Running an Optimetrics Analysis

Once you have created all necessary Optimetrics based analyses, you have several options for running the simulations.

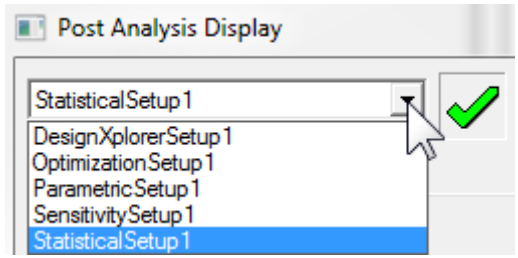
- To use the **Analyze All** command at the Project or design level to simulate the nominal problem and subsequently run all Optimetrics setups, do the following:
    1. In the Project Manager window, right-click on the **project** or **design** name.
    2. Click **Analyze All** from the shortcut menu.
- To use the **Analyze All** command from the Optimetrics menu to simulate only the Optimetrics based setups, do the following:
    1. In the Project Manager window, right-click on **Optimetrics**.
    2. Click **Analyze>All** from the shortcut menu.
- You can choose to analyze only the setups related to a specific Optimetrics type of analysis. In order to simulate setups of a specific type, do the following:
    1. In the Project Manager window, right-click on **Optimetrics**.
    2. Click **Analyze>All {*TYPE*}** from the shortcut menu where *TYPE* is the specific analysis type of interest, Parametric, Optimization, Sensitivity, or Statistical.

Viewing Analysis Results for Optimetrics Solutions

To view data specific to an Optimetrics solution, in general, do the following:

- In the project tree, right-click the Optimetrics setup for which you want to view the results, and then click **View Analysis Result** on the shortcut menu.

  The **Post Analysis Display** dialog box appears.
- Select from available setups by using the dropdown selection menu.



- Select the **Results** tab to view results in plot or table form. When you view results in Table form you can resort the results based on each column. Click the Iteration column head to invert the sort from lowest to highest setup number. Click the variable name column to resort the results by step value. Click the Cost column head to sort the results from lowest cost to highest cost. Clicking a column again inverts the current sort.
- Click the **Options...** button to open a dialog that permits you to specify the Maximum number of significant digits to display when showing the analysis result The default is 4.
- Select the **Profile** tab to view start, stop, and elapsed times for each variable, and the analysis machine for each variation. You can click the column heads to sort the table by variation number, variable value, start, stop, or elapsed time, or (if you have run a distributed analysis) machine.

See the help topics in this section for more details about viewing Optimetrics analysis results.

Related Topics

Viewing Solution Data for an Optimetrics Design Variation

Viewing an Optimetrics Solution's Profile Data

Viewing Results for Parametric Solution Quantities

Viewing Cost Results for an Optimization Analysis

Viewing Output Parameter Results for Sensitivity Analysis

Viewing Distribution Results for Statistical Analysis

Viewing Solution Data for an Optimetrics Design Variation

To view the convergence information, computing resources used, or matrices computed for any design variation solved during an optimization analysis, you must first select the design variation in

the **Set Design Variation** dialog box. This dialog box is accessible from the **Solutions** Data window and via the **Results>Apply Solved Variation** command in the or **HFSS** menu.

1. Click **HFSS** and then select **Results>Solution Data**.

   The **Solutions** dialog box appears.
2. Click the browsing dots beside the **Design Variation** box.

   The **Set Design Variation** dialog box appears.
3. Clear the **Use nominal design** option.
4. Click the design variation for which you want to view the solution data, and then click **OK**.

   The solution data is displayed in the table.

Related Topics

Viewing an Optimetrics Solution's Profile Data

Viewing an Optimetrics Solution's Profile Data

At any time during or after the Optimetrics solution process, you can see an overview of the computing resources or profile data that was used by HFSS as it solved each design variation. Optimetrics writes the variation information to the profile table before the solve. It then updates the entry with end data (end time, elapsed time, etc) once the solve variation is completed.

1. In the project tree, right-click the Optimetrics solution setup of interest, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Click the **Profile** tab.
3. Select the Optimetrics setup with the results you want to view from the pull-down list at the top of the dialog box.
4. Optionally, to examine more detailed profile data for a specific design variation, do the following:
   a. Click a design variation in the table.
   b. Click **Solver Profile**.

The **Solutions** dialog box appears with the profile data for the selected design variation.

The profile line for the matrix solver is in the following format:

`Solver` *123*

where:

- *1* is the precision type: M (mixed) or D (double)
- *2* is the matrix data type: R (real) or C (complex)
- *3* is the symmetry type: S (symmetric), A (asymmetric), H (hermitian)

Related Topics

Viewing a Solution's Profile

Viewing Solution Data for an Optimetrics Design Variation

Viewing Results for Parametric Solution Quantities

1. In the project tree, right-click the parametric setup for which you want to view the results calculated for the solution quantities, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Select the parametric setup with the results you want to view from the pull-down list at the top of the dialog box.
3. If it is not already selected, select **Table** as the view type.

   The results for the selected solution quantities are listed in table format for each solved design variation. The variation column in the table lists the entries in order. Clicking the Vision header inverts the order. Clicking other headers sorts the entries by value, and clicking again inverts the order.
4. Optionally, select **Show complete output name**.

   The complete name of the solution for which the results are being displayed will be listed in the column headings.
5. Optionally, click a design variation in the table, and then click **Apply** (at the far right side of the dialog box).

   The design displayed in the **3D Modeler** window is changed to represent the selected design variation.

Related Topics

Plotting Solution Quantity Results vs. a Swept Variable

Plotting Solution Quantity Results vs. a Swept Variable

To plot solution quantity results versus a swept variable's values on a rectangular (x - y) plot:

1. In the project tree, right-click the parametric setup for which you want to view the results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. If it is not already selected, select **Plot** as the view type.
3. Select the variable with the swept values you want to plot on the x-axis from the **X** pull-down list.
4. Only one sweep variable at a time can be plotted against solution quantity results. Any other variables that were swept during the parametric analysis remain constant.

   Optionally, to modify the constant values of other swept variables, do the following:

   a. Click **Set Other Sweep Variables Value**.

      The **Setup Plot** dialog box appears. All of the other solved variable values are listed.

   b. Click the row with the variable value you want to use as the constant value in the plot, and then click **OK**.
5. Select the solution quantity results you want to plot on the y-axis from the **Y** pull-down list.

   The xy plot appears in the view window.

6. Right-click in the plot area to get the shortcut menu where you can set modify the plots display properties, print, copy to the clipboard, or export the data to a file.

Viewing Cost Results for an Optimization Analysis

To view cost values versus completed iterations in data table format:

1. In the project tree, right-click the optimization setup for which you want to view the cost results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.

2. Under the **Result** tab, select **Table** as the view type, if it is not already selected.

   The cost value at each solved design variation is listed in table format.

3. Optionally, click a design variation in the table, and then click **Apply**.

   HFSS now points to the selected design variation as the nominal solution and as a result, the design displayed in the **Modeler** window is changed to represent the selected design variation.

   Click **Revert** to return the design in the view window to the original value.

Related Topics

Plotting Cost Data for an Optimization Analysis

Viewing Solution Data for an Optimetrics Design Variation

Plotting Cost Results for an Optimization Analysis

To view cost values versus completed iterations in rectangular (x-y) plot format:

1. In the project tree, right-click the optimization setup for which you want to view the cost results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.

2. Under the **Result** tab, select **Plot** as the view type.

   A plot of the cost value at each iteration appears.

Viewing Output Parameter Results for a Sensitivity Analysis

To view actual output parameter values versus design point in data table format:

1. In the project tree, right-click the sensitivity setup for which you want to view the parameter results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.

2. Under the **Result** tab, select **Table** as the view type, if it is not already selected.

   The following values are listed in table format:

   - The regression value of the output parameter at the design point is listed in the **Func. Value** column.
   - The first derivative of the regression is listed in the **1st D** column.
   - The second derivative of the regression is listed in the **2nd D** column.

3. Click **Apply**.

HFSS now points to the selected design variation as the nominal solution and as a result, the design displayed in the **Modeler** window is changed to represent the selected design variation.

Click **Revert** to return the design in the view window to the original value.

Related Topics

Plotting Output Parameter Results for a Sensitivity Analysis

Viewing Solution Data for an Optimetrics Design Variation

Plotting Output Parameter Results for a Sensitivity Analysis

To plot output parameter results versus sensitivity variable values on a rectangular (xy) plot:

1. In the project tree, right-click the sensitivity setup for which you want to view the output parameter results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Under the **Result** tab, select **Plot** as the view type.
3. Select the sensitivity variable with the sweep values you want to plot on the x-axis from the **X** pull-down list.
4. Select the output parameter results you want to plot on the y-axis from the **Y** pull-down list.

   The xy plot appears in the **Post Analysis Display** dialog box.

   The plot displays actual output parameter results for each solved design variation. It also displays a parabola that best fits these results. The parabola is a more accurate representation of sensitivity around the design point than any individual solved design variation.

Viewing Distribution Results for a Statistical Analysis

1. In the project tree, right-click the statistical setup for which you want to view the distribution results calculated for the solution quantities, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Select the statistical setup with the results you want to view from the pull-down list at the top of the dialog box.
3. To view the results in tabular form, select **Table** as the view type.

   The distribution results for the selected solution quantities are listed in table format for each solved design variation.
4. Optionally, click a design variation in the table, and then click **Apply** (at the far right side of the dialog box).

   The design displayed in the **3D Modeler** window is changed to represent the selected design variation.
5. To view the results in graphic format, select **Plot** as the view type.
6. Type the number of bins you want to plot on the x-axis.
7. Select the solution quantity for which you want to plot distribution results on the y-axis from the **Y** pull-down list.

A histogram plot appears in the **Post Analysis Display** dialog box. It displays the distribution of the selected solution quantity.

8. Optionally, click a design variation in the table, and then click **Apply** (at the far right side of the dialog box).

   HFSS now points to the selected design variation as the nominal solution and as a result, the design displayed in the **Modeler** window is changed to represent the selected design variation.

   Click **Revert** to return the design in the view window to the original value.

Related Topics

Plotting Distribution Results for a Statistical Analysis

Viewing Solution Data for an Optimetrics Design Variation

Plotting Distribution Results for a Statistical Analysis

1. In the project tree, right-click the statistical setup for which you want to view the distribution results calculated for the solution quantities, and then click **View Analysis Result** on the short-cut menu.

   The **Post Analysis Display** dialog box appears.

2. Select the statistical setup with the results you want to view from the pull-down list at the top of the dialog box.
3. If it is not already selected, select **Plot** as the view type.
4. Type the number of bins you want to plot on the x-axis.
5. Select the solution quantity for which you want to plot distribution results on the y-axis from the **Y** pull-down list.

   A histogram plot appears in the **Post Analysis Display** dialog box. It displays the distribution of the selected solution quantity.

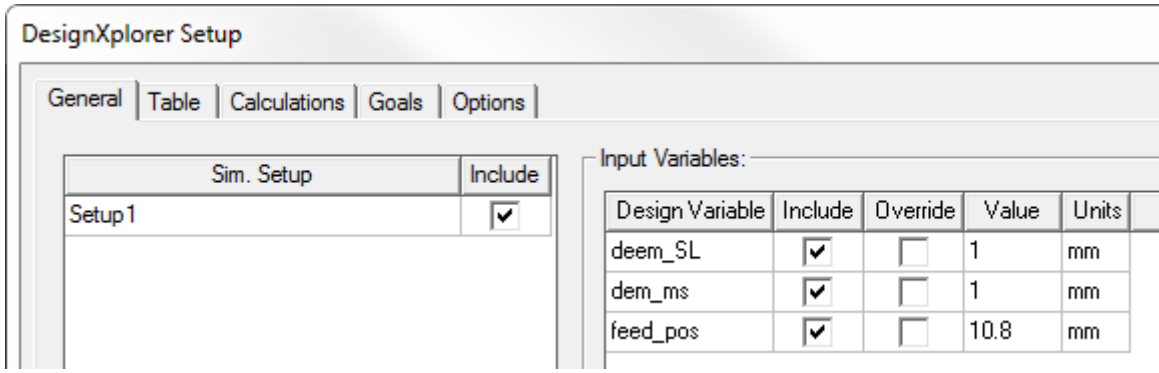
Link to Design Xplorer

You can export a .xml file containing information on an HFSS setup, optimization variables, and output variables that enables ANSYS Design Xplorer to manage HFSS simulations, for example, for design of experiments and optimization. Design Xplorer will launch HFSS simulations of design variations and evaluate the HFSS outputs.

To do so:

1. Click **HFSS>Optimetrics Analysis>Add Design Xplorer Setup..**. or right-click on **Optimetrics** in the **Project** window, and select **Add Design Xplorer Setup...** from the short-cut menu.

   This opens the **Design Xplorer** dialog with the **General** tab selected. it lists the setups available in the current project, and the input variables it contains.

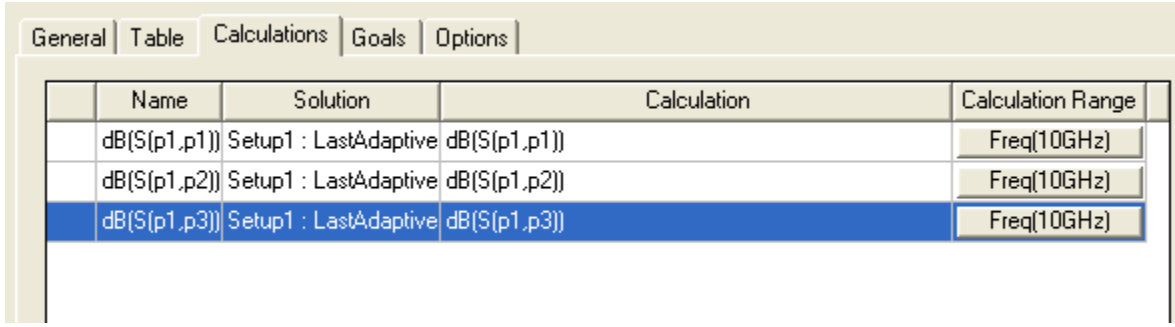


2. Check Include for the simulation setups you want to use.
3. Check the Design variables to use. You can also chose to Override the value of a design variable. You can edit the Value and Units fields. Unchecking Override returns the values to their original state.
4. To setup any output calculations, click the **Calculation** tab and click the **Setup Calculations** button.

   This opens the **Add/Edit Calculation** dialog. Here you can define the simulation results of interest. The dialog box contains distinct panes and tabs to set the **Context**, the **Calculation Expression**, and the **Calculation Range**. See Setup Calculations for Optimetrics for details.

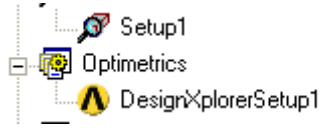
   Use the **Add Calculation** button to add expressions to the Calculations table of the **Design**

Xplorer Setup dialog, **Calculations** tab.

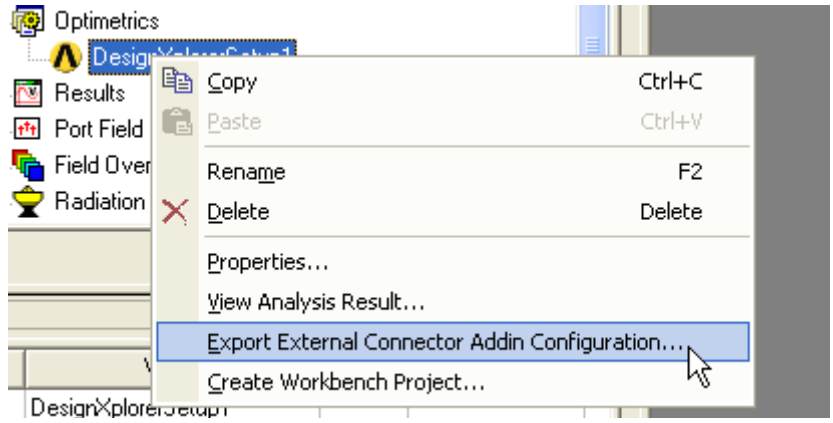


5. When you have added the calculations of interest, click **OK** to save the setup.

   An icon for the Design Xplorer setup appears under Optimization in the Project tree.



6. To create a .xml file with the setup information for Design Xplorer, first Save your project.
7. Then right-click on the setup and select **Export External Connector Addin Configuration.**



   This displays a browser dialog that you can use to navigate your file system, and name and saves the .xml file. The .xml file contains information regarding the HFSS path along with the setup, variables, and simulation results that you specified.

8. If you have an ANSYS Workbench installation you can perform additional steps. You should have provided a path to the Workbench installation in the **Tools>General Options** dialog Miscellaneous tab, to provide a path.
9. Then click **Create Workbench Project**.

   This lets you name a Workbench project containing the information in the setup. The ANSYS Workbench will be launched with the connection to the HFSS project established. To this con-

nection, you can add a Design Xplorer Setup. See the documentation of Ansys Workbench for details on Design Xplorer.

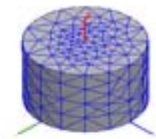
17

Post Processing and Generating Reports

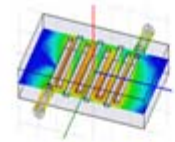
When HFSS has completed a solution, you can display and analyze the results as follows:

View Solution Data

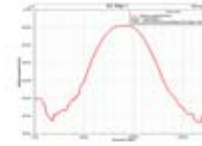
View Optimetrics Results



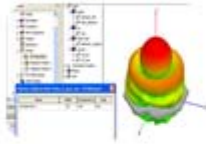
Create Mesh Plots



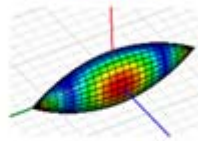
Plot Field Overlays



Create or Modify 2D or 3D Reports

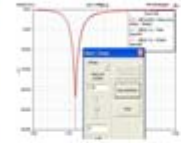


Overlay 3D Field Plot on Model

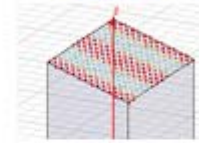


Create Animations

Scale Excitation Mag/Phase



Tune Reports



Port Field Display

Specify Output Variables

Radiated Fields Post Processing

Export Equivalent Circuit Data

Fields Calculator / Derived Field Quantities

User Defined Outputs

Toolkit

User Defined Documents

Network Data Explorer

- View solution data including the following: convergence information, computing resources that were used during the solution process, mesh statistics, and matrices computed for the S-parameters, impedances, and propagation constants during each adaptive, non-adaptive, or sweep solution. For eigenmode solutions, you can view the real and imaginary parts of the frequency and quality factor Q computed for each eigenmode. Solution data can also be viewed while HFSS is generating a solution.
- View analysis results for Optimetrics solutions.
- Plot field overlays - representations of basic or derived field quantities - on surfaces or objects.
- Overlay Field Plots on Models
- Create 2D or 3D reports of S-parameters, basic and derived field quantities, and radiated field data.
- Plot the finite element mesh on surfaces or within 3D objects.
- Create animations of field quantities, the finite element mesh, and defined project variables.
- Scale an excitation's magnitude and modify its phase.
- Apply Derivative Tuning to Reports

Note Except in the case of non-model boxes drawn in the global coordinate system (CS), non-model objects cannot be used for any fields post processing operation You can use non-model boxes drawn in the global CS for post processing operations, including integration and solution domaining.

Viewing Solution Data

While HFSS is generating a solution, or when it is complete, you can view the following information about the solution:

- Convergence information.
- Computing resources, or profile information, that were used during the solution process.
- Matrices computed for the S-parameters, impedances, and propagation constants during each adaptive, non-adaptive, or sweep solution.
- Mesh statistics
- For eigenmode solutions, view the real and imaginary parts of the frequency and quality factor Q computed for each eigenmode.
- The state of solved solutions.
- For transient solutions, Transient Data.

To access the **Solution Data** window, in which the information above can be accessed, do one of the following:

- Click **HFSS**>**Results**>**Solution Data**  .
- Right-click **Results** in the project tree, and then click **Solution Data** on the shortcut menu.

Related Topics

Viewing Solution Data for an Optimetrics Design Variation

Viewing Convergence Data

To view an adaptive solution's convergence information, either during or after the solution process:

1. In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

   The **Solution Data** window appears. The **Convergence** tab is selected.
2. From the **Simulation** list, select the solution setup for which you want to view convergence data.

   By default, the most recently solved solution is selected.
3. Under the **Convergence** tab, depending on your design setup, you can review the following convergence data:
   - Whether the solution is converged or not converged.
   - Number of adaptive passes completed and remaining.
   - The Solved Elements at each adaptive pass, which includes solve inside tetrahedra, and, for projects using IE Regions for metal objects, also includes the number of solved IE surface triangles.
   - Maximum magnitude of delta S between two passes.
   - Maximum delta Energy between two passes.
   - Magnitude margin between passes.

- Phase margin (deg) between passes.
- Maximum delta frequency between passes.

If for the Solution Setup, you elected to **Use Matrix Convergence**, and selected specific table entries for the Magnitude and Phase, the **Convergence** tab also shows the following values with the Magnitude Margin and Phase Margin:

- Max Delta (Mag S)
- Max Delta (Phase S)

4. Select **Table** to display the convergence data in table format or **Plot** to plot the convergence data on a rectangular (X - Y) plot.

Note If you receive a message that the eigenmodes have not converged, it may indicate that the existing mesh is too coarse. You may need to refine the mesh.

Related Topics

Viewing Solution Data for an Optimetrics Design Variation

Viewing the Number of Completed Passes

At any time during the solution process, you can view the number of adaptive passes (solve — error analysis — refine cycles) that have been completed and that have yet to be completed. When the solution is complete, you can view the number of adaptive passes that were performed. If the solution converged within the specified stopping criteria, fewer passes than requested may have been performed.

To view the number of passes:

- In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

  The **Solution Data** window appears. The **Convergence** tab is selected.

  The number of completed and remaining passes is listed in the **Number of Passes** area.

Viewing the Max Magnitude of Delta S Between Passes

For solutions with ports.

At any time during or after the solution process, you can view the maximum change in the magnitude of the S-parameters between two consecutive passes. This information is available after two or more passes are completed.

To view the maximum magnitude of delta S between passes:

- In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

  The **Solution Data** window appears. The **Convergence** tab is selected.

  The **Max. Mag. Delta S** column lists the maximum magnitude of delta S from one pass to the

next.

The **Max. Mag. Delta S** area lists the target change in magnitude of delta S and the change in magnitude of delta S between the last two solved passes.

Note Delta S is computed on the appropriate S-parameters - modal or terminal - after the S-parameters have been de-embedded and renormalized.

Note You can renormalize mathematically, without having to re-solve, by accessing the postprocessing tab on the port definition panel and de-selecting the **Deembed** selection box.

Related Topics

Setting the Maximum Delta S Per Pass

Technical Notes: Maximum Delta S

Viewing the Output Variable Convergence

At any time during or after the solution process, you can view the real and imaginary values of the output variable.

To view the output variable convergence, use the Reporter to create a plot that displays the output variable values.

Related Topics

Specifying Expressions for Adaptive Convergence

Viewing the Delta Magnitude Energy

For designs with voltage sources, current sources, or incident waves. Not applicable to designs with ports.

At any time during or after the solution process, you can view the difference in the relative energy error from one adaptive pass to the next. The change in the magnitude of delta energy is available after two or more passes are completed.

To view the delta magnitude E between passes:

- In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

  The **Solution Data** window appears. The **Convergence** tab is selected.

  The **Delta Mag. Energy** column lists the delta energy from one pass to the next.

  The **Delta Mag. Energy** area lists the target change in delta energy and the change in delta Energy between the last two solved passes.

Related Topics

Setting the Maximum Delta Energy Per Pass

Technical Notes: Maximum Delta Energy

Viewing the Magnitude Margin

For solutions in which convergence criteria for specific S-matrix entries were specified.

At any time during or after the solution process, you can view the solution's proximity to the target delta magnitude, which was specified in the **Matrix Convergence** dialog box. The magnitude margin is available after two or more passes are completed.

To view the magnitude margin between passes:

- In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

  The **Solution Data** window appears. The **Convergence** tab is selected.

  The **Magnitude Margin** column lists the magnitude margin from one pass to the next.

Related Topics

Setting Matrix Convergence Criteria

Technical Notes: Magnitude Margin

Viewing the Phase Margin

For solutions in which convergence criteria for specific S-matrix entries were specified.

At any time during or after the solution process, you can view the solution's proximity to the target delta phase, which was specified in the **Matrix Convergence** dialog box. The phase margin is available after two or more passes are completed.

To view the phase margin between passes:

- In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

  The **Solution Data** window appears. The **Convergence** tab is selected.

  The **Phase Margin** column lists the phase margin from one pass to the next.

Note When the Mag S becomes small (near to zero) its phase becomes indefinite and insignificant due to mathematical issues so that Phase Margin will be discarded.

Related Topics

Setting Matrix Convergence Criteria

Technical Notes: Phase Margin

Viewing the Max Delta (Mag S)

For solutions in which convergence criteria for specific S-matrix entries were specified.

At any time during or after the solution process, you can view the maximum difference of the S matrix magnitudes between two consecutive passes. The Max Delta (Mag S) is available after two or more passes are completed.

To view the Mag S between passes:

- In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

  The **Solution Data** window appears. The **Convergence** tab is selected.

  The **Max Delta (Mag S)** column lists the Max Delta (Mag S) from one pass to the next.

Related Topics

Setting Matrix Convergence Criteria

Technical Notes: Max Delta (Mag S)

Viewing the Max Delta (Phase S)

For solutions in which convergence criteria for specific S-matrix entries were specified.

At any time during or after the solution process, you can view the maximum difference of the S Matrix phase between two consecutive passes. The Max Delta (Phase S) is available after two or more passes are completed.

To view the Max Delta (Phase S) between passes:

- In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

  The **Solution Data** window appears. The **Convergence** tab is selected.

  The **Max Delta (Phase S)** column lists the Max Delta (Phase S) from one pass to the next.

Related Topics

Setting Matrix Convergence Criteria

Technical Notes: Max Delta (Phase S)

Viewing the Maximum Delta Frequency

For Eigenmode solutions.

At any time during the solution process, you can view the maximum delta frequency, the largest percent difference in the resonant frequencies from one adaptive pass to the next. It is a measure of the stability of the computed frequencies from pass to pass and is available after two or more passes are completed.

To view the maximum delta frequency between passes:

- In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

  The **Solution Data** window appears. The **Convergence** tab is selected.

  The **Max Delta Freq. %** column lists the maximum delta frequency from one pass to the next.

  The **Max Delta Freq. %** area lists the target maximum delta frequency and the maximum delta frequency between the last two solved passes.

Related Topics

Technical Notes: Maximum Delta Frequency

Plotting Convergence Data

To display convergence data vs. pass on a rectangular (x - y) plot:

1. In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

   The **Solution Data** window appears. The **Convergence** tab is selected.
2. In the lower-left corner of the window, select **Plot** as the view type.
3. Select the data you want to plot on the x-axis from the **X** pull-down list.
4. Select the data type you want to plot on the y-axis from the **Y** pull-down list.

   The x -y plot appears in the view window.

Viewing a Solution Profile

At any time during or after the solution process, you can examine the computing resources - or profile data - that were used by HFSS during the analysis. The profile data is essentially a log of the tasks performed by HFSS during the solution. The log indicates the length of time each task took and how much physical memory/disk memory was required.

In the project tree, right-click the solution setup of interest, and then click **Profile** on the shortcut menu.

The **Solution Data** dialog box appears. The **Profile** tab is selected. The displayed data depends on the type of problem and solution setup. If one or more dependent setups exist, the profile information for these can be selected from drop down menu in the Simulation text field at the top of the dialog. In general, it includes the following information:

Task	Lists the type of task that was performed.
Real Time	The difference in time between the start of the task and the end of the task (elapsed time).
CPU Time	The amount of CPU time required to perform the task.
Memory	The peak amount of physical memory (RAM) used by the individual executable running the task. The memory is freed for other uses after each task is complete.
Information	General information about the solution, for example, the number of tetrahedra used in the mesh.

The matrix solver writes specific information in some of these fields as outlined below:

Task The matrix solver task reports the type of solution performed by the solver, based on the physics of the problem. It is always of the form "Solver pdsn" (e.g. Solver MRS2), where

- p, the precision type is: M (mixed) or D (double)
- d, the matrix data type is: R (real) or C (complex)
- s, the symmetry type is: S (symmetric), A (asymmetric), or H (hermitian)
- n, the number of processors used. You specify the number of available processors on the local machine in the HFSS or HFSS-IE options. If a solve does not use all available processors (local or distributed), the number reported may be less than the number available.

Information: The matrix solver information line includes three sets of information (for example, 960885 matrix, 3030MB disk offcore)

- # matrix: The size of the matrix that was solved (the number of unknowns)
- # disk: The amount of hard disk space used during the calculation of the matrix solution
- "offcore": After the disk information, the word "offcore" may appear. This means that the solver could not place all of the data it needs to calculate the matrix solution in physical memory. If this word does not appear, then the solver was able to fit the necessary data into physical memory (known as "in-core"). If the matrix solver must solve off-core, smaller blocks of the data to be solved are created on disk, each block is then solved in physical memory, and then the matrix solution is reassembled. As a result of this additional processing, the time required to calculate a solution is higher.

To Export the Profile data:

1. Open the **Solution Data** dialog with the **Profile** tab selected.
2. Click the **Export Profile** button.

   This opens a file save dialog that lets you provide a file name and location.
3. Click **Save**.

   The data is saved in a text file with a .prof extension.

Related Topics

Viewing an Optimetrics Solution's Profile Data

Viewing Matrix Data

To view matrices computed for the S-parameters, impedances, and propagation constants during each adaptive, non-adaptive, or sweep solution:

1. In the project tree, right-click the solution setup of interest, and then click **Matrix Data** on the shortcut menu.

   The **Solution Data** dialog box appears. The **Matrix Data** tab is selected.

2. In the **Design Variation** text box, specify the design with the matrices you want to view.

   Optionally, choose a design variation solved during an Optimetrics analysis from the **Set Design Variation** dialog box. This lists all the solved variations in the design. This dialog box is accessible from the **Solution Data** window by clicking the ellipsis button on the right of the Design Variation field, and via the **>HFSS** or **HFSS-IE>Results>Apply Solved Variation** command.

3. In the **Simulation** pull-down list, click the solution setup and solved pass - adaptive, single frequency solution, or frequency sweep - for which you want to view matrices.

4. Select the type of matrix you want to view: **S-matrix**, **Y-matrix**, **Z-matrix**, **Gamma**, or **Zo** (characteristic impedance.) The available types depend on the solution type.

5. Select the display format — **Magnitude/ Phase (deg)**, **Real/ Imaginary**, **dB/ Phase (deg)**, **Magnitude**, **Phase (deg)**, **Real**, **Imaginary**, or **dB** — in which to display the matrix information.

6. Select the solved frequencies to display:

   - To display the matrix entries for all solved frequencies, select **Display All Freqs**.

     If **Display All Freqs** is enabled, the drop-down frequency select list is disabled.

   - To show the matrix entries for a selected solved frequency, ensure that **Display All Freqs** unchecked and use the dropdown list to select the solved frequency for which you want to view matrix entries. You can use a scroll bar for selecting from long frequency lists.

     For adaptive passes, only the solution frequency specified in the **Solution Setup** dialog box is available. For frequency sweeps, the entire frequency range is available.

   - To insert or delete one or more displayed frequencies, click **Edit Freqs**.

     Note: This command is only available if the sweep type is Fast or Interpolating.

     Clicking Edit Freqs displays the **Edit Sweep** dialog. It contains Generate New Values fields for specifying the Start Value, the End Value, and the Number of Values. The current values are displayed in a table. When you specify a New value, click **Update Values** to refresh the table.

     Note: Changes to the Start Value and End Value cannot be outside of the initial range. No message is issued: rather the range is implicitly restricted.

     Use the **Insert** button to add a new frequency to the table above the currently selected value. If no value, or the start value is selected, the new frequency repeats the current Start value and increments the count in the Number of Values field. If you select any other value for the insertion point, Insert adds a new value halfway between the selected value

and the previous value, and increments the Number of Values field.

Incrementing or decrementing the **Number of Values** fields, and the clicking **Update Values** updates the table based in the current Start and End value fields (given the range restriction within the initial range).

The **Delete** button enabled only if a value is selected. **Delete** removes the selected value and decrements the Number of Values field.

Click **OK** to apply the changes to the **Solutions** dialog **Matrix Data** tab and close the **Edit Sweep** dialog, or **Cancel** to close the dialog without applying the changes.

If you choose to export the matrix data for the Fast or Interpolating sweep after modifying the frequencies in the **Edit Sweep** dialog box, only those frequencies displayed under the **Matrix Data** tab will be exported.

The data is displayed in the table. By default, Waveports are listed in alphabetical, then numerical order, just as they appear in the excitation tree. To change the port order, change setting for Default Matrix sort order in the HFSS or HFSS-IE General options. You may also want see how you can Reorder Matrix Data.

7. Optionally, **Check Passivity**.

   This passivity check tests whether the S-parameter data from HFSS is passive or not. If the S-Matrix is not passive at one or more frequencies, this check displays a dialog that identifies the worst frequency violation and identifies the passivity in that case.

Related Topics

Selecting the Matrix Display Format

Viewing Solution Data for an Optimetrics Design Variation

Exporting Matrix Data

Renaming Matrix Data

Reordering Matrix Data

Exporting Equivalent Circuit Data

Technical Notes: Passivity

Selecting the Matrix Display Format

Use either the drop down menu under the formats, or you can also right-click on the column headings to display a pop-up format menu. Selecting from this menu also adds the name of the selected format to the column head, if the format is anything other than dB.

The available formats depend on the matrix type being displayed. When selected, dB formatting only applies to S -matrix data, even if other matrix types are displayed. The column heads in the display identify the format for the matrix type.You can display matrix data in the following formats.

Magnitude, Phase (deg)	Displays the magnitude and phase (in degrees) of the matrix type.
Real, Imaginary	Displays the real and imaginary parts of the matrix type.
dB, Phase (deg)	Displays the magnitude in decibels and phase in degrees of the matrix type.
Magnitude	Displays the magnitude of the matrix type.
Phase (deg)	Displays the phase in degrees of the matrix type.
Real	Displays the real parts of the matrix type.
Imaginary	Displays the imaginary parts of the matrix type.
dB	Displays the magnitude in decibels of the matrix type.

Related Topics

Renaming Matrix Data

Reordering Matrix Data

Exporting Matrix Data

1. In the project tree, right-click the solution setup of interest, and then click **Matrix Data** on the shortcut menu.

   The **Solution Data** window appears. The **Matrix Data** tab is selected.
2. Select the type of matrix you want to view: **S-matrix**, **Y-matrix**, **Z-matrix**, **Gamma**, or **Zo** (characteristic impedance.)
3. Click **Export Matrix Data**.

   A file browser appears.
4. Type the name of the file you are exporting to in the **File name** text box.

5. Select one of the following file formats from the **Save as type** pull-down list:

Format	Type	Description
(spreadsheet) *.tab	data table	A text file in which the elements of the S-matrix are arranged in a series of columns that are tab-separated and include a first row of headings. The file may be imported into a spreadsheet or similar utility.
***.sNp**	Touchstone/Libra	A Touchstone S-parameter file in which the number of ports is indicated by *n*. For example, a Touchstone file with one port would have the file extension .s1p. When you export this format, you are presented with a dialog where your can specify: • **Number of Digits Precision** (Default 15) • **Override Solution Renormalization**. This refers to overriding the renormalization impedances that may have been asked for in the port setup. • if so, the export renormalizing impedance • **Do Not Override Solution Renormalization.** This leaves any port setup renormalization options in place, and greys out the Impedance setting on the dialog. • whether to include **Gamma and Impedance Comments** If you want to export raw S-Parameter data for later use, you may choose to not renormalize the solution. If all ports and associated modes/terminals are normalized to the same impedance and you choose **Do Not Override Solution Renormalization** during export, the Touchstone file header will indicate the normalized impedance. The comment header in the Touchstone file lists the port and mode numbers to show which column contains which port name (in case of confusion between alphabetical and force repriority ordering of ports and associated modes).

***.nmf**	Neutral file format	Neutral file format defined by the MAFET Consortium. When you save to this format, if the design includes variables, you are presented with the NMF Parameters dialog that lets you select which variables to select as parameters in the NMF file. Non-selected variables will be give the constant value shown. You are then presented a dialog that lets you specify the number of digits precision for the renormalizing impedance.
***.m**	MATLAB	The Mathworks' MATLAB file in which the elements of the S-, Y-, or Z-matrix are arranged in a series of rows.
***.cit**	Citifile	Common Instrumentation Transfer and Interchange file format. It is an ASCII format defined by instrument and CAE designers.

6. For Touchstone files, you see a **Combine Sweeps** option on the **Export Network Data** solution dialog. This lets you combine sweeps into a single output file if:
   - You can combine any combination of discrete, interpolating, and fast sweeps, as long as the ranges don't overlap except possibly at the endpoints. (Note that the interpolating sweeps incorporate pre-solved data.)
   - The files must not have overlaps in the frequencies. (They can meet at a single frequency. For example, you can combine sweeps from 8 to 10 GHz with sweeps from 10 to 12 GHz, and 8-10 GHz and 11-13 GHz, but not sweeps from 9 to 11 GHz and 10 to 11 GHz.)

   a. Select the **Combine Sweeps** button to display a **Combine Interpolating Sweeps For Export** dialog with a list of sweeps.

   b. Select the sweeps to combine and click **Combine**.

      This closes the **Combine Interpolating Sweeps for Export** dialog.

7. Click **Save**.

   The data is exported to the file.

   - By default, wave ports are listed in alphabetical, then numerical order, just as they appear in the excitation tree. You can change this order to creation order and back without invalidating the solution on the HFSS Options dialog.
   - If you select Touchstone format, you are first presented with a dialog in which you can specify the Number of Digits precision, and Override the Solution Renormalization. If so, you co specify the export renormalizing impedance (an integer value). Here you also can specify and whether to include **Gamma and Impedance Comments**, and **Number of Digits Precision** (Default 15).
   - If you select Neutral File Format, you are presented with a **Specify Export Renormalizing Impedance** dialog that lets you specify the Number of digits precision for the save

file.

Note If you modify the display of solved frequencies in an Interpolating or Fast sweep under the **Matrix Data** tab (by clicking **Edit Freqs** and then modifying the values in the **Edit Sweep** dialog box,) only those frequencies listed will be exported to the file.

Renaming Matrix Data

In the project tree, you can right-click on a port excitation to rename it. When you rename a port excitation, the associated data is reordered so that it can be presented in the same manner. The reordering is done to match the tree-sort order presented for the ports (renamed matrix data is reordered so that alphabetic values appear before numeric values).

Exports of the matrix data are ordered in the same manner. This reordering is conducted as part of post processing and does not force a re-solve.

Related Topics

Reordering Matrix Data

HFSS: General Options

Viewing Matrix Data

Reordering Matrix Data

HFSS lets you reorder the matrix data as a post-processing step.

1. To re-order the matrix data, either as the default ascending alphanumeric order, or a user specified order, click **HFSS>Excitations>Reorder Matrix**, or right-click on Excitations in the Project tree, and click **Reorder Matrix** on the shortcut menu.

   This displays the **Reorder Matrix** dialog. It lists the ports for a modal solution, or the terminals for a terminal solution. For Transient solutions, the dialog displays separate lists for Active and Passive ports.

2. You can select the radio button for Sort in ascending alphanumeric order, or Sort in the below order.

   If you select Sort in the below order, you can select any port or terminal to enable the up and down arrow keys. Use the keys to move the ports into any desired order. In the case of Transient, the arrow keys operate only within a partition, that is only within Active or Passive lists.

   You can also use CTRL-click to select multiple arbitrary ports, or hold Shift to select a range of ports. Clicking outside the list deselects all selections.

Note If there are differential pairs, the sort order is still specified) in terms of the underlying terminal names, but the entries that make up the pair should appear in the appropriate sort location for the terminals that are used to define them.

3. Click OK to accept the order specified.

   For Transient solutions, removing an active source will not affect the solve but might affect the matrix order. Adding an active source will require a resolve if that source has not already been solved.

Related Topics

HFSS: General Options

Viewing Matrix Data

Renaming Matrix Data

Exporting Equivalent Circuit Data

You can export S-parameter data from a Driven Terminal solution to PSpice, HSPICE, Spectre or Maxwell Spice format. Importing the new data file to PSpice, HSPICE, Spectre or Maxwell Spice will enable you to include wave effects in the circuit simulations. You can also export a W-Element model for a port.

Note

You must have a frequency sweep solution and five or more frequency points to successfully export an equivalent circuit data file. See the *Choosing Frequencies for Full-Wave SPICE* topic of the online help for suggestions about the frequency range of the sweep.

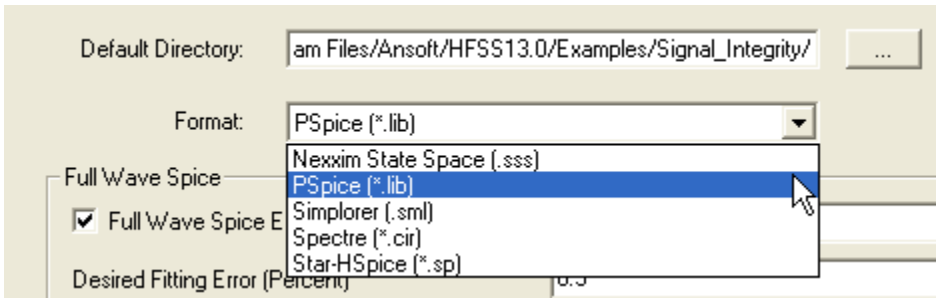
The GUI lets you export full-wave Spice for a model that contains differential pairs, but it will silently export the data in its original single-ended form. The full-wave Spice model is a "broadband" equivalent circuit (that is, its S-parameters match those of the HFSS solution across the whole frequency sweep range.)

Certain discrete sweeps permit Full-Wave SPICE exports. It is allowed if the discrete data is evenly spaced, includes DC, and has at least 500 frequency points.

1. In the project tree, right-click the solution setup of interest, and then click **Matrix Data** on the shortcut menu.

   The **Solution Data** window appears. The **Matrix Data** tab is selected.
2. Click **Equivalent Circuit Export**.

   The **Equivalent Circuit Export Options** dialog box appears.
3. Type the name or browse to the directory in which you want to store the data.



4. Click one of the following formats in the **Format** list:

   PSpice (*.lib)

   Nexxim State Spalce (.sss)

   Simplorer (.sml)

   Star HSpice (*.sp)

   Spectre (*.cir)

   Your format selection affects the options available under **Full Wave Spice Export**. When Simplorer format is requested, both *.sml and *.png are created. The latter contains the image in GIF format.

5. If the **Full-Wave Spice Export** checkbox is enabled, you can select it. Checking the box enables the text field for the file name, and depending on the format selection, other options may be enabled.

   - **Desired Fitting Error (percent)**
   - **Maximum Order**

   HFSS supports Full Wave Spice Export from a driven modal design as long as all ports have exactly one mode each. However, HFSS does not support definition of differential pairs in a driven modal design.

6. Optionally, select **Use Common Ground** to combine the negative reference nodes for all of the ports into a single reference node.

7. Optionally, select **Enforce Passivity**. Selecting this enforces passivity in the output file. Passive devices can only dissipate or temporarily store energy, but never generate it. (You can also check passivity from the **Matrix Data** tab using the Check Passivity button.)

   This option is useful in cases where the transient simulation fails due to passivity violations in the circuit model. This circuit model is based on fitting a rational function to the S-parameter data computed by the field solver. Small errors in the data fitting can result in non-passive behavior. Selecting the **Enforce Passivity** option will take more CPU time, but ensures that the resulting model will be passive.

   The passivity check tests whether the S-parameter data from HFSS is passive or not. For more information see Passivity.

8. Optionally, select **Lumped Element Export (Low Bandwidth)** if you want to save the data as a low-frequency circuit model using simple lumped elements (resistors, capacitors, inductors, and dependent current sources). The low-bandwidth model is only going to be accurate in a limited frequency range around the adaptive solution frequency

   This option is not enabled for Spectre export.

9. Optionally, select **Partial Fraction Expansion for Matlab** if you want to specify a file that expands the partial fractions for use in Matlab. The partial fractions involved describe the frequency response of the low-bandwidth model from the previous step.

10. You can also select **Combine Sweeps** to select and combine available sweeps into a single out-

put file.

By option you can Export W-Element Data.

11. Click **OK**.

    The S-matrices are written to the data file that you specified in the equivalent circuit data format.

    Related Topics

    Exporting W-Element Data

Exporting W-element Data

It is possible to extract a W-element model for a port. This W-element model can be used in a SPICE model to represent a length of transmission line of the same cross section as the port. A W-element model can be extracted for a port only solution and for a full 3D solution.

1. In the project tree, right-click the solution setup of interest, and then click **Matrix Data** on the shortcut menu.

   The **Solution Data** window appears. The **Matrix Data** tab is selected.

2. Click **Equivalent Circuit Export**.

   The **Equivalent Circuit Export Options** dialog box appears. At the bottom of the dialog you see the W-element model check box.

3. Click the W-element model check box to enable the W-element fields.
4. The W-element model name field has the project name by default. You can change this if desired.
5. Choose the format as Tabular Format (the default) or RGLC format for W-element export.

   Tabular Format: provides a unique RLGC model for each frequency in the solution.

   RLGC Format: provides a RLGC fit over a frequency range based on Ro, Lo, Go, Co, Rs and Gd parameters.

Note For the RLGC Format, if only a single frequency solution is selected (e.g. LastAdaptive) then Rs and Gd parameters are ignored.

6. In the **Model name** field, provide a model name.
7. Either select the port from the **Port Name** pull down, or, to export a W-element model for all ports, select the **Export for All Ports** radio button.
8. Click **OK**.

   The W-element model is written to the data file that you specified.

   Related Topics

   Technical Notes: Calculating the W-Elements

Viewing Mesh Statistics

To view an adaptive solution's mesh information, either during or after the solution process:

1. In the Project tree, right-click the solution setup of interest, and then click **Mesh Statistics** on the shortcut menu.

   The **Solutions** dialog box appears with the Mesh Statistics tab selected.

   For HFSS projects, the table lists the design elements and for each includes: Num Elements, Min edge length, Max edge length, RMS edge length, min tet vol., max tet vol., mean tet vol. and standard deviation.

   For projects with HFSS-IE regions, the Mesh data table is similar but includes min, max, and mean element area information for triangles, as well as instead of tet volume information for solve-inside portions of the project.

   If mesh repairs have been performed, two additional columns appear in the table; Recovered %, Repaired %. These columns indicate the fraction of an object that was successfully recovered and the fraction that needed some repair.

To toggle the mesh statistics display from low to high values or visa versa:

1. Click on the column header.

   This displays a shadowed triangle pointing down to indicate a list ordered from highest to lowest, and a triangle pointing up to indicate a list ordered from lowest to highest. Clicking again inverts the current order.

   Click on blank cell above the object list to invert the order of objects, though in this case, the cell does not display a directional triangle.

   Related topics

   Technical Notes: The Finite Element Method

   Technical Notes: The Mesh Generation Process

Viewing Eigenmode Solution Data

To view the real and imaginary parts of the frequency and quality factor Q computed for each eigenmode:

1. In the project tree, right-click the solution setup of interest, and then click **Eigenmode Data** on the shortcut menu.

   The **Solution Data** window appears. The **Eigenmode Data** tab is selected.

2. In the **Simulation** pull-down list, select the solution setup and solved pass - adaptive or single frequency solution - for which you want to view data.

   The solved eigenmodes are listed in the table below.

   The **Frequency** column lists the real and (for lossy materials) imaginary parts of the frequency (or resonant frequency) for each solved eigenmode.The display uses the *re* + j *im* format.

   For lossy Eigenmode solutions, a **Q** column appears, which lists the unloaded quality factor Q computed for each eigenmode.

3. To export the Eigenmode solutions to a text file with an eig extension, click the **Export** button.

This displays **Save As** dialog. You can provide a file name, and if desired, change to a non-default location. Click the **Save** button to save the text file and close the Save As dialog.

Related Topics

Technical Notes: Eigenmode Solutions

Technical Notes: Calculating the Resonant Frequency

Technical Notes: Calculating the Quality Factor

Technical Notes: Calculating the Free Space Wave Number

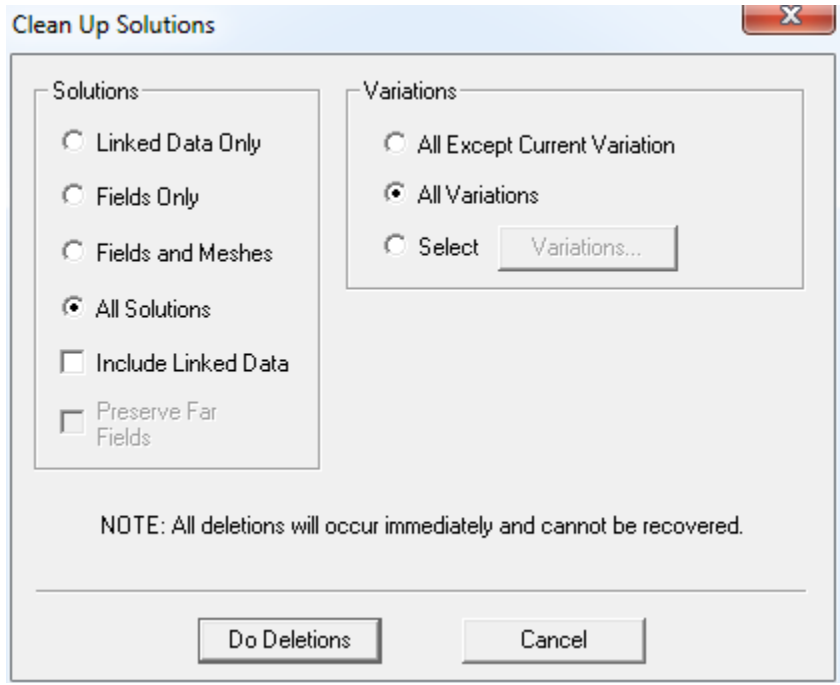
Deleting Solution Data

You can use **Clean Up Solutions** to selectively make deletions, or remove all solutions from the results.

To use **Clean Up Solutions**:

1. Click **HFSS**>**Results**>**Clean Up Solutions**.

   The **Clean Up Solutions** dialog box appears.



2. Under **Solutions**, select whether you want to delete only fields data, only fields and mesh data, only linked data, or all solution data. Deleting all solution data erases all mesh, matrix, and fields data for all adaptive passes and frequency sweeps for the selected **Variations**.

   By option, you can include linked data in the deletions.

   Linked data can be mesh, field or some other post-processing data that the source design generated. The target design for the link caches these data internally to minimize the need to acti-

vate the source design.

3. Under **Variations**, select which solution data you want to delete:
   - Select **All Except Current Variation** to delete all solution data that do not correspond to the current project and design variable values for the current design.
   - Select **All Variations** to delete all solution data for the current design.
   - Select **Select** to specify the variations you wish to delete. Click **Variations** to select the variations for deletion.
4. Click **Do Deletions**.

   The solution data you selected are deleted. Any post processing reports or field overlays you created that included data you deleted will be marked with an X in the project tree. They will be invalid until new solution data are generated.

Related Topics

Monitoring the Solution Process

Deleting Reports

Deleting Reports

To use **Delete All Reports**:

1. Click **HFSS**>**Results**>**Delete All Reports.** You can also rightclick on the Results folder in the Project tree to display the shortcut menu, and click **Delete All reports.**

   All items under the Results folder in the Project tree are removed.

To use **Delete** for a selected report:

1. Select a report icon in the Project tree, and right-click to display the shortcut menu.
2. Click Delete on the shortcut menu or the "X" icon on the toolbar to delete the selected report.

Warning Solution data that have been deleted cannot be recovered!

Related Topics

Clear Linked Data

Deleting Solution Data

Scaling a Source's Magnitude and Phase

You can scale the magnitude and set the phase of ports, voltage and current sources, Eigenmodes, and incident waves in the **Edit post process sources** dialog box. For HFSS modal projects, and HFSS-IE, you specify the total voltages. For HFSS terminal projects, you can also select either Incident Voltage or Total Voltage. The incident voltage selection includes magnitude, phase, and the impedance of a hypothetical external line. For HFSS Transient Network projects, the **Edit Sources** dialog includes tabs that let you specify scales for Spectral and Transient data separately. For HFSS Transient, **Edit Sources** is disabled.

For Driven Modal and Terminal Analysis solution types that involve near or far fields, a **Source Contexts** tab appears on the **Edit post process sources** dialog. This provides a means to select the sources to use as context when creating a radiated field report. The tab lists the same sources as the Spectral Fields tab. Sources are scaled by exciting a single source (the selected context in the reporter dialog) while turning off all other sources.

You can also **Save to File** and **Load from File**, where the file is comma delimited data (.csv) format or tab delimited data (.txt). This can help in defining source values for projects with a large number of sources. An example of the .csv format for HFSS follows:

```
Name,Magnitude,Phase
Port1:1,1W,0deg
Port2:1,0W,0deg
Port3:1,0W,0deg
```

An example of the tab delimited format follows:

```
Name      Magnitude   Phase
Port1:1   1W          0deg
Port2:1   0W          0deg
Port3:1   0W          0deg
```

Scaling Sources for HFSS or HFSS-IE

Scaling Sources for HFSS Transient Network

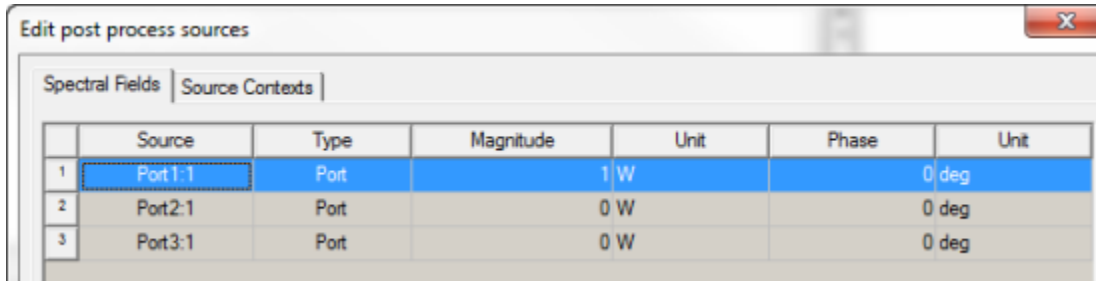
Specifying Source Contexts for Creating Radiated Field Reports

Scaling Sources for HFSS or HFSS-IE

1. Click **HFSS** or **HFSS-IE**>**Fields**>**Edit Sources**.

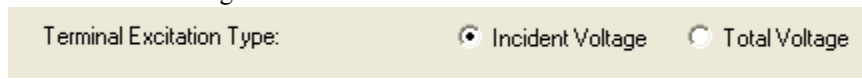
   The **Edit post process sources** dialog box appears For modal projects, the Spectral fields tab

shows the following column headings.

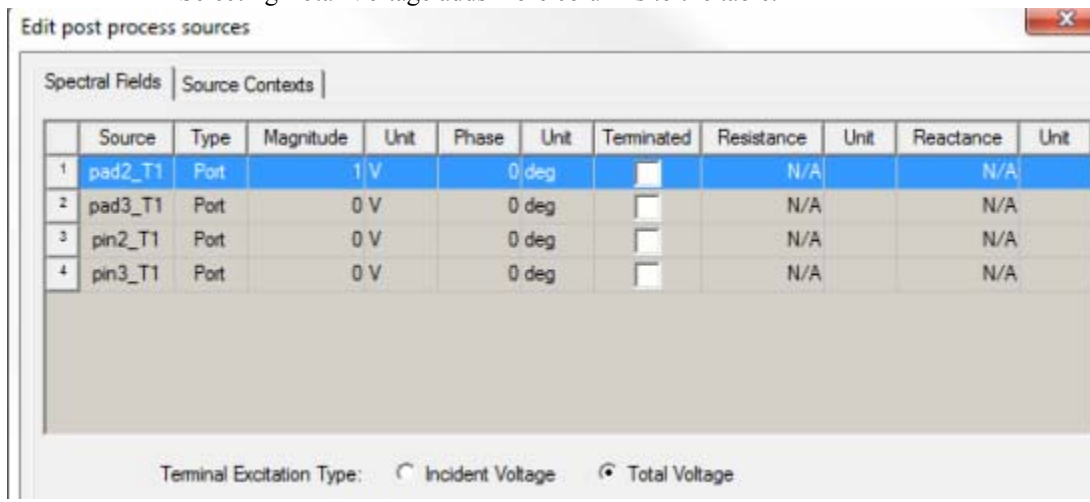


If the project contains symmetry planes, the dialog includes a reminder that you may need to adjust the scaling factor accordingly.

For HFSS Terminal solutions, you can select a Terminal Excitation Type as Incident Voltage or Total Voltage.



Selecting Total Voltage adds more columns to the table.



Note that in the **modal** case a unit stimulation means 1 Watt of incident power at the port; in the **terminal** case a unit stimulation means 1 volt of total voltage at the terminal. After converting the voltage stimulation to the equivalent power stimulation the antenna results agree perfectly. In particular, the "ratioed" antenna parameters such as gain, directivity, and efficiency agree between the modal and terminal projects, while absolute antenna quantities such as incident power, accepted power may initially appear different. This is a direct result of the difference in edit-sources stimulations in the two types of projects.

You can scale the sources individually through the source table in the **Edit post processing sources** dialog, or you can **Save to File** or **Load From File**, using the .csv format or tab delimited data (.txt) format. This feature can help for projects with many sources. The steps below

describe how to directly edit the source table.

2. Select the source whose magnitude and phase you want to scale.
3. In the **Magnitude** text box, enter the magnitude you want. Design variables can be used as source scalings.

Note You may not enter a negative voltage. To obtain the equivalent of a negative magnitude, add or subtract 180 degrees from the phase value.

If you use a design variable as a scaling factor note that solutions are invalidated if the variable is changed.

If the model contains symmetry planes, the **Edit Sources** dialog alerts you that you may need to adjust the scaling factor accordingly.

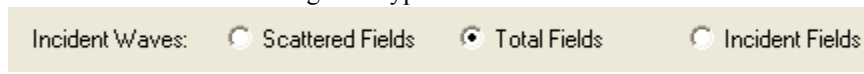
At least one source should be excited (non-zero). If you set all sources to zero, you will receive a warning, but the values do go through.

4. In the **Phase** text box, enter the new phase for the source.

   The phase of the source is changed by the value that you enter.
5. Optionally, if your solution type is driven terminal, you may specify a complex reference impedance:
   a. For the selected terminal, select **Terminated**.

      This disables the values to the left of the checkbox, and enables the Resistance and Reactance text boxes. Use the scroll bar to view them.
   b. Enter the **Resistance** and the **Reactance** and select the units. Ohms is the default.
6. By option, you can click a checkbox to **Include Post Processing Effects**.

   Checking this box also enables an **Apply** button.
7. For HFSS, if an incident wave is present, use the radio buttons at the bottom of the panel to select one of the following field types to use:



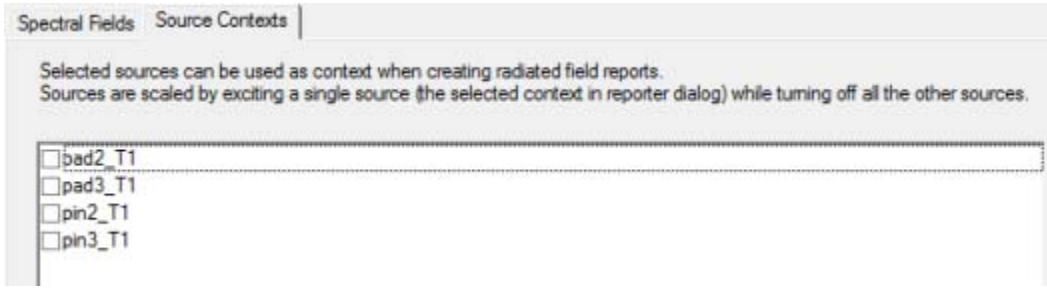
Scattered Fields	The differential field formed by subtracting the incident field from the total field.
Total Fields	The physically measurable field that exists with the model present and a non-zero incident field.
Incident Fields	The plane-wave field that would exist in the absence of the model.

For HFSS-IE, these options do not appear in the **Edit post processing sources** dialog. The Port Processing effects are always on, and the entered voltages in **Edit post processing sources** are always in Total Voltage form.

For Eigenmode solutions, the **Edit post process sources dialog** displays a Spectral Fields tab and radio buttons for selecting as the Eigenmode Excitation Type, either Peak Electric Field or Stored Energy.

For Stored Energy, the table lists Magnitude and Unit as editable fields. For Peak Electric field, the editable fields are Magnitude, Unit, Phase, and Unit.

8. The Source Contexts tab lets you select the sources to use as context when you create radiated fields reports.



9. Click **OK** to apply the changes and close the dialog, or click **Apply** to view the changes without closing the dialog.

   The magnitude and phase are assigned to the selected excitation.

Note When you scale an excitation, keep in mind that the original value of the excitation remains unchanged.

Related Topics

Guidelines for Scaling a Source's Magnitude and Phase

Guidelines for Scaling a Source's Magnitude and Phase

When specifying the magnitude of a source keep the following guidelines in mind:

For ports, driven modal case	• The excitation's magnitude specifies time-averaged incident power in watts. • If you are using a symmetry plane, remember to scale the input signal appropriately. For example, if you have one symmetry plane, use an input value of 0.5 watts to excite the full structure with 1 watt; if you have two symmetry planes, use an input value of 0.25 watts to excite the full structure with 1 watt, and so forth. • Generally, use the default value of **1**. This specifies that the solution's E- and H-fields be scaled such that the excitation wave delivers 1 watt of power. To view the solution at some other power, enter a positive value. • Only port-mode combinations with non-zero magnitudes will be used.
For voltage and current sources	• The source magnitude for voltage and current sources specifies peak value volts and peak value amperes, respectively. • If you have defined multiple voltage and current sources, you can "remove" them by setting their magnitudes to **0**. This enables you to easily observe the effects that individual or specific groups of sources have on the problem.
For incident waves	• Source magnitude specifies peak value E-field in volts per meter. • When you scale the incident E-field, the scattered E-field and the total E-field are scaled as well. • This scaling factor affects all incident angles in the incident wave setup.
For ports, driven terminal case	• The excitation's magnitude specifies peak value volts. This is the sum of the incident and reflected waves at this terminal. See the equations here.
For Eigenmodes	• Source magnitude is unitless and represents a relative value. • When you enter a scaling factor for an eigenmode the relative source magnitude is amplified by this value. Exactly one eigenmode must be excited by setting its scaling factor to a non-zero positive number.
Ports of Transient Network solutions	• See Ports and Edit Sources Behavior for Transient Network

When specifying the new phase for ports, generally use zero. This zero-phase solution results from excitations phased in such a way that, at $\omega t = 0$, peak values occur at the port faces.

Scaling Sources and Setting Delays for HFSS Transient Network

For HFSS Transient Network solutions you can separately edit sources for Spectral Fields and Transient Fields. The **Transient** tab on **Edit Sources** controls transient far fields and fields saved on the mesh. The **Spectral Fields** tab on **Edit Sources** controls Frequency domain far fields and antenna parameters.

- Editing Sources for Spectral Fields for Transient Network

- Editing Sources for Transient Fields for Transient Network

When non-port sources exist, the port post processing effects are off. For both Spectral and Transient fields, you can scale the sources individually through the source table in the **Edit Sources** dialog, or you can **Save to File** or **Load From File**, using the .csv format or tab delimited data (.txt) format. This feature can help for projects with many sources. The linked steps describe how to directly edit the source table.

In Transient Network analysis each source is solved separately which allows arbitrary superposition of the results from all sources controlled by **Edit Sources** settings. This feature enables users great flexibility in studying a large set of excitation variations in an efficient and convenient manner.

Note When superposing results it is assumed that the fields have decayed to zero at the end of the simulation for each source. In practice, the user settings for Target Residual and time duration can cause this assumption to be violated. In this case the displayed field quantities may contain significant error.

For example, consider a design with two sources where the user has specified a large Target Residual setting, and one source converges at 1ns while the other converges at 2ns. If the user had not specified any time shifts in **Edit Sources**, the displayed field quantities will not be reliable in the range 1ns to 2ns because the fields from the first source cannot be assumed to be zero.

Related Topics

Editing Sources for Spectral Fields for Transient Network

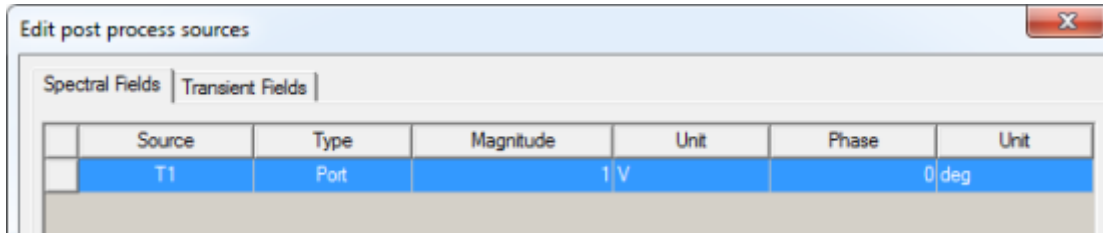
Editing Sources for Transient Fields for Transient Network

Editing Sources for Spectral Fields for Transient Network

For Spectral fields, you can optionally include port post-processing effects. The **Spectral Fields** tab on **Edit post process sources** controls Frequency domain far fields and antenna parameters. The voltage specifications are only for the incident voltage.

1. Click **HFSS**>**Fields**>**Edit Sources**.

   The **Edit post process sources** dialog box appears with the **Spectral Fields** tab selected.



   For HFSS Transient Network, the **Edit post process sources** dialog displays information for sources for Spectral Fields in table format. You can edit the Magnitude and Phase values including the use of variables, and set the units for each. You can scale the sources individually

through the source table in the **Edit Sources** dialog, or you can **Save to File** or **Load From File**, using the .csv format. This feature can help for projects with many sources. The steps below describe how to directly edit the source table.

Note

You may not enter a negative voltage. To obtain the equivalent of a negative magnitude, add or subtract 180 degrees from the phase value.

If you use a design variable as a magnitude note that solutions are invalidated if the variable is changed.

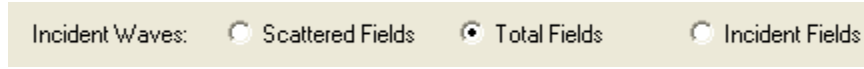
If the model contains symmetry planes, the **Edit Sources** dialog alerts you that you may need to adjust the scaling factor accordingly.

At least one source should be excited (non-zero). If you set all sources to zero, you will receive a warning, but the values do go through.

2. By option, you can click a checkbox to **Include Post Processing Effects**.

Checking this box also enables an **Apply** button.

3. For HFSS, if an incident wave is present, use the radio buttons at the bottom of the panel to select one of the following field types to use:



Scattered Fields	The differential field formed by subtracting the incident field from the total field.
Total Fields	The physically measurable field that exists with the model present and a non-zero incident field.
Incident Fields	The plane-wave field that would exist in the absence of the model.

Related Topics

Guidelines for Scaling a Source's Magnitude and Phase

Ports and Edit Sources Behavior for Transient Network

Scaling Sources for HFSS Transient Network

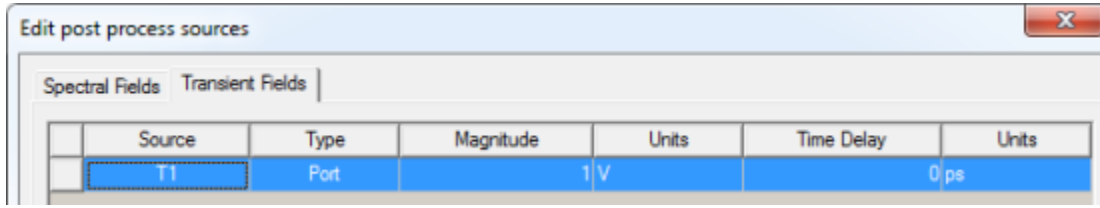
Editing Sources for Transient Fields for Transient Network

For Transient fields you can set magnitude and time delay for each source. The **Transient** tab on **Edit post process sources** controls transient far fields and fields saved on the mesh. The voltage specifications are only for the incident voltage.

1. Click **HFSS**>**Fields**>**Edit Sources**.

The **Edit post process sources** dialog box appears with the **Spectral Fields** tab selected.

2. Select the **Transient Fields** tab.



3. Edit the Magnitude and Time Delay values as desired, and select the Units for each.

   You can scale the sources individually through the source table in the **Edit Sources** dialog, or you can **Save to File** or **Load From File**, using the .csv format. This feature can help for projects with many sources. An example of the format for HFSS Transient follows:

```
Name,Transient Magnitude,Delay
bw_tr_1_T1,1V,0ps
bw_tr_2_T1,0V,0ps
sb_tr_1_T1,0V,0ps
sb_tr_2_T1,0V,0ps
```

   The magnitude and delay are assigned to the selected excitation.

Note When you change the Magnitude of an excitation, keep in mind that the original value of the excitation remains unchanged.

Related Topics

Guidelines for Scaling a Source's Magnitude and Phase

Ports and Edit Sources Behavior for Transient Network

Scaling Sources for HFSS Transient Network

Ports and Edit Sources Behavior for Transient Network

Transient Network solutions support the following port features:

- Just as in Driven Terminal we have two types of ports namely wave ports and lumped ports.
- Both port types are restricted to be single TEM terminal ports such as coax and gap ports.
- Waveports are matched loaded.
- Lumped ports are loaded with the user defined "Full Port Impedance".
- Both ports support renormalization of the spectral quantities S-parameters and far fields including antenna parameters.
- Neither port supports renormalization of transient quantities.

Edit Sources for Transient Network involves specifying the incident voltages at the terminals of the ports not the total voltages. This means that all ports are terminated with the port settings and if the edit source for a given port is set to zero it means that port is loaded with the port setting and it does

not imply a short. As mentioned above, transient fields are not affected by any renormalization but the frequency domain far fields and antenna parameters are affected by renormalization.

In Transient Network analysis each source is solved separately which allows arbitrary superposition of the results from all sources controlled by Edit Sources settings. This feature enables users great flexibility in studying a large set of excitation variations in an efficient and convenient manner.

Note

When superposing results it is assumed that the fields have decayed to zero at the end of the simulation for each source. In practice, the user settings for Target Residual and time duration can cause this assumption to be violated. In this case the displayed field quantities may contain significant error.

For example, consider a design with two sources where the user has specified a large Target Residual setting, and one source converges at 1ns while the other converges at 2ns. If the user had not specified any time shifts in Edit Sources, the displayed field quantities will not be reliable in the range 1ns to 2ns because the fields from the first source cannot be assumed to be zero.

For HFSS Transient, **Edit Sources** is disabled.

Related Topics

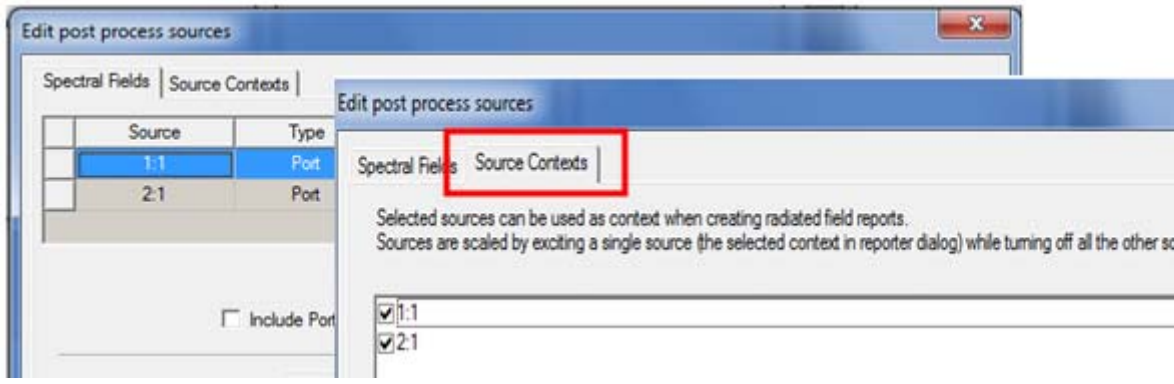
Editing Sources for Spectral Fields for Transient Network

Editing Sources for Transient Fields for Transient Network

Scaling Sources for HFSS Transient Network

Specifying Source Contexts for Creating Radiated Field Reports

For Driven Modal and Terminal Analysis solution types that involve near or far fields, a **Source Contexts** tab appears on the **Edit post process sources** dialog. This provides a means to select the sources to use as context when creating a radiated field report. The **Source Contexts** tab lists the same sources as the **Spectral Fields** tab. Sources are scaled by exciting a single source (the selected context in the reporter dialog) while turning off all other sources.



All types of sources can be enabled as Source context (ports, currents, incident waves). The default for all sources is "disabled". This applies to newly created sources and sources in legacy projects.

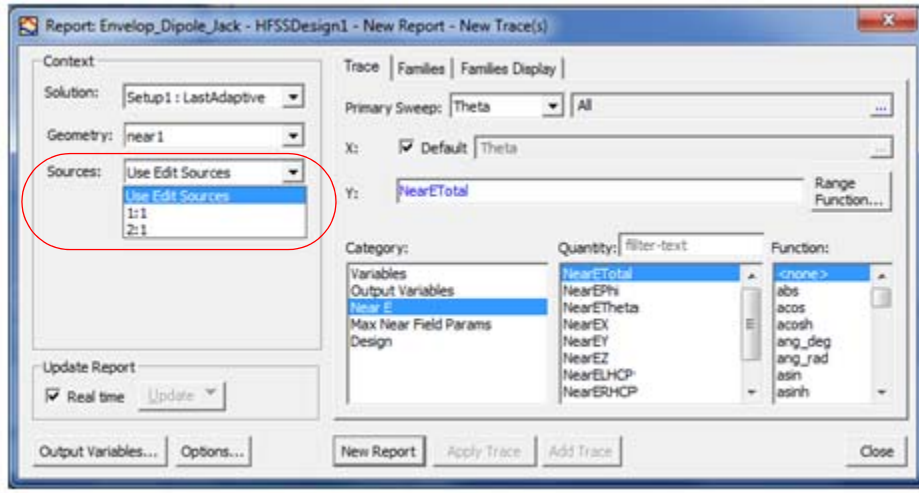
Each singly selected source is excited with factory default values. The phase will always be 0 deg and the magnitude will be 1 Watt for modal design and 1 Volt (Total voltage) for terminal design. These factory default values cannot be changed by users.

Design **Edit post process sources** settings such as "Include Port Post Processing Effects" are applicable for these singly excited sources. Thus changing any setups on the first tab of the **Edit post processing sources** dialog invalidates all traces that are defined with source context.

Enabling any source as context does not impact traces that are already defined

Disabling source as context invalidates an existing trace that uses that source as context.

This **Edit post process sources** selection applies to a Source selection in the Reporter that allows you to excite a single source.



The Reporter Context selection for "Sources" appears when there is at least one source selected for "Source Contexts". The default selection of this combo box is "Use Edit Sources". When a source is selected, the **Edit Sources** setup in the design will be ignored during trace population. Instead, the trace will be calculated with the selected source excited while turning off all the other sources in the design. The above figure illustrates the extended **Create Near Field Report** dialog. Note that "1" and "2" are port names and each has 1 mode.

For an example use of these features, see the User Defined Solution for MIMO Calculations example, that describes python scripts for the Toolkits and User Defined Solution features to generate calculations and reports.

Creating Animations

An animated plot is a series of frames that displays a field, mesh, or geometry at varying values. To create an animated plot, you specify the values of the plot that you want to include, just as an animator takes snapshots of individual drawings that make up a cartoon. Each value is a frame in the animation. You specify how many frames to include in the animation.

Note Each animation frame requires memory for storage which depends upon the mesh size and type of plot. Memory usage may become very large during plot animations. To reduce memory usage, specify the minimum number of frames possible. See General Options for more information.

Note If animation is slow, especially for complex models, for some graphics cards, you can improve performance by setting **NVIDIA Control Panel>3D Settings>Manage 3D Settings Global Settings>Global Presets: Workstation App - Dynamic Streaming**

You can export the animation to animated Graphics Interchange Format (GIF) or to Audio Video Interleave (AVI) format.

Related Topics

General Options: Miscellaneous Options Tab

Creating Phase Animations

Creating Frequency Animations

Creating Geometry Animations

Controlling the Animations Display

Exporting Animations

Creating Phase Animations

To animate a plot with respect to the phase of the plotted field:

1. Create a field overlay plot to animate.
2. Click **HFSS**>**Fields**>**Animate**  .

   If you already created an animation, the **Select Animation** dialog box appears. Selecting an existing animation from that list starts it. To create a new animation, click **New**.

   The **Setup Animation** dialog box appears.
3. Type a name for the animation in the **Name** text box or accept the default name.
4. Optionally, type a description of the animation in the **Description** text box.
5. Under the **Swept Variable** tab, select **Phase** from the **Swept Variable** list.
6. Specify the phase values you want to include in the animation:
   a. Type the starting value of the phase in the **Start** text box.
   b. Type the stopping value of the phase in the **Stop** text box.

c. Type the number of **Steps** to include in the animation.

For example, if the **Start** value is **10**, the **Stop** value is **160**, and the number of steps is **10**, the animation will display the plot at 10 phase values between 10 and 160. The start value will be the first frame displayed, resulting in a total of 11 frames in the animation.

d. If the design has multiple project or intrinsic variables, click the **Design Point** tab to set the values of the non-animated variables.

1. Click the **Design Point** tab.
2. Deselect the **Use defaults** checkbox.

In the table, select the row corresponding to the variable setting of interest.

3. Click **OK**.

The animation begins in the view window. The play panel appears in the upper-left corner of the desktop, enabling you to stop, restart, and control the speed and sequence of the frames.

Related Topics

Creating Animations

Controlling the Animation's Display

Creating Frequency Animations

1. Create a field overlay plot to animate.

In the **Create Field Plot** dialog box, make sure to select a sweep solution to plot from the **Solution** pull-down list.

2. Click **HFSS**>**Fields**>**Animate**  .

If you already created an animation, the **Select Animation** dialog box appears. Selecting an existing animation from that list starts it. To create a new animation, click **New**.

The **Setup Animation** dialog box appears.

3. Type a name for the animation in the **Name** text box or accept the default name.
4. Optionally, type a description of the animation in the **Description** text box.
5. Under the **Swept Variable** tab, select **Frequency** from the **Swept Variable** list.
6. Select the frequency values you want to include in the animation from the **Select values** list.

Use the **Shift** key to select a series of values, and the **Ctrl** key to select values that are not in sequence.

7. If the design has multiple project or intrinsic variables, click the **Design Point** tab to set the values of the non-animated variables.

a. Click the **Design Point** tab.

b. Deselect the **Use defaults** checkbox.

In the table, select the row corresponding to the variable setting of interest.

8. Click **OK**:

The animation begins in the view window. It will display one frame for each frequency value you selected.

The play panel appears in the upper-left corner of the desktop, enabling you to stop, restart, and control the speed and sequence of the frames.

Related Topics

Creating Animations

Controlling the Animation's Display

Creating Geometry Animations

Geometry animations may be created to evaluate the effect of varying geometry variables on the model. You must define at least one variable associated with the geometry prior to creating a geometry animation. Following is the general procedure for creating an animation that varies a part of the model geometry.

1. Right-click in the view window, point to **View**, and then click **Animate**.

   If multiple geometries can be varied in the design, the **Select Drawing** dialog box appears, proceed to step 2. If only one geometry is variable, proceed to step 3.

2. In the **Select Drawing** dialog box:
   a. Select the geometry variable to vary in the animation.
   b. Select the object you want to animate.

Note If previous animations have been created for this project, the **Select Animation** dialog will appear. You may choose an animation setup from the list if one is associated with the geometry variable of interest and the animation will start. If no existing animation setup is acceptable, select **New** and continue at Step 3 below.

   The **Setup Animation** dialog box appears.

3. In the **Setup Animation** dialog box:
   a. Type a name for the animation in the **Name** text box or accept the default name.
   b. Optionally, type a description of the animation in the **Description** text box.
   c. Under the **Swept Variable** tab, the **Swept Variable** list includes all of the defined geometric project and design variables. Select the geometry variable that you want to animate from the **Swept Variable** list.
   d. Specify the values of the variable that you want to include in the animation:
      1. Type the starting value of the variable in the **Start** text box.
      2. Type the stopping value of the variable in the **Stop** text box.
      3. Type the number of **Steps** to include in the animation.
         For example, if the **Start** value is **0.15in**, the **Stop** value is **0.45in**, and the number of steps is **15**, the animation will display the geometry at 15 values between 0.15 inches and 0.45 inches. The animation will also include the start value, which will be the first frame displayed, resulting in a total of 16 frames in the animation.
   e. If the design has multiple project or intrinsic variables, click the **Design Point** tab to set the values of the non-animated variables.

1. Click the **Design Point** tab.
2. Deselect the **Use defaults** checkbox.
3. In the table, select the row corresponding to the variable setting of interest.

4. Click **OK**.

The animation begins in the view window. It will display one frame for each variable value.

The play panel appears in the upper-left corner of the desktop, enabling you to stop, restart, and control the speed and sequence of the frames.

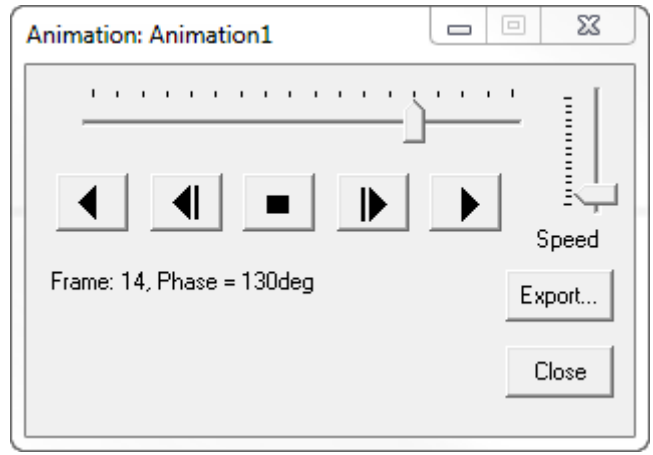
Related Topics

Creating Animations

Controlling the Animation's Display

Controlling the Animation's Display

When an animation is displayed in the view window, the **Animation** window, also called the *play panel*, appears in the upper-left corner of the desktop. It has buttons that enable you to control the speed and sequence of the frames, start and stop the animation and export the animation. Click an area of the window below to learn its function.



Animation slider	Each dot on the slider represents a frame in the animation. Drag the slider to the right to display the next frame in the animated plot. Drag the slider to the left to display the previous frame in the animation.
	Plays the plot's animation sequence backwards.
	Steps backward through the animated plot one frame at a time.
	Stops the animation.

Steps forward through the animated plot one frame at a time.

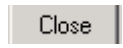
Plays the plot's animation sequence forwards.

Drag the **Speed** slider to the top to increase the speed of the animation. Drag the **Speed** slider to the bottom to decrease its speed.

Frame information

The current frame and phase at which the plot is being displayed is listed below the control buttons.

Enables you to export the animation to an animated Graphics Interchange Format (GIF) or to Audio Video Interleave (AVI) format.



Closes the animation window.

Related Topics

Creating Animations

Exporting Animations

1. Create the animation you want to export.
2. In the play panel, click **Export**.

   The **Save As** dialog box appears.
3. Follow the procedure for saving a new file. Select **Animated GIF File (.gif)** or **AVI File (.avi)** as the file type.

   The **Animation Options** dialog box appears.
4. To replace colors in the file with 256 shades of gray, select **Grayscale**.

   Grayscale animations tend to use less memory than full color animations.
5. For AVI format export, specify the **Compression factor** (the default is 85) and one of the following **Compression types**:

   INTEL Indeo

   Cinepak

   Microsoft Video 1

   None
6. For GIF format export, specify the number of loops. The default "0" denotes infinite loops.
7. Click **OK** to close the **Animations Options** dialog.

The animation is exported to the file format you specified.

Related Topics

Creating Animations

Creating Reports

After HFSS or HFSS-IE generates a solution, you can analyze all the results for that solution. HFSS lets you create 2D or 3D plots. A 2D or 3D plot shows the relationship between a design's values and the corresponding results of the analysis. You can create reports using either the **Create Quick Report** option, or the **Create <*type*> Report** commands. The **Quick Report** feature lets you select from a list of predefined categories (such as S-parameters) from which to create a rectangular plot.

For each solution **<type>** (Eigenmode, Modal, Fields, Far Fields and Emission test, and Terminal), the **Results** menus present a list of **Create <*type*> Report** commands based on the solution data of direct interest for the design. For example, for the Eigenmode solution type, the **Results** menu contains templates for Eigenmode Parameters and for Fields. These appear on the menus as **Create Eigenmode Parameters Report** and **Create Fields Report**. For the Modal and Terminal Solution types, several different types appear, appropriate to each solution type. Each of these **Create <*type*> Report** menu items includes a further cascading menu that lists the Display Types available for that report. For some reports you can modify the Display Type from the Properties for that Report. For reports for Transient designs, see this discussion.

If you have created custom report templates (for example, including your company name or other format changes), you can also create a report based on that template by selecting **HFSS** or **HFSS-IE>Results>Report Templates>PersonalLib>**<*templateName*>. You can also use the Report2D>Export feature and select ReportData File (.rdat) format file which you can then select for Create Report from File.

Related Topics

Creating a Quick Report

Creating a New Report

Modifying Reports

Zooming and Fitting Reports

Modifying the Background Properties of a Report

Creating Custom Report Templates

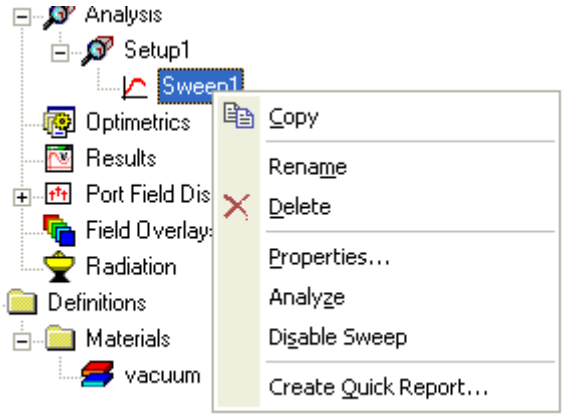
Plotting in the Time Domain

Report Setup Options (for maximum significant digits, and drag and drop behavior, and setting the size for displaying matrix in tree format.)

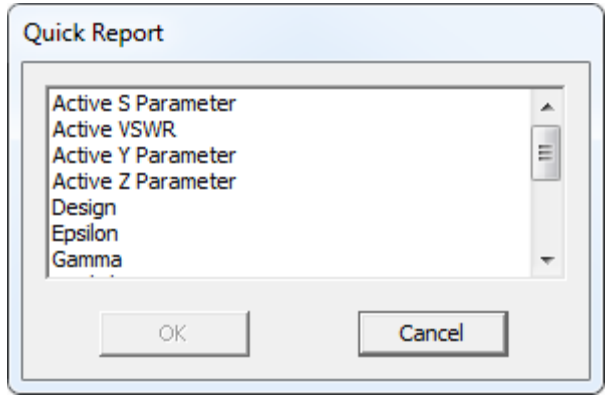
Creating a Quick Report

Following is the procedure for creating a quick report.

1. On the Project tree under "Analysis", select a **setup** or **sweep icon**, or the **Results** icon.
2. Right-click to display the shortcut menu and select **Create Quick Report**.



The **Quick Report** dialog appears.



3. Select the one or more categories for the report from the list and click OK.

   A rectangular plot for each selected category displays. The new plot or plots appear in the Project tree under the Results icon.

Related Topics

Creating Reports

Modifying Reports

Creating Custom Report Templates

Creating a New Report

Following is the general procedure for creating a new report:

1. On the HFSS or HFSS-IE menu or the Project tree, point to **Results**, and then select **Create <*type*> Report** and from the menu select the **Display Type** for that template**.** There are more templates of **Report Types** available for terminal solutions (terminal, model, fields, near fields, and far fields) and for modal solutions (modal and fields). For Eigenmode solutions, the **<*templates*>** of **Report Types** are for Eigenmode Parameters and for Fields. If an HFSS design contains IE Regions, you can select the IE Surface Fields Report Type.

   If you have created custom report templates (for example, including your company name or other format changes), you can also create a report based on that template by selecting **HFSS** or **HFSS-IE Results>Report Templates>PersonalLib>**<*templateName*>.

   When you have selected the **<*type*>** and display type from the **Results** menu, the **Report** dialog appears, with the **Trace** tab selected by default.
2. In the Context section you make selections depending on the design and solution type.
3. In the **Y** Component section of the dialog make selections for the following:
   a. Categories - those depend on the Solution type and the design. For example, Eigenmode quantities include Eigenmodes, variables, output variables, and the design. Driven solutions include such categories as S parameters. Report categories for Transient designs include Spectral and Transient. For a Transient Network design with differential pairs defined, the Reporter interface allows selection of single-ended or differential signals just as for driven terminal.
   b. Quantities for Y are relative to the selected category.

Note The Quantity text field can be used to filter the Quantity list by typing in text, or by using the four predefined selections. This is useful if the Category selected produces a lengthy Quantities list. See Filtering Quantity Selections for the Reporter.

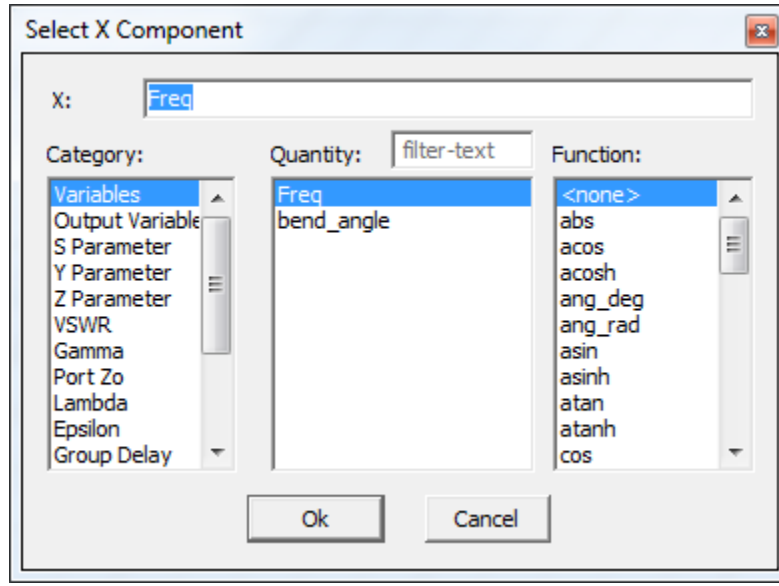
When the matrix is very large, the number of quantities can be correspondingly huge. Therefore, the Quantities field can optionally use a tree structure to divide matrix quantities into groups by their first element name. The initial display shows groups, without initially listing group members. See Report Setup Options.

   c. **Function** list to apply to the Y quantities.
   d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

Note Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog. This applies to the currently specified Quantity and Function.
4. In the **X (Primary Sweep)** section, make selections for the following:
   a. Select the Primary value(s) from the drop down menu.

To select an X component that is different than the Primary Sweep, uncheck the Default field to enable the X field and browse [...] button. Click the browse [...] button to display the **Select X Component** dialog.



This lets you specify the X component as you do the Y; that is, in terms of Categories which define the selectable Quantities, and Functions to apply. After making selections, **OK** the dialog to assign the X component.

b. If sweeps are available, you can select the browse [...] button to display a panel that lets you select Use all values, or selected sweep or sweeps, or access an Edit Sweep dialog. with further editing options. Post-Processing variables are Post-Processing sweeps/editable sweeps, so you can use the **Edit Sweep** dialog to create your own sweep.

c. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (for example, for a parametric sweep). If so, the variables other than the one chosen as the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the value, and an Edit column with an ellipsis [**...**] button. See Using Families tab for Reports.

5. **Update Report** setting
   - **Real Time** checked -- enable real time updates for all reports while the reports are being edited.
   - **Real Time** unchecked -- enables drop down menu to **Update All Reports** or **Update Report**. Reports will only be updated with one of these user selectable update options or upon exiting the report dialog. This can be useful if you expect a trace to take time to display. You can then add additional traces without having to wait.
6. The **Report** dialog command buttons permit you create a new report with the settings you provide, or to modify an existing report.

- **Output Variables** - opens the **Output Variables** dialog.
- **Add Trace** - this is enabled when you have created or selected a report. Add one or more traces to include in the report.
- **Update Trace** - updates the selected traces in a report based on further processing or changes.
- **New Report**. Adds a report to the Project tree under the Results icon. The new Report is displayed in the Project window.
- **Options** - opens the **Report Setup Options** dialog. This contains a checkbox for using the advanced mode for editing and viewing trace components. This mode is automatic if the trace requires it. It also contains a field for setting the maximum number of significant digits to display for numerical quantities.
- **Close** - closes the **Report** dialog.

7. Click **New Report** to create a new report in the Project tree.

   The report appears in the view window. It will be listed in the project tree under Reports. Traces within the report also appear in the project tree.

   Some plots may take time to complete. Performing a **File>Save** in such cases after the plot has been created will permit you to review the plot later without having to repeat the calculation time when you reopen the project later.

8. To speed redraw times for changed plots, perform a **Save**. This saves the data that comprises expressions. For example if re(S_{11})*re(S_{22}) is requested over multiple widths, each of the S_{11} and S_{22} are stored when you save. If you do not do a save of a changed plot, the changed version is not stored.

Note Remember the evaluated value of an expression is always interpreted as in SI units. However, when a quantity is plotted in a report, you have the option to plot values in units other than SI. For example, the expression "1+ang_deg(S_{11})" represents an "angle" quantity evaluated in radians, though plotted in degree units. To represent an angle quantity in degrees, you would specify units as "1 deg + ang_deg(S_{11})".

Related Topics

Creating Reports

Modifying Reports

Zooming and Fitting Reports

Creating a Quick Report

Using Families tab for Reports

Context Section for Reports

Plotting in the Time Domain

Context Section for Reports

In the **Context** section make selections from the following field or fields, depending on the design and solution type.

1. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes. The selections for HFSS Transient include Transient, and for Transient Network includes Spectral.
2. Domain field with a drop down selection list. Whether this field appears, and the domains listed depend on the Solution type and the **<*type*>** selected. For modal and terminal solution data r reports, the domain can be **Sweep** or **Time**.

   Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.
3. Geometry field with a drop down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.
4. Show field with a drop down selection list for Differential pairs or Terminals. This field appears for designs using terminal solutions that have differential pairs defined. It lets you plot either differential pair data, or single-ended terminal data, or both in the same plot without having to disable or enable differential pairs under the Excitations heading in the Project tree. Note that single-ended quantities are computed as if no differential pairs existed. So in the unlikely case of several terminals where only a subset is combined into pairs, the results may not be as expected.

5. Derivative field with a drop drown selection list of none, all, and specific variables for which you specified Use on the Derivatives tab of the solution setup. You can use derivatives in some Optimetrics situations, and with the Derivative Tuning feature in the Reporter.
6. A Sources combo box appears in the Reporter when you have specified at least one source in the **Edit Sources** dialog. See Specifying Source Contexts for Creating Radiated Field Reports. For an example use, see User Defined Solution for MIMO Calculations.

Related Topics

Creating Reports

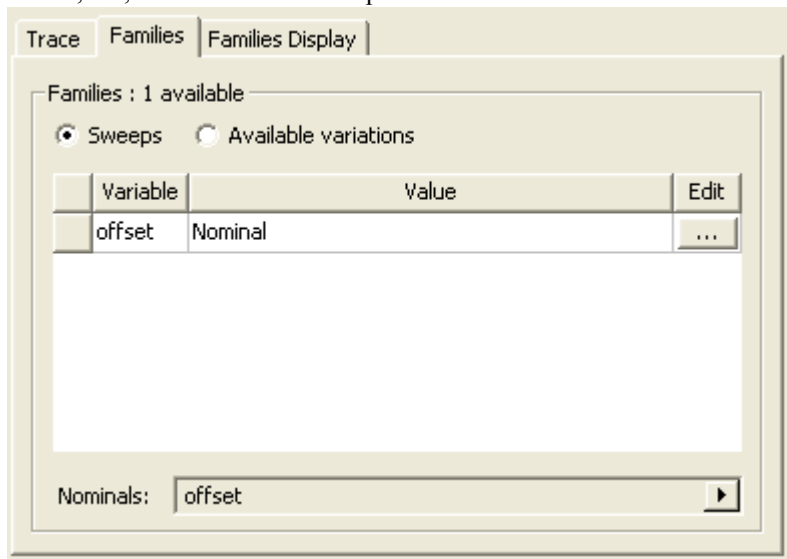
Modifying Reports

Using Families tab for Reports

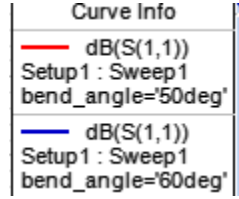
Selecting the Report Type in HFSS Transient

Using Families Tab for Reports

The **Families** tab of the **Report** dialog provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined and solutions exist for multiple variable values (for example, for a parametric sweep or re-running an analysis with a different variable value). If no variables are defined, or none have solutions for different values, 0 families will be available. If so, the variables other than the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the solution value (which may be All, Nominal, or a Specific value), and an Edit column with an ellipsis [**...**] button. Families gives the number available. If an existing variable is specified as Nominal, only that value is currently available. You can set any solved variables as Nominal, All, or select from values provided for Available solutions.

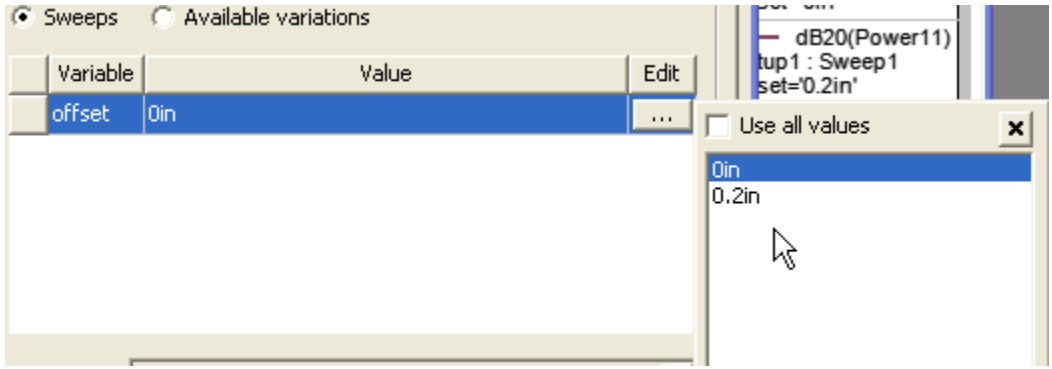


When you select a variable with multiple solved values, a trace for each solved value appears in the Report, with the variable value appended to the trace name in the Report legend.

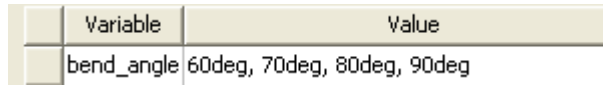


When families are available, you can make selections for the following:

1. Select the **Sweeps** radio button (the default) to list the swept variables you can select or the Available variations button to list and select variation values for which solutions exist.
2. With the **Sweeps** radio button selected, click the ellipsis [**...**] button to display a list of variable values for a particular variable. If many variables exist, you can use a scroll bar to navigate the list.



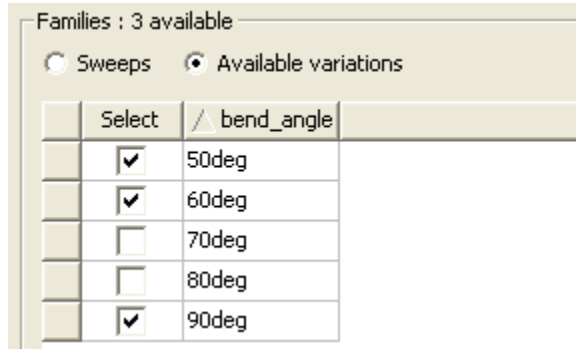
- To select all values, click the checkbox for **Use all** values. This writes "All" in the value field for that variable. You can also select individual values by clicking on them.
- To select a range of values, hold down the shift key, and click again.
- To select intermittent additional values, hold the CTRL key and click additional. The values you select are highlighted in the list, and are also listed in the Values column for that variable.
- To select all, use the **Select All** button. This highlights the complete list, as well as listing all values for the variable in the Value field.



- To clear the selections, use the **Clear All** button.

.Select the **Available variations** radio button to list the choices that derive from variable com-

bination.



a. To select individual variations, check the select box.

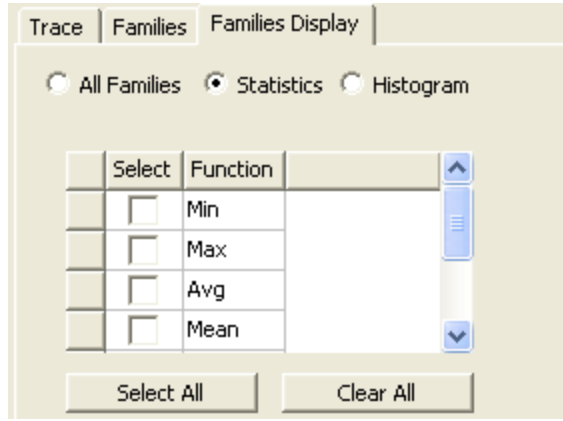
b. To check or clear all variations at once, click the **Select** button at the top of the column.

c. To invert the list order, click the triangle beside the variable name.

The **Families Displa**y tab has three radio button selections.

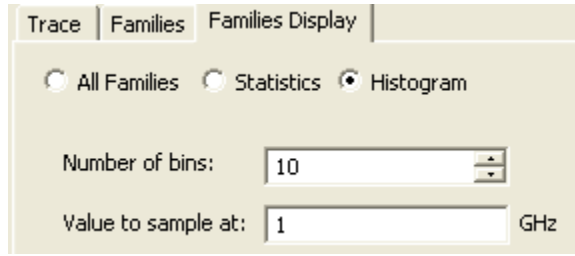
a. All Families

b. **Statistics** which lists a table statistical functions that you can select to apply to the plot. The functions include Min, Max, Avg, Mean, Variance, Std Dev, and Sum. You can use the Select checkboxes or the Select All and Clear All button.



c. **Histogram** which lets you select the number of bins to use for a histogram plot, and the

sampling frequency to use.



Related Topics

Creating Reports

Modifying Reports

Modifying the Background Properties of a Report

Modifying the Legend in a Report

Creating Custom Report Templates

Working with Traces

Editing the Display Properties of Traces

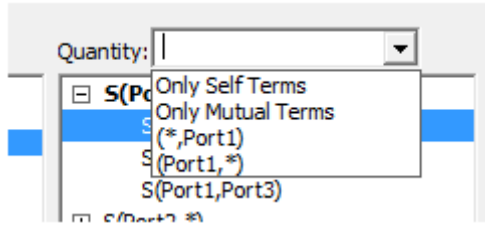
Discarding Report Values Below a Specified Threshold

Add Trace Characteristics

Adding Data Markers to Traces

Filtering Quantity Selections for the Reporter

When a two port quantity Category is selected, four predefined filters are added to the combo box. "Port1" is the first matrix element name found in the quantity list.



- Only Self Terms -- Only display quantities when the first and second port are same.
- Only Mutual Terms -- Only display quantities when the first and second port are different.
- (*,Port1) -- Only display quantities when the second element name is "Port1". You can edit the element name to display quantities for other elements.
- (Port1,*) - Only display quantities when the first element name is "Port1". You can edit the element name to display quantities for other element.

Related Topics

Report Setup Options.

Modifying Reports

To modify the data that is plotted in a report:

1. In the project tree, click the report you want to modify.
2. Right-click **Modify Report**.

   The **Report** dialog appears.
3. The **Report** dialog command buttons permit you create a new report with the settings you provide, or to modify an existing report.
   - **Output Variables** - opens the **Output Variables** dialog.
   - **Add Trace** - this is enabled when you have created or selected a report. Add one or more traces to include in the report.
   - **Update Trace** - updates the selected traces in a report based on further processing or changes.
   - **New Report** - adds a report to the Project tree under the Results icon. The new Report is displayed in the main window.
   - **Options** - opens the **Report Setup Options** dialog. This contains a checkbox for using the advanced mode for editing and viewing trace components. This mode is automatic if the trace requires it. It also contains a field for setting the maximum number of significant digits to display for numerical quantities.
   - **Close** - closes the **Report** dialog.

   The updated report appears in the view window.
4. **Update Report** setting
   - **Real Time** checked -- enable real time updates for all reports while the reports are being edited.
   - **Real Time** unchecked -- enables drop down menu to **Update All Reports** or **Update Report**. Reports will only be updated with one of these user selectable update options or upon exiting the report dialog. This can be useful if you expect a trace to take time to display. You can then add additional traces without having to wait.
5. In the Context section you make selections depending on the design and solution type.
6. The Families tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (for example, for a parametric sweep). If so, the variables other than the one chosen as the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the value, and an Edit column with an ellipsis [**...**] button. See Using Families tab for Reports.
7. In the **Y** Component section of the dialog make selections for the following:
   a. Categories - those depend on the Solution type and the design. For example, Eigenmode quantities include Eigenmodes, variables, output variables, and the design. Driven solutions include such categories as S parameters. Report categories for Transient designs include Spectral and Transient. For a Transient Network design with differential pairs defined, the Reporter interface allows selection of single-ended or differential signals just

as for driven terminal.

b. Quantities for Y are relative to the selected category.

Note The Quantity text field can be used to filter the Quantity list by typing in text, or by using the four predefined selections. This is useful if the Category selected produces a lengthy Quantities list. See Filtering Quantity Selections for the Reporter.

When the matrix is very large, the number of quantities can be correspondingly huge. Therefore, the Quantities field can optionally use a tree structure to divide matrix quantities into groups by their first element name. The initial display shows groups, without initially listing group members. See Report Setup Options.

c. **Function** list to apply to the Y quantities.

d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

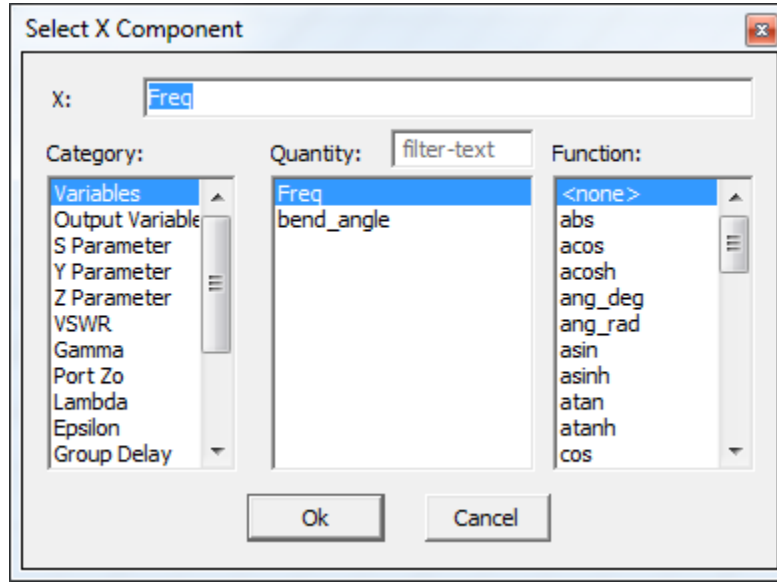
Note Color shows valid expression.

e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.

8. In the **X (Primary Sweep)** section, make selections for the following:

a. Select the Primary value(s) from the drop down menu.

To select an X component that is different than the Primary Sweep, uncheck the Default field to enable the X field and browse [...] button. Click the browse [...] button to display the **Select X Component** dialog.



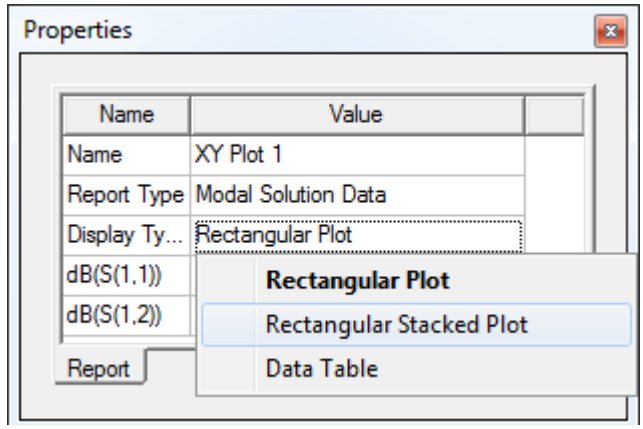
This lets you specify the X component as you do the Y; that is, in terms of Categories which define the selectable Quantities, and Functions to apply. After making selections, **OK** the dialog to assign the X component.

b. If sweeps are available, you can select the browse [...] button to display a panel that lets you select Use all values, or selected sweep or sweeps, or access an Edit Sweep dialog. with further editing options. Post-Processing variables are Post-Processing sweeps/editable sweeps, so you can use the **Edit Sweep** dialog to create your own sweep.

c. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (for example, for a parametric sweep). If so, the variables other than the one chosen as the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the value, and an Edit column with an ellipsis [**...**] button. See Using Families tab for Reports.

You can also view and edit the properties of Reports and their traces via their Properties windows. See Modifying the Background Properties of a Report.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the **Project** tree to display the **Properties** dialog. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Note Remember that for many excitations of interest for plotting, you can control the default base names through the dialog described here: Setting Default Boundary/Excitation Base Names.

This may save you the need to edit individual names in the plots.

Related Topics

Selecting the Report Type in HFSS Transient

Zooming and Fitting Reports

Modifying the Background Properties of a Report

Modifying the Legend in a Report

Creating Custom Report Templates

Report Setup Options (for maximum significant digits and drag and drop behavior)

Working with Traces

Editing the Display Properties of Traces

Setting Default Boundary/Excitation Base Names

Discarding Report Values Below a Specified Threshold

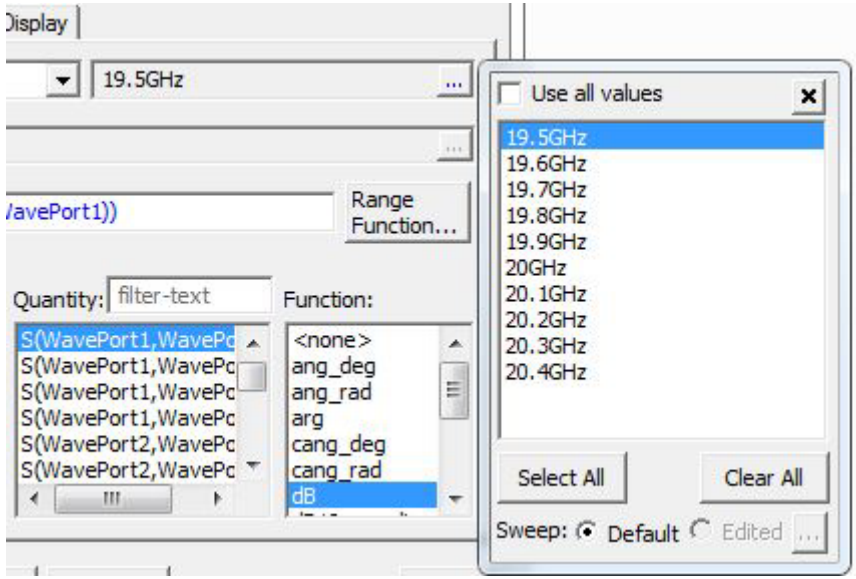
Add Trace Characteristics

Adding Data Markers to Traces

Perform FFT on a Report

Modify Report: Selecting Use all values or Making Selection

Clicking the browse button on Primary Sweep line shows the default selection to Use all values, and the sweeps grayed out. Uncheck the Use all values box to enable editing, including the **Select All** and **Clear All** buttons.



With Use all values unchecked, you can select one or more by clicking an individual value, dragging to select multiple values, or using Alt-Click to specify specific values.

You select either the Sweep radio button for Default or Edited selection.

You can also select the browse [...] button here to display the Edit Sweep dialog for Modify Reports, which includes additional editing features.

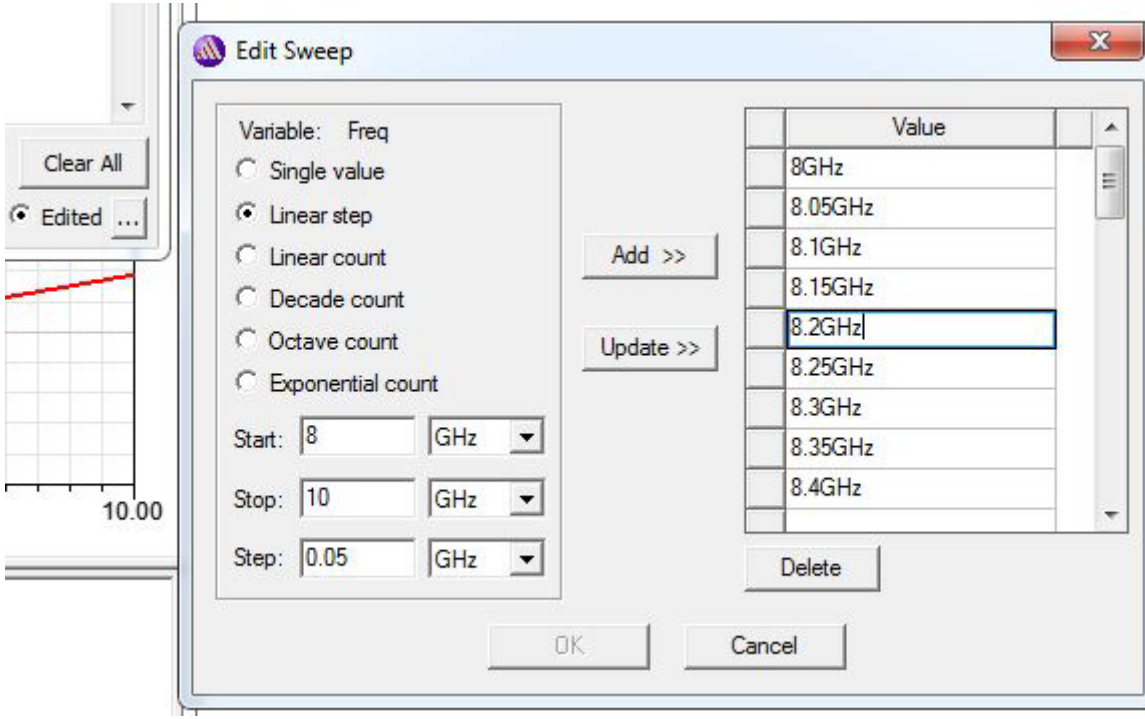
Related Topics

Creating Reports

Modifying Reports

Modify Report: Using the Edit Sweep Dialog

Clicking the browse [...] button at the lower right corner of the Use all values pane opens the Edit Sweep dialog for Modify Report. The lets you edit the current Primary sweep variable values, including radio button selections for Single value, Linear step or count, and Decade, Octave, or Exponential Counts.



You can specify start, stop and step values and units, and add specific values to the list of current sweep values. The **Add>>** and **Update>>** buttons let you edit the value list.

You can use the mouse click, drag, and Alt-Click to select values. You can also edit individual values.

Related Topics

Creating Reports

Modifying Reports

Creating a Report from an Ansoft Report Data File

If you have previously saved an Ansoft Report Data Format (using Report2D>Export), you can create a report from that rdat file. This provides a way to reuse the data and/or the format of a previously created report.

1. Right-click on the Results icon in the Project tree to display the short cut menu and select **Create Report From File**, or click **HFSS** or **HFSS-IE>Results>Create Report From File**.

   A file browser displays.

2. Select an .rdat format file, and click **Open**.

   The report is created. If it contains data, it displays the report with traces. If not, the report uses the format exported to the .rdat file.

Related Topics

Creating Custom Report Templates

Exporting Ansoft Report Data Format Files

Zooming and Fitting Reports

The standard Zoom and Fit commands operate on reports. After clicking in an open report, you can also use a mouse wheel, to zoom in and out.

Related Topics

Modifying the Background Properties of a Report

Modifying the Background Properties of a Report

To modify the appearance of a report, or the display properties an object in a report.

1. Open the report you want to modify.
2. You must select an editable object in the report to be able to edit its properties. Click on an object to select it and to view its Properties in the docked properties window. To open a floating Properties window, either double click on the selected object, or click **Edit>Properties** on the toolbar.

   The selectable objects in reports are as follows:

   - **Header** -- this lets you edit the Properties for the text displayed at the top of the report, including the Title font, Company Name, Show Design Name, Subtitle Font. The plot title is tied to the report's name and is not a Header property. If you change the report name in the Project tree, plot title synchronizes. The Company Name and the Show Design Name checkbox are grouped in the Properties dialog as Subtitle. Edits to the Subtitle Font Property affects both of them.
   - **General** -- this dialog (or General tab for other Report properties windows) lets you edit the background color (the perimeter around the trace display) for the plot, the contrast color (the trace display background), the Field width, the Precision, and whether to use scientific notation for marker and delta marker displays. (X and Y notation display is set separately, in the Axis property tabs.)
   - **Legend** -- this lets you edit the Properties for whether to Show Trace Name, Solution

Name, and Variation Key. At least one of these three must be selected. You can also edit the Font, the background color of the Legend box, the Border Color, the Border Width, Grid Color (for the lines between Trace descriptions), and the Grid line width. Also see Modifying the Legend in a Report

- **Traces** -- you can select traces either in the Legend or on the plot. The properties for traces include: Color, Line Style, Line Width, Trace Type, whether to Show a symbol, Symbol Frequency, Symbol style, whether to Fill symbol, symbol color, and whether to Show arrows. See Editing the Display Properties of Traces.
- X or Y **Axis Tab**-- the defaults for most of these values are set in the **Report 2D Options** Axis tab.
  - Display name -- checkbox for whether to display the axis name.
  - Specify name -- checkbox for specifying the Axis name.
  - Name -- this describes the axis to which the following properties/options refer. These are selected in the **Report** dialog.
  - Axis Color -- set the color by double clicking to display the Set color dialog. Select a default or custom color and click OK.
  - Text Font -- click the cell to display the Edit Text Font dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.
  - Display Units -- this specifies whether to display units.
  - **Window (section)**
    - Window Mode-- can be Axis range, Continuous moving window, or Step moving window.
    - Window Width (in) -- provide an integer value for the previous selection.
  - **Manual Format** (section)
    - Number format -- select from the drop down menu, Auto, Decimal, or Scientific notation.
    - Field Width -- enter a real value.
    - Field Precision -- enter a real value.
  - X or Y **Scaling Tab** -- These properties provide control over scaling.
    - Axis Scaling -- use the drop down menu to select scaling as Linear or Log. For the Y axis, all zero or negative values are discarded before log scaling is applied.
    - Specify Min -- check box
    - Min -- text entry in same units as axis units. Saved as SI internally.
    - Specify Max -- check box
    - Max -- text entry in same units as axis units. Saved as SI internally.
    - Specify Spacing -- check box

- Spacing -- text entry in same units as axis units. Saved as SI internally
- **Manual Units** (section)
- Auto Units -- use the check box compute the correct units for the axis.
- Units -- click on the cell to select from a menu of available units if you have not checked Auto Units.
- **Infinity Visualization** (section)
- Map Infinity Mode -- checkbox.

  Each axis now can be set to treat infinity values in a user defined way. When you check the Map Infinity Mode, any infinity values in the input data get the infinityMap value (negative infinity get the value*-1 and positive infinity the positive value specified). This can be useful if there are zeros, or very small values that HFSS treats as zero, in the data, for example, dB Gain.
- Map Infinity To -- enter a real value for the Map Infinity Mode.

3. Edit the properties, and **OK** the dialog to apply the changes.

Related Topics

Modifying Reports

Zooming and Fitting Reports

Working with Traces

Discarding Report Values Below a Specified Threshold

Modifying the Legend in a Report

Editing the Display Properties of Traces

Creating Custom Report Templates

Exporting Ansoft Report Data Format Files

Setting Report2D options

Zoom in or out.

Fit contents in the view window.

Modifying the Legend in a Report

The legend in a report is a list of the curves being plotted. For each curve, the legend gives the name, shows the line color, and lists the setup and the adaptive pass used to generate the curve.

To **show** or **hide** a legend in a report:

1. Make the report the active view.
2. Use **View>Active View Visibility** or the **Show/Hide** icons on the toolbar to display or hide the report.

   Either command displays the **Active View** dialog.
3. Select the **Legends** tab.

   This lists the legend (or legends) in the report.

4. Check the visibility checkbox, and OK the dialog to close it and apply the change.

To **edit the display properties** of a legend:

1. Select the legend in a report by clicking on the Curve Info panel to display a docked properties window, or right-click on the legend and select **Edit>Properties** to display the floating properties window.

   This lets you edit the Properties for whether to Show Trace Name, Solution Name, and Variation Key. At least one of these three must be selected.

   You can also edit the Font by clicking the Font cell to display the **Edit Text Font** dialog. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog

   You can also edit the background color of the Legend box, the Border Color, the Border Width, Grid Color (for the lines between Trace descriptions), and the Grid line width.
2. Click OK to close the Properties window and apply the selections.

To **change the display name** for traces, see Editing Trace Properties.

To **move** a legend in a report:

1. Click and hold and the legend.

   The cursor changes to crossed lines with arrow tips.
2. Still holding, drag the legend to a new location and release.

   The legend is released and the crossed lines change back to a mouse pointer.

To **resize** a legend in a report:

1. Position the mouse tip over the edge you want to resize.

   The mouse pointer changes to a horizontal or vertical line with arrow tips.
2. Click and drag the horizontal or vertical edge to the desire size.
3. Release.

Related Topics

Editing Trace Properties

Showing Objects

Hiding Objects

Modifying Reports

Creating Custom Report Templates

Discarding Report Values Below a Specified Threshold

Setting Report2D options

Editing the Display Properties of Traces

Creating Custom Report Templates

You can edit properties from any report type and save it as a template. This can save repeated editing of properties (for example, the company name, or color schemes) when you create other reports. Once you create templates, you can access them from the **Results>Report Templates>** menu.

See Modifying the Background Properties of a Report for a discussion of format changes you can make to any report.

To save an edited report as a template:

1. In the Project Tree, right-click on the report name of interest to display the shortcut menu and click **Save as Template**:

   This displays the **Report Save As** file browser. By default, the directory is your AnsysEM\\<*productName*>\userlib\ReportTemplates directory. You can also save to the SysLib directory.
2. Typically, you accept the directory.
3. You must provide a file name, which will be given an *.rpt extension.

   It is good practice to give the template a descriptive name, showing both the kind of format you begin with (such as XY Plot or 3D Plot) and apt description of the distinguishing edits (such as for company name, or color scheme). Once, saved, this name will appear on the **PersonalLib** menu.
4. The **Save As Type** field currently supports the Ansoft Report Format (*rpt) format.
5. Click **Save** to save the template to the PersonalLib menu.

All *.rpt templates in the userLib directory appear on the **Results>Report Templates>PersonalLib** menu. Selecting a report from the PersonalLib menu opens a report that you can then Modify to add traces or perform other edits. Templates in the SysLib directory appear on the **Report Templates** menu.

Related Topics

Modifying Reports

Setting Report2D options

Exporting Ansoft Report Data Format Files

Zoom in or out.

Fit contents in the view window.

Modifying the Background Properties of a Report

Modifying the Legend in a Report

Working with Traces

Editing the Display Properties of Traces

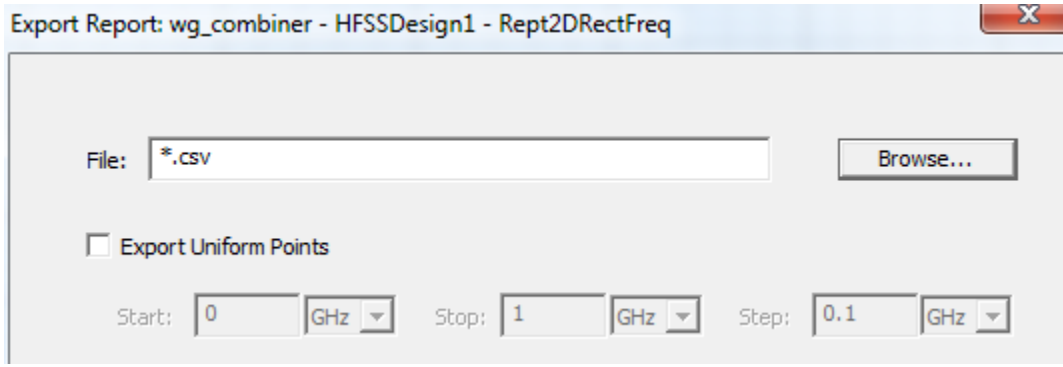
Discarding Report Values Below a Specified Threshold

Exporting Ansoft Report Data Format Files

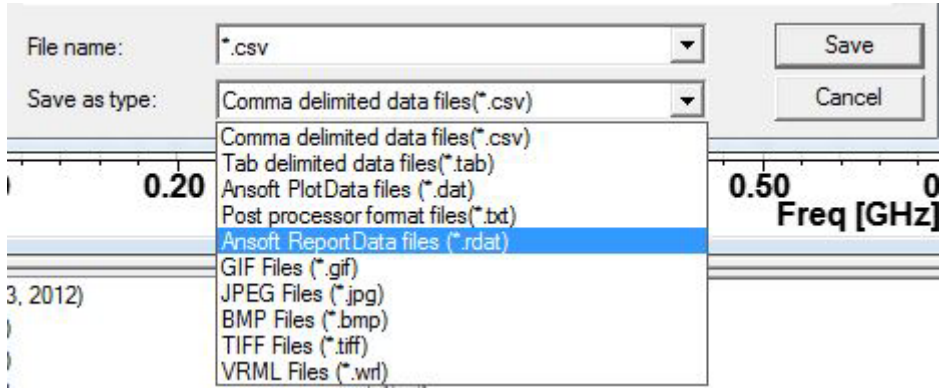
Ansoft Report Data format files provide a way to export reports or report formats, which you can then import using Reports>Create Report From File. This can save repeated editing of properties (for example, the company name, or color schemes) when you create other reports. You must have an existing plot open to see the **Report2D** menu.

1. Click **Report2D**>**Export...**

   The **Export Report** dialog box appears.



2. If you check, Export Uniform Points, the rdat file will contain the points for the given start, stop, and step at the given frequencies. If you do not, the file contains only the current file format, including any modifications you have applied.
3. Use the file browser to find the directory where you want to save the file.
4. Type the name of the file in the **File** box.
5. Select the Ansoft Report Data (.rdat) file formats from the **Save as type** pull-down list:



6. Click **Save**. The file is exported to the specified location as an Ansoft Report Data file.

The file will then be available for import using Create Report from file. If you

Related Topics

Creating a Report from an Ansoft Report Data File

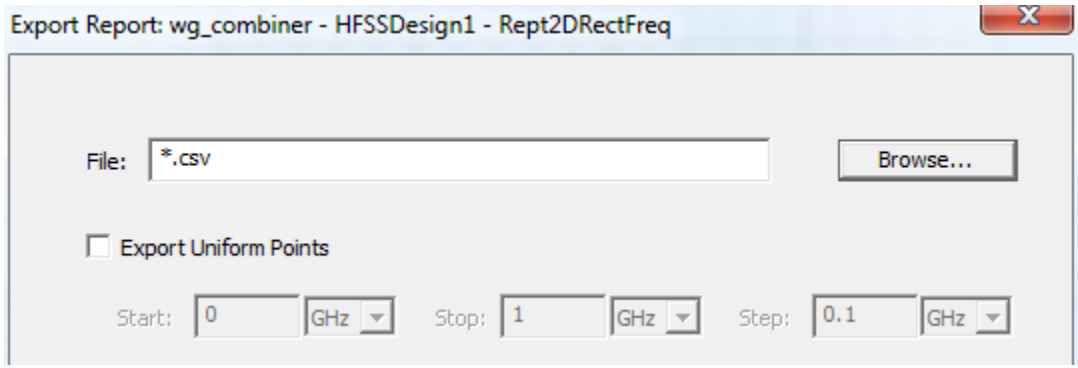
Exporting Reports as Graphics
Creating Custom Report Templates

Exporting Reports as Graphics

You can export reports as figures in several formats. You must have an existing plot open to see the **Report2D** menu.

1. With a report open, click **Report2D**>**Export...**

   The **Export Report** dialog box appears.



2. Click the **Browse..**. button to open the Export Report browser window.
3. Specify the file location and name, and select a graphics format from the dropdown list.

Extension	Contents
.bmp	Bitmap files.
.gif	Graphics Interchange Format files.
.jpeg	Joint Photographics Experts Group files.
.tiff	Tagged Image File Format files.
.wrl	Virtual Reality Modeling Language (VRML) files.

4. Click **Save** to close the browser window, and then **OK** to close the **Export** window.

Related Topics

Exporting Graphics Files

Selecting the Report Type

The **Report Types** available for creating a report depends on the simulation setup. Depending on the setup, you can make a selection from the following report types:

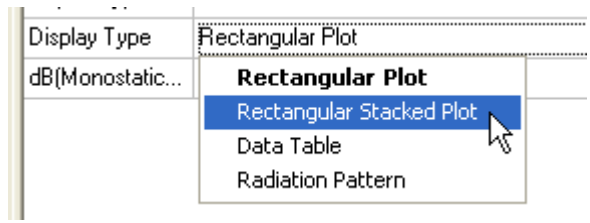
Modal Solution Data	S-, Y-, and Z-parameter data will be available to plot, as well as propagation constant, characteristic port impedance, reflection/ transmission coefficients for FSS designs, and voltage standing wave ratio (VSWR) data. **Note:** For FSS calculations, phase is currently assigned zero value.
Terminal Solution Data	This solution type results in a terminal-based description in terms of voltages and currents. Some modal data is also available. The terminal-based S-, Y-, and Z parameters, voltage standing wave (VSWR), Port Zo, and Active S-, Y-, Z, and VSWR parameters are available to plot.
Eigenmode Parameters	The Eigen Modes and Eigen Q data are available to plot.
Fields	Basic or derived field quantities calculated on lines or integrated over surfaces or objects will be available to plot.
IE Surface Fields	For an HFSS design that uses IE Regions the solver produces J and Q surface fields for the metallic parts (but not the dielectric parts) of these regions. The Fields Calculator provides access to the IE surface fields quantities via the existing Field Type pulldown in the Context Area.
Far Fields	Radiated fields computed in the far-field region. The following quantities will be available to plot: rE, gain, realized gain, directivity, axial ratio, polarization ratio, antenna parameters, and normalized antenna calculated by HFSS. Note: You must have defined an infinite sphere geometry and at least one radiation or PML boundary to create a far-fields report.
Near Fields	Radiated fields computed in the near-field region. These include: variables, output variables, near E, max near field parameters, and near normalized antenna. Note: You must have defined a near-field line or near-field sphere and at least one radiation or PML boundary to create a near-fields report.
Emission Test	You can conduct an emission test under the same conditions as for a near field report except that.an emission test cannot be conducted for a ports-only solution. You must have defined a near-field line or near-field sphere and at least one radiation or PML boundary.

Selecting the Display Type

The information in a report can be displayed in several formats. For the initial plot, you can select from the following **Display Type** formats in the **Create <*type*> Report** submenu:

Rectangular Plot	A 2D rectangular (x-y) graph.
Rectangular Stacked Plot	This choice puts each trace into its own 2D rectangular plot, and stacks each plot, rather than overlaying the traces on the same plot.
3D Rectangular Plot	A 3D rectangular (x-y-z) graph.
Rectangular Contour Plot	A rectangular (x-y-z) graph. Contour plots are useful to visualize surfaces (for e.g. Directivity as a function of phi/theta).
Polar Plot	A 2D circular chart divided by spherical coordinates.
3D Polar Plot	A 3D circular plot divided by spherical coordinates.
Smith Chart	A 2D polar chart of S-parameters upon which a normalized impedance grid has been superimposed.
Smith Contour Plot	A polar chart. Contour plots are useful to visualize surfaces.
Data Table	A grid with rows and columns that displays, in numeric form, selected quantities against a swept variable or another quantity.
Radiation Pattern	A 2D polar plot of radiated fields.

You can also modify the display type of an existing plot from the **Properties** dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog. Selecting the **Display Type** field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Related Topics

Selecting the Report Type

Creating Reports

Modifying Reports

Creating Custom Report Templates

Creating 2D Rectangular Plots

A rectangular plot is a 2D, x-y graph of results.

1. On the **Results** menu (HFSS menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Rectangular Plot**.

   The **Report** dialog appears.

2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.

   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.

   b. Domain field with a drop down selection list. Whether this field appears, and the domains listed depend on the Solution type and the **<*type*>** selected. For modal and terminal solution data reports, the domain can be **Sweep** or **Time**.

      Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.

   c. Geometry field with a drop down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.

3. Under the **Trace** tab, **Y** component section, specify the information to plot along the y-axis:

   a. In the **Category** list, click the type of information to plot.

   b. In the **Quantity** list, click the value to plot.

   c. In the **Function** list, click the mathematical function of the quantity to plot.

   d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

   Note Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.

4. On the **Trace** tab, **X** (Primary sweep) line, specify the quantity to plot along the x-axis in one of the following ways:.

   - Select the sweep variable to use from the drop down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular sweep or sweeps, or all sweeps. The quantity will be plotted against the primary sweep variable listed.

5. On the **Families** tab, confirm or modify the sweep variables that will be plotted.

6. Click **New Report.**

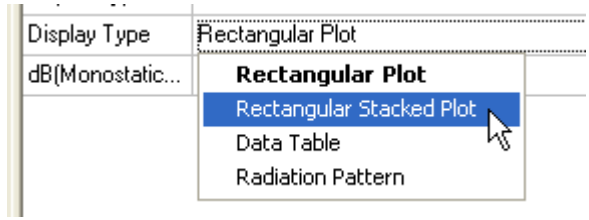
   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity will be plotted against the swept variable values or quan-

tities you specified on an x-y graph. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Related Topics

Sweeping a Variable

Working with Traces

Add Trace Characteristics

Delta Markers in 2DPlots

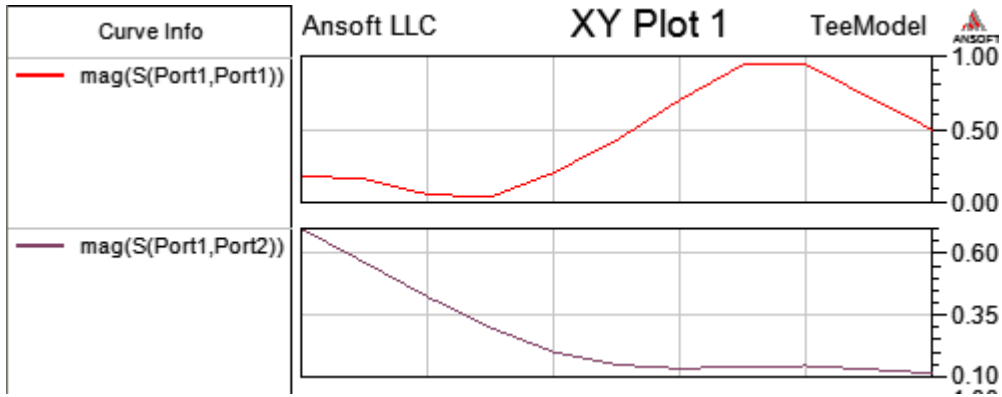
Modifying Background Properties of a Report

Discarding Report Values Below a Specified Threshold

Setting Report2D options

Creating 2D Rectangular Stacked Plots

A rectangular stacked plot is a 2D, x-y graph of results, with each trace displayed on a separate plot.



1. On the **Results** menu (HFSS menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Rectangular Stacked Plot**.

   The **Report** dialog appears.

2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.

   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.

   b. Domain field with a drop down selection list. Whether this field appears, and the domains listed depend on the Solution type and the **<*type*>** selected. For modal and terminal solution data reports, the domain can be **Sweep** or **Time**.

      Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.

   c. Geometry field with a drop down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.

3. Under the **Trace** tab, **Y** component section, specify the information to plot along the y-axis:

   a. In the **Category** list, click the type of information to plot.

   b. In the **Quantity** list, click the value to plot.

   c. In the **Function** list, click the mathematical function of the quantity to plot.

   d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

Note Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently

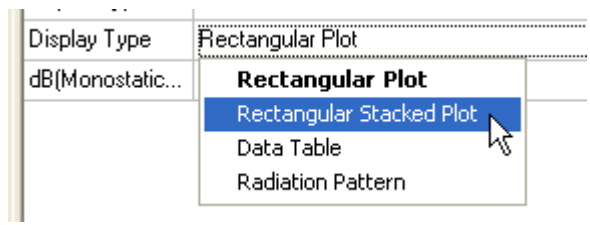
specified Quantity and Function.

4. On the **Trace** tab, **X** (Primary sweep) line, specify the quantity to plot along the x-axis in one of the following ways:.
   - Select the sweep variable to use from the drop down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular sweep or sweeps, or all sweeps. The quantity will be plotted against the primary sweep variable listed.
5. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
6. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity will be plotted against the swept variable values or quantities you specified on an x-y graph. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.
7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Related Topics

Sweeping a Variable

Working with Traces

Add Trace Characteristics

Delta Markers in 2DPlots

Modifying Background Properties of a Report

Discarding Report Values Below a Specified Threshold

Setting Report2D options

Creating 3D Rectangular Plots

This is a 3D, x-y-z graph of results.

1. On the **Results** menu (HFSS menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **3D Rectangular plot** from the report type menu.

   The **Report** dialog appears.

2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
   b. Geometry field with a drop down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.
3. Under the **Trace** tab **Z** Component area, specify the information to plot along the z-axis:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.
4. On the **Trace** tab **Y** (Secondary sweep) lines, specify the information to plot along the y-axis in one of the following ways:
   - Select the sweep variable to use from the drop down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.
5. On the **Trace** tab **X** (Primary sweep) lines, specify the information to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the drop down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.
6. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity or quantities will be plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the project tree.When you select the

traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

Related Topics

Sweeping a Variable

Working with Traces

Add Trace Characteristics

Creating Rectangular Contour Plots

This is an x-y-z graph of results. Any data that you can current plot in 3D (as 3D cartesian or 3D polar) is a candidate for a contour plot.

1. On the **Results** menu (HFSS menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Rectangular Contour plot** from the report type menu.

   The **Report** dialog appears.

2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
   b. Geometry field with a drop down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.
3. Under the **Trace** tab **Z** Component area, specify the information to plot as contours:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.
4. On the **Trace** tab **Y** (Secondary sweep) lines, specify the information to plot along the y-axis in one of the following ways:
   - Select the sweep variable to use from the drop down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.
5. On the **Trace** tab **X** (Primary sweep) lines, specify the information to plot along the x-axis in

one of the following ways:

- Select the sweep variable to use from the drop down list.
- If sweeps are available, you can select the browse button to display a dialog that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.

6. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity or quantities will be plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the project tree.When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

Related Topics

Sweeping a Variable

Working with Traces

Add Trace Characteristics

Creating 2D Polar Plots

In HFSS, a polar plot is a 2D circular chart divided by the spherical coordinates R and theta, where R is the radius, or distance from the origin, and theta is the angle from the x-axis. Following is the general procedure for drawing a polar graph of results:

1. On the **Results** menu (HFSS menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Polar plot** from the report type menu.

   The **Report** dialog appears.

2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.

   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.

   b. Domain field with a drop down selection list. Whether this field appears, and the domains listed depend on the Solution type and the **<*type*>** selected. For modal and terminal solution data reports, the domain can be **Sweep** or **Time**.

      Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.

   c. Geometry field with a drop down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.

3. In the **Trace** tab **Polar Component** area, specify the information to plot:

a. On the **Category** drop down list, click the type of information to plot.

b. On the **Quantity** list, click the values to plot. Use CTRL-click to make multiple selections.

c. In the **Function** list, click the mathematical function to apply to the quantity for the plot.

d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note Color shows valid expression.

e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.

4. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity will be plotted against the swept variable values or quantities you specified on an x-y graph. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

5. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

Related Topics

Reviewing 2D Polar Plots

Sweeping a Variable

Working with Traces

Add Trace Characteristics

Reviewing 2D Polar Plots

For a polar plot of S-parameters, HFSS displays in the lower-left corner the following derived information about the cursor's location:

MP The magnitude and phase of the point.

RX The normalized resistance (**R**) and reactance (**X**).

GB An alternate view of the normalized resistance and reactance in the form of

$$R + jX = \frac{1}{G + jB}$$

where

- ***G*** = conductance
- ***B*** = susceptance

Q The quality factor.

VSWR The voltage standing wave ratio, calculated from the equation $\frac{1 + |S_{ij}|}{1 - |S_{ij}|}$.

A scale below the plot displays the scale of points along the R-axis.

Related Topics

Creating 2D Polar Plots

Creating 3D Polar Plots

A 3D polar plot is a 3D circular chart divided by the spherical coordinates R, theta, and phi, where R is the radius, or distance from the origin, theta is the angle from the x-axis, and phi is the angle from the origin in the z direction. Once created, you can also overlay the 3D polar plot on the model window by using the **HFSS>Fields>Plot Fields>Radiation** command, or by right-clicking on Field Overlays in the Project tree and selecting **Plot Fields>Radiation Field**. Following is the general procedure for drawing a 3D polar plot of results:

1. On the **Results** menu (HFSS menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **3D Polar plot** from the report type menu.

   The **Report** dialog appears.
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
   b. Geometry field with a drop down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.
3. In the **Trace** tab **Mag** area, specify the information to plot along the R-axis, or the axis measuring magnitude:

a. On the **Category** drop down list, click the type of information to plot.

b. On the **Quantity** list, click the values to plot. Use CTRL-click to make multiple selections.

c. In the **Function** list, click the mathematical function to apply to the quantity for the plot.

d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note Color shows valid expression.

e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.

4. On the **Trace** tab **Theta (Secondary Sweep)** line, select the sweep variable from the drop down list and specify all values or select values to plot along the theta-axis:
5. On the **Trace** tab **Phi (Primary Sweep)** line, select the sweep variable from the drop down list, and specify all values or select values to plot along the phi-axis:
6. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity or quantities will be plotted against the R-, phi-, and theta-axes on a 3D polar graph. The plot is listed under **Results** in the project tree. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.
7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

Related Topics

Sweeping a Variable

Working with Traces

Add Trace Characteristics

Creating Smith Charts

A Smith chart is a 2D polar plot of S-parameters upon which a normalized impedance grid has been superimposed. Following is the general procedure for creating a Smith chart of results:

1. On the **Results** menu (HFSS menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Smith Chart** from the report type menu.

   The **Report** dialog appears.
2. In the **Trace** tab **Polar Component** area, specify the information to plot:

   a. On the **Category** drop down list, click the type of information to plot.

   b. On the **Quantity** list, click the values to plot. Use CTRL-click to make multiple selec-

tions.

c. In the **Function** list, click the mathematical function to apply to the quantity for the plot.

d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note Color shows valid expression.

e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.

3. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity will be plotted against the values you specified on a polar plot. In addition, each circle on the plot is labeled with values of *R*, measuring normalized resistance, and each line is labeled with values of *X*, measuring normalized reactance. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

4. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

Related Topics

Reviewing 2D Polar Plots

Sweeping a Variable

Working with Traces

Add Trace Characteristics

Creating Smith Contour Charts

A Smith contour chart is a polar plot of S-parameters upon which a normalized impedance grid has been superimposed. Following is the general procedure for creating a Smith chart of results:

1. On the **Results** menu (HFSS menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Smith Chart** from the report type menu.

   The **Report** dialog appears.

2. In the **Trace** tab **Mag** area, specify the information to plot:

   a. On the **Category** drop down list, click the type of information to plot.

   b. On the **Quantity** list, click the values to plot. Use CTRL-click to make multiple selections.

   c. In the **Function** list, click the mathematical function to apply to the quantity for the plot.

   d. The **Value** field displays the currently specified Quantity and Function. You can edit this

field directly.

Note Color shows valid expression.

e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.

3. On the **Trace** tab **(Secondary Sweep)** line, select the sweep variable from the drop down list and specify all values or select values to plot along the theta-axis:

   To select an Secondary sweep component that is different than the default, uncheck the Default field to enable the X field and browse [...] button. Click the browse [...] button to display the **Select X Component** dialog. This lets you specify the X component as you do the Y; that is, in terms of Categories which define the selectable Quantities, and Functions to apply. After making selections, **OK** the dialog to assign the X component.

   a. If sweeps are available, you can select the browse [...] button to display a dialog that lets you select particular sweep or sweeps, or all sweeps.

   b. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (for example, for a parametric sweep). If so, the variables other than the one chosen as the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the value, and an Edit column with an ellipsis [**...**] button. See Using Families tab for Reports.

4. On the **Trace** tab **(Primary Sweep)** line, select the sweep variable from the drop down list, and specify all values or select values to plot along the phi-axis:

   To select an X component that is different than the default, uncheck the Default field to enable the X field and browse [...] button. Click the browse [...] button to display the **Select X Component** dialog. This lets you specify the X component as you do the Y; that is, in terms of Categories which define the selectable Quantities, and Functions to apply. After making selections, **OK** the dialog to assign the X component.

   a. If sweeps are available, you can select the browse [...] button to display a dialog that lets you select particular sweep or sweeps, or all sweeps.

   b. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (for example, for a parametric sweep). If so, the variables other than the one chosen as the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the value, and an Edit column with an ellipsis [**...**] button. See Using Families tab for Reports.

5. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity will be plotted against the values you specified on a polar plot. In addition, each circle on the plot is labeled with values of *R*, measuring normalized resistance, and each line is labeled with values of *X*, measuring normalized reactance. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you

select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

6. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

Related Topics

Reviewing 2D Polar Plots

Sweeping a Variable

Working with Traces

Add Trace Characteristics

Creating Data Tables

A data table is a grid with rows and columns that displays, in numeric form, selected quantities against a swept variable or other quantities.

1. Click **HFSS**>**Results**>**Create <*type*> Report**, or right click on the Results icon in the Project tree and click **Create <*type*> Report**.
2. In the display type menu, click **Data Table**.

   The **Report** dialog box appears.
3. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
   b. Domain field with a drop down selection list. Whether this field appears, and the domains listed depend on the Solution type and the **<*type*>** selected. For modal and terminal solution data reports, the domain can be **Sweep** or **Time**.

      Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.
   c. Geometry field with a drop down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.
4. Under the **Trace** tab **Y** component section, select the quantity you are interested in and its associated function:
   a. On the **Category** drop down list, click the type of information to plot.
   b. On the **Quantity** list, click the values to plot. Use CTRL-click to make multiple selections.
   c. In the **Function** list, click the mathematical function to apply to the quantity for the plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this

field directly.

Note Color shows valid expression.

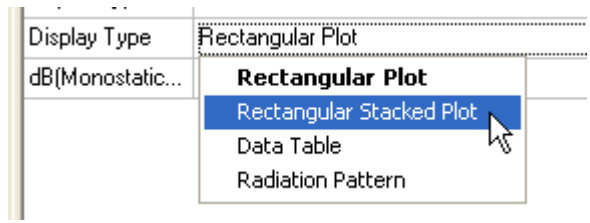
e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.

5. On the **Trace** tab **X (Primary sweep)** line, select the sweep variable from the drop down list, and specify all values or select values.
6. Click **New Report**.

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The Y quantity will be listed at each variable value or additional quantity value you specified. The data table is listed under **Results** in the project tree. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.
7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

If you choose to print a data table:

- Selecting print "All" prints the whole table for current data page (if there are more than one data page)
- Selecting print "Pages" prints user specified pages
- If the table is bigger than the screen view (that is, it has scroll bar), printing first scrolls right, prints until no more scrolling and then scroll down.
- The Page number appears at the bottom of the page, aligned at center
- The table layout of each page follows the screen, but with no scroll bar will be printed, and no data page bar as on screen.

Related Topics

Sweeping a Variable

Working with Traces

Add Trace Characteristics

Creating Radiation Patterns

A radiation pattern is a 2D polar plot displaying the intensity of near- or far-field radiation patterns. It is divided by the spherical coordinates R and theta, where R is the radius, or distance from the origin, and theta is the angle from the x-axis. Following is the general procedure for drawing a radiation pattern of results:

1. Click **HFSS**>**Results**>**Create <*type*> Report**, or right click on the Results icon in the Project tree and click **Create <*type*> Report**.
2. In the display type menu, click **Radiation Pattern**.

   The **Report** dialog box appears, and a Radiation Pattern Plots icon appears under Results in the Project tree.
3. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
4. In the **Trace** tab **Mag Component** area, specify the information to plot along the R-axis, or the axis measuring magnitude:
   a. On the **Category** drop down list, click the type of information to plot.
   b. On the **Quantity** list, click the values to plot. Use CTRL-click to make multiple selections.
   c. In the **Function** list, click the mathematical function to apply to the quantity for the plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.
5. In the **Trace** tab **Ang (Primary sweep)** line, specify the sweep variable from the drop down list, and specify all values or select values.
6. Click **New Report.**

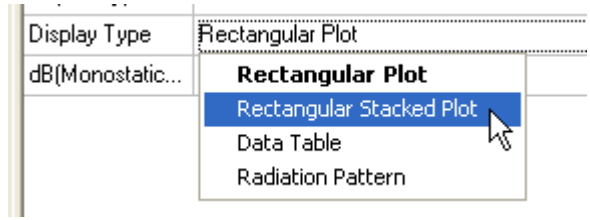
   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity or quantities will be plotted against the values you specified on a 2D polar plot. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the

Properties window. These properties can be edited directly to modify the plot.

7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Related Topics

Sweeping a Variable in a Report

Working with Traces

Add Trace Characteristics

Delta Markers in 2D Reports

To view the difference between any two marker points in a report:

1. Set the first marker by left-clicking and holding the mouse button.
2. Move the mouse without releasing left button to another position, and then release the left button to create second marker.

In the marker text window, you see the difference between the two markers instead of the X, Y value of marker.

Related Topics

Setting Report2D options

Working with Traces

Editing the Display Properties of Traces

Adding Data Markers to Traces

Plotting in the Time Domain

The idea behind Time-Domain Reflectometry (TDR) is to excite a structure with a step function, and inspect the reflections as a function of time. Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal or Transient). You can then select **Time** from the **Domain** list in the **Report** dialog. You also need to specify the input signal, whether step or impulse.

With **Time** selected as the domain, you can select from several Categories and associated Quantities to plot, for example mag(S_{11}). When you plot in the Time domain, every frequency domain

quantity is first converted to the time-domain before the formula is evaluated. For example, if you type in

$S_{11} / (1 - S_{11})$

and plot it in the time domain the reporter will plot

$IFFT(S_{11} * input) / (1 - IFFT(S_{11} * input))$

It will NOT plot

$IFFT(S_{11}/ (1 - S_{11}) * input)$

The two expressions are not equivalent.

If you select Time Domain Impedance as the Category, you can select the TDRZ quantity. This is defined as

$TDRZ(t) = Zref * (1 + IFFT(S_{11} * input)) / (1 - IFFT(S_{11} * input))$

where "input" denotes the Fourier transform of the input signal (step or impulse) and "IFFT(.)" denotes the inverse FFT.

This equation is the instantaneous ratio of the time-domain voltage v(t) to the time-domain current i(t). That is because voltage and current are defined (in the frequency domain) in terms of the incident and reflected waves a and b, respectively, as

$V = sqrt(Zo) * (a + b) = sqrt(Zo) * (1 + S_{ii}) * a$

$I = 1/sqrt(Zo) * (a - b) = 1/sqrt(Zo) * (1 - S_{ii}) * a$

This lets the incident wave be the input step signal, and so when we take the inverse FFT of V and I, we get v(t) and i(t) in the time domain. Taking their ratio as a function of time then yields TDRZ(t). By default, Zo is equal to 50 Ohm.

To create a plot in the **Time Domain**:

1. For a design with an existing sweep setup, follow steps 1 - 4 for creating a report for design.
2. In the **Report** dialog box, in the **Domain** list, click **Time**.

   This enables the **TDR Options** button and for terminal solution data reports includes the Terminal TDR Impedance in the Category list.
3. Click the **TDR Options** button.

   The **TDR Options** dialog box appears.
4. Select the input signal type, **Step** or **Impulse**.

   A **Step** describes a sustained change in the signal, whereas the **Impulse** is a brief excitation. **Impulse** is a very narrow rectangular pulse, with zero rise and fall time, width of 1 time step, and height of 1/(time step).

   Selecting **Step** enables the **Rise Time** field, and **Impulse** disables it.
5. If you selected **Step**, enter the rise time of the pulse in the **Rise Time** text box.

   The rise time should be appropriate for the frequency context.

   With a band width from DC to f_{max}, the best time resolution that can be achieved is $1/(2f_{max})$.

   A rise time of $1/(2f_{max})$ is the shortest rise time that can be resolved. However, a rise time of 0

s gives equally valuable information, so 0 is the default in this panel. See the example plot.

6. Enter the total time on the plot in the **Maximum Plot Time** text box.

   The default maximum plot time in the **TDR Options** dialog is related to the delta frequency df in the frequency sweep: it is 1/2df, since that is the extent of time for which the IFFT gives information. This is often very long relative to the time delay that corresponds to the length of your device under test, so you may want to reduce this value. Alternatively, you can adjust the time axis of your TDR plot after it has been created.

7. Set the number of time points to plot in the **Delta Time** text box. By default, this is set to the number of points in the frequency sweep.

   The delta time is based on the bandwidth of the sweep: with a frequency sweep from DC to f_{max}, the smallest time resolution you can obtain is given by $1/(2f_{max})$. The IFFT algorithm provides data points as a spacing of $1/(2f_{max})$, but you can smoothly interpolate between points by setting a finer resolution, e.g. to $1/(10f_{max})$, at the expense of extra computation time.

8. Optionally, under **TDR Window**, modify the window type and width.

9. You can use the **Save as Default** to set the current values as a default, and the **Use Defaults** button to use previously saved options. Note that when you select a trace, the initial displayed values are those of the selected trace.

10. Click **OK**.

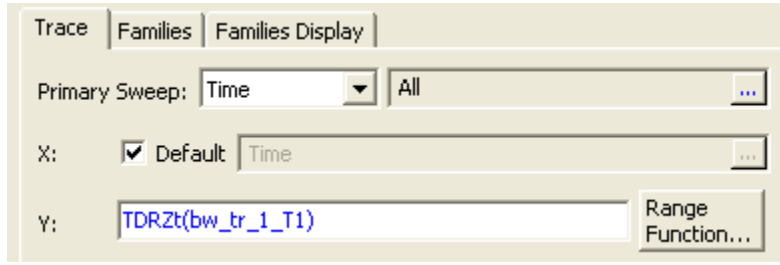
    Optionally, to plot Terminal TDR impedance (that is, rather than calculate the S-parameter for waveport1 versus frequency, instead calculate the delay versus time at a particular impedance), do the following:

    a. In the **Category** list, click **Terminal TDR Impedance**.

    b. In the **Quantity** list, click a quantity to plot.

       The default impedance (Zo) for the TDRZ quantity is 50 Ohms, unless you specified differently when you Set Renormalizing Impedance for Terminals when you created the terminals in the model. If you need a different impedance value, you can either edit the value in the **Report** dialog (as shown below), or you can create an Output Variable representing $Zo \times (1+S_{ii})/(1-S_{ii})$ with the Zo of your choice. To edit the Zo value in the **Report** dialog:

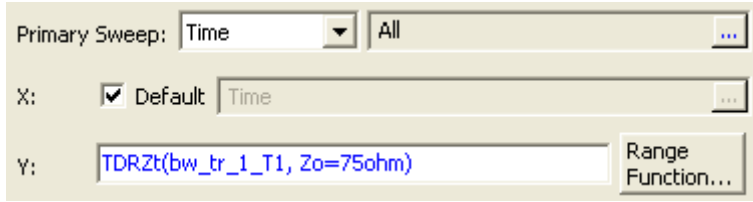
       1. For the Category, select Terminal TDR Impedance, and the Port and Function of interest.



       2. Edit the value by placing the cursor in the Value field.

          In this example, the value for Zo is changed from the default to 75 Ohms by typing

',Zo=75ohm' in the Y-column field.



c. In the **Function** list, click the mathematical function of the quantity to plot.

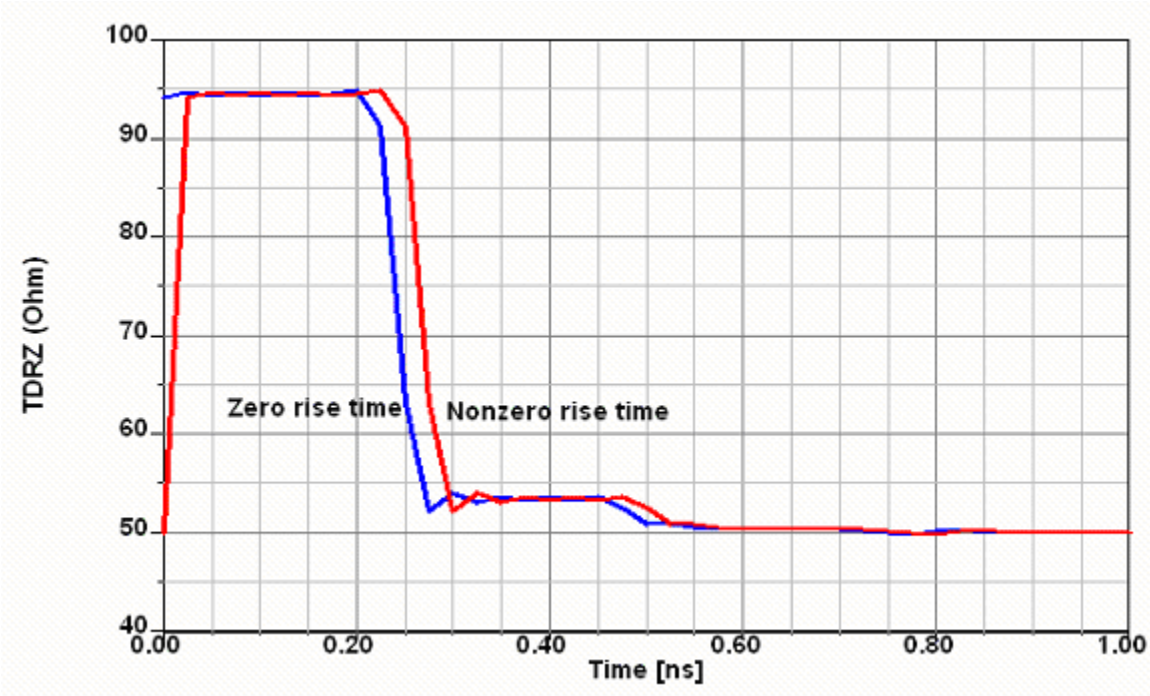
3. Click **Done**.

The report appears in the view window. It will be listed in the project tree.

If $S_{11} = 0$ at DC, the time-domain step response will settle to zero and the TDRZ step response settles to Zref. If S_{11} is nonzero at DC, the time-domain step response will settle to a nonzero value and TDRZ will settle to a value different from Zref. The time-domain impulse response will always settle to zero, since it can be seen as the derivative of the step response. The TDRZ impulse response will always settle to Zref.

The plot below shows the difference between a short nonzero rise time and zero rise time for a transmission line segment of 94 Ohm. Note that the trace with zero rise time starts at the correct line impedance while the other starts at the renormalizing impedance. Other than that, one trace is a shifted version of the other. The reason the plot with finite rise time starts at 50 ohms is that the time-domain voltage and current are still at their steady state values, so v = Zref * i. As the pulse arrives, the TDRZ response changes from the steady-state behavior because there's a reflection from the transmission line back to the exciting source, which has a different

renormalizing impedance from the characteristic impedance of the transmission line.



Some things to keep in mind with TDR:

$$Spatial resolution \Delta x = c/(2B) \quad (1)$$

where c is the speed of light in the medium and B is the bandwidth of the signal. Since TDR is usually based on a frequency band that starts at DC, the spatial resolution becomes

$$\Delta x = c/(2F_{max}) \quad (2)$$

where F_{max} is the highest frequency in the frequency sweep. For example, if F_{max} = 15 GHz and the medium has ε_r=4, the spatial resolution will be (1.5E8 m/s)/(3E10 /s) = 5 mm.

A spatial resolution of $c/(2F_{max})$ corresponds to a resolution in time

$$\Delta t = 1/(2F_{max}) \quad (3)$$

Let N be the number of points in the IFFT. N equals the number of time samples, and it also equals twice the number of frequency samples. The density of frequency samples in the frequency sweep influences the total time T as follows:

$$2F_{max}/(\Delta f) = N(\text{number of points in IFFT}) = T/\Delta t \quad (4)$$

So increasing the density of the frequency samples leads to an increase in total time T. In practical case, this often leads to a long tail in the TDR plot with little useful information. Therefore, the TDR Options interface lets you set the maximum plot time to a smaller value.

The TDR Options interface also lets you choose a smaller Δt than given by equation (3) above. When you choose a smaller Δt, you increase F_{max} by "zero padding", i.e. adding zero values for S_{11} beyond the calculated frequency sweep. Whether this is justified depends on your judgment. It leads in practice to a smoother TDR signal.

HFSS also lets you set the rise time of your input signal. The rise time should be at least $1/(2F_{max})$. Even this rise time is a bit short for comfort, as it equals the duration of only one time sample. An input signal with a longer rise time has a smaller high-frequency content and will lead to reduced "ringing" in the TDR response.

A Hamming or Hann filter will also reduce the high-frequency content and tends to lead to a smoother TDR response. With these filters, one can select a width. A width of 100% is often a good choice.

Related Topics

Interpolating sweep for a driven solution (Modal or Terminal or Transient).

Creating a report for design

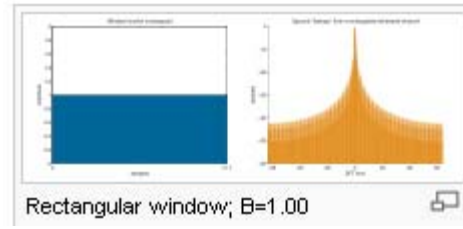
TDR Windowing Functions

Perform TDR on Report

TDR Windowing Functions

Windowing functions cause the FFT of the signal to have non-zero values away from ω. Each window function trades off the ability to resolve comparable signals and frequencies versus the ability to resolve signals of different strengths and frequencies. The window type list includes:

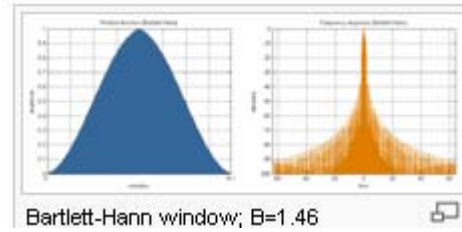
Window Function	Preferred Use
Rectangular	A low dynamic range function offering good resolution for signals of comparable strength. Poor when signals have very different amplitudes. $w(n)=1$.



Bartlett — A high dynamic range function, with lower resolution, designed for wide band applications.

$$w(n) = a_0 - a_1\left|\frac{n}{N-1} - \frac{1}{2}\right| - a_2\cos\left(\frac{2\pi n}{N-1}\right)$$

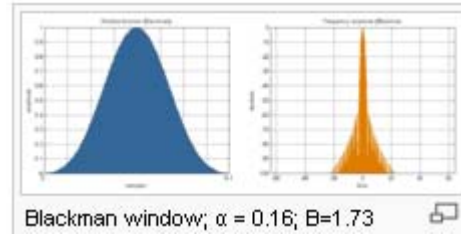
where a_0=0.62; a_1=0.48; a_2=0.38

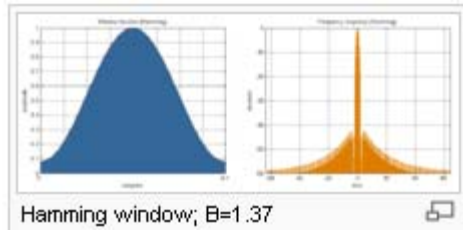


Blackman — A high dynamic range function, with lower resolution, designed for wide band applications.

$$w(n) = a_0 - a_1\cos\left(\frac{2\pi n}{N-1}\right) + a_2\cos\left(\frac{4\pi n}{N-1}\right)$$

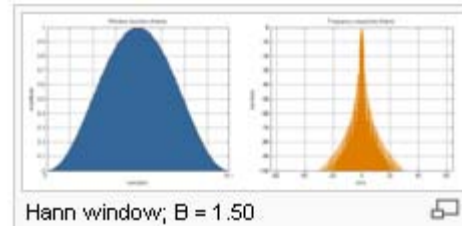
where a_0 =(1-α)/2; α_1=1/2; α_2=α/2



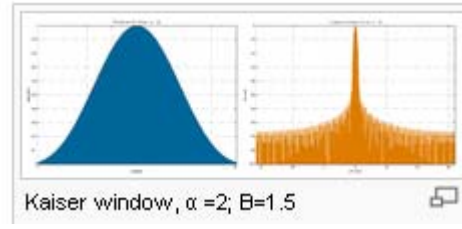
Window Function	Preferred Use
Hamming	A moderate dynamic range function, designed for narrow band applications. It minimizes the maximum sidelobe. $$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$$ 

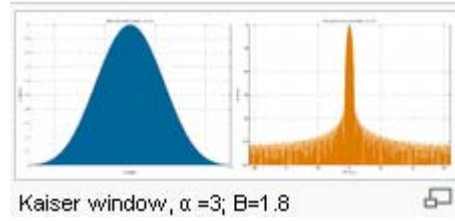
Window Function	Preferred Use
Hanning (default)	A moderate dynamic range function, designed for narrow band applications.

$$w(n) = 0.5\left(1-\cos\frac{2\pi n}{N-1}\right)$$



Kaiser	Selecting the Kaiser plot also enables a field to specify an associated Kaiser parameter. The larger the Kaiser parameter, the wider the window. The parameter controls the trade off between width of the central lobe and the area of the side lobes.





Window Function	Preferred Use
Welch	This approach applies a parabola-shaped window to the frequency domain data. It is based on the Bartlett method but splits the signal into overlapping segments, which are then windowed. The intent is to balance the influence of data in the center of the function.

You can use the **Save as Default** to set the current values as a default, and the **Use Defaults**

Working with Traces

A trace in a 2D or 3D report defines one or more curves on a graph. A trace in a data table defines part of the displayed matrix of text values.

The values used for a plot's axes (which may be X, Y, Z, phi, theta, or R depending on the display type) can be variables in the design, such as frequency, or functions and expressions based on the design's solutions. If you have solved one or more variables at several values, you can "sweep" over some or all of those values, resulting in a curve in 2D or 3D space.

A report can include any number of traces and, for rectangular graphs, up to four independent y-axes. Traces appear in the Project tree under their report. They can be selected, copied and pasted.

When you move a cursor over a trace in a report, the cursor changes to show that you can make a selection:

- For PC systems, the cursor changes to the color of the selectable trace.
- For Unix systems, the cursor changes to a solid black arrow, rather than the default black outline.

In general, to add a trace to a report:

1. Select a report in the Project window and right-click and select **Modify Report.**
2. In the **Report** dialog specify the Y component information.
   a. Specify the Category of information you want to plot from the drop down menu.

      The Category drop down menu lists the available categories for the Solution type and the current design. Selecting a category changes the Quantity and Function lists to represent what is available for that category.

   b. Specify the Quantity you want to plot by selecting from the Quantity list.

      The selected quantity appears in the Value field, operated on any selected function.

   c. Select the Function to apply to the specified quantity.

   d. The Value field shows the trace being readied for plotting on the Y-axis. This field is editable when the text cursor is present. You can modify the information to be plotted by typing the name of the quantity or sweep variable to plot along an axis directly in the text boxes.

Note Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog. This applies currently specified Quantity and Function.

3. In the **Report** dialog specify the X axis information (for example Primary Sweep).
4. Click **Add Trace**.

   A trace is added to the traces list under its report icon in the Project tree. The trace represents the function of the quantity you selected and will be plotted against other quantities or swept variable values. Selecting a Trace in the Project tree displays the Properties window for that Trace. Selecting a trace in the report or legend displays the display Properties window for that trace.

   Trace icons can be selected, copied, and pasted for their definitions or their data. They can be selected and deleted from the Project tree.

   By the default, the Trace name is the definition (the category, quantity and function). The trace will be visible in the report when you click **Add Trace**.

   Trace properties can be edited directly in the respective Properties windows or edited in the **Report** dialog. To change the name or definition of a trace, see Editing Trace Properties. To edit other display properties of a trace, see Editing the Display Properties of Traces

Related Topics

Removing Traces

Editing Trace Properties

Editing the Display Properties of Traces

Discarding Report Values Below a Specified Threshold

Add Trace Characteristics

Adding Data Markers to Traces

Setting Report2D options

Copy and Paste of Report and Trace Definitions

Copy and Paste of Report and Trace Data

Delta Markers in 2D Reports

Editing Trace Properties

To edit trace properties such as the name, the component definition, or the context, or the variables select the trace in the Project tree.

To edit a **trace name**:

1. Select the trace in the Project tree.

   This displays a docked Properties window for the Trace.
2. Check the Specify Name box.

   This enables editing of either the Name field in the docked properties dialog, or the Trace label text in the Project tree. Editing this name changes the display in the Legend and in the Project

tree, but not the underlying Y-component definition.

Note To control the display of the Solution Name and Variation Key in the Legend, see Report 2D: Legend Tab.

To edit a **trace component definition**:

1. Select the trace in the Project tree.
2. In the docked Properties window for the trace, select the component field of interest, and select Edit... from the drop down menu.

   This displays the an edit Component field window.form which you can edit the category, quantity and function.
3. Click OK to apply the changes and close the dialog.

To edit a **trace Context**:

1. Select the trace in the Project tree to display the docked properties window.
2. In properties window, click the Solution field or the Domain field. If other selections are possible, they can be selected from the drop down menu.

To edit a **variable** for a trace:

1. Select the trace in the Project tree to display the docked properties window.
2. Under the -Variables category, on the Families line, click the Edit button to display the Edit families dialog.

   From this dialog, you can select the Sweeps or Variations radio buttons. Each selection changes the

   If other nominal values are available you can click the ellipsis button to select from a list.

Related Topics

Removing Traces

Editing the Display Properties of Traces

Discarding Report Values Below a Specified Threshold

Add Trace Characteristics

Adding Data Markers to Traces

Setting Report2D options

Copy and Paste of Report and Trace Definitions

Copy and Paste of Report and Trace Data

Delta Markers in 2D Reports

Editing the Display Properties of Traces

To edit the display properties of a trace:

1. Select a trace in an open **Report** window.
2. Click once on the trace to view a **Docked Properties** window, or double click to open **Proper-**

ties window.

The display properties window for a trace includes a **General** tab and an **Attributes** tab.

The General tab properties apply to the general appearance of the plot. They include the Background color, Contrast color, Field width, and Whether to use Scientific notation for marker and delta marker displays. (X and Y notation display is set separately, in the Axis property tabs.)

The Attributes Tab properties apply specifically to the Trace. The defaults are set in the Report2D options. They include:

- Name -- not editable by selecting the trace from the Report. It shows the characteristics of the trace as defined in the **Report** dialog.

  To edit a trace name, see Editing Trace Properties
- Color -- shows the Trace color. Double click to open a Color dialog. You can select from Basic colors, or custom colors. You can define up to 16 custom colors by selecting or by editing the Hue, Saturation, Luminescence, and the Red, Green, and Blue values.
- Line style -- a drop down menu lets you select Solid, Dot, Dash, or Dot-dash.
- Line width -- a text field lets you edit the numeric value.
- Trace type -- the drop down menu contains entries for Continuous, Discrete, Bar-Zero, Bar Infinity, Stick Zero, Stick Infinity, Histogram, Step, and Stair.
- Show Symbol -- whether to show a symbol at the data points on the line.
- Symbol Frequency -- how often to show symbols on the trace.
- Symbol Style -- use a drop down menu to select from box, circle, vertical ellipse, horizontal ellipse, vertical up triangle, vertical down triangle, horizontal left triangle, horizontal right triangle
- Fill Symbol -- use the check box to set the symbol display as a solid or as hollow.
- Symbol Arrows -- use the check box to use arrows on the curve ends

.

Note So that curves with single points always appear, Box is the default symbol. For HFSS 11, None cannot be selected as the symbol.

3. Edit the properties of interest and OK the Properties window to apply the changes and close the window.

Related Topics

Setting Report2D options

Working with Traces

Editing Trace Properties

Add Trace Characteristics

Adding Data Markers to Traces

Removing Traces

Discarding Report Values Below a Specified Threshold
Copy and Paste of Report and Trace Definitions

Adding Data Markers to Traces

The Reporter includes **Report 2D>Marker>** menu commands and toolbar icons

that let you add markers to traces. A marker appears as "m*N*" at the marked point, where *N* increments from 1 as you place additional markers. Each marker can be selected and has editable properties including name, font, background and color. As you place markers, one or more marker legends may be displayed, depending on the **View>Active View Visibility** settings for the legends. The main marker legend appears in the upper left of the plot, and lists the marker names and their X and Y values in a table. You can control the number format for the table values via the properties window, general tab. Under Marker/Other Number format, you can specify field width, precision, and whether to use scientific notation. This value is independent of the Axis tab number properties. A separate marker legend appears for Delta Markers, as described for the **Delta Marker** command.

When you enter Marker mode, the cursor arrow is accompanied by an "m" while a circle on the selected trace shows the current position for a potential marker.

To end Marker mode, right-click to display the shortcut menu, and select **End Marker Mode**.

The available Marker mode commands and associated icons are the following:

- **Marker** -- this command lets you place a marker at an arbitrary point on a selected trace.
- **X Marker** -- this command adds a movable marker at the origin of the plot with a vertical line rising from the X axis. To move an X marker, click on the X label and drag it to the desired location. The label at the bottom of the line gives the X coordinate, and flag on the vertical line identifies the Y coordinate on the trace. A trace property lets you lock the drag feature to leave the marker in place. This marker is not cleared by the **Clear All** command, and must be deleted by selecting it and using the **Edit Delete** command.
- **Maximum** -- places a marker at the Maximum value on the selected trace.
- **Minimum** -- places a marker at the Minimum value on the selected trace.
- **Delta Marker** enters delta marker mode, placing a circle on the selected trace. Clicking on the trace sets an initial point and subsequent clicks on arbitrary points on the trace place additional markers until you leave marker mode. These markers have their own legend, which includes the following information for each pair of markers specified.:

Name	Delta(X)	Delta(Y)	Slope(Y)	InvSlope(Y)
d(m2,m3)	0.4700	1.8319	3.8976	0.2566

- **Next Peak** -- moves a selected marker on the next peak on a trace. You must exit marker mode and select a marker to enable this command.
- **Next Minimum** -- moves a selected marker to the next minimum on a selected trace. You must exit marker mode and select a marker to enable this command.

- **Previous Peak**  -- moves a selected marker on the previous peak on a selected trace. You must exit marker mode and select a marker to enable this command.
- **Previous Minimum**  -- places a marker on the previous minimum on a selected trace. You must exit marker mode and select a marker to enable this command.
- **Next Data Point** (Right) -- moves a selected X marker to the next data point.
- **Previous Data Point** (Left) -- moves a selected X marker to the previous data point.
- **Next Curve** -- selects the next curve in the report, based on the order in the trace legend.
- **Previous Curve** -- selects the previous curve in the report, based on the order in the trace legend.
- **Clear All** -- clears all markers on a report except X Markers.

Related Topics

Setting Report2D options

Working with Traces

Add Trace Characteristics

Removing Traces

Discarding Report Values Below a Specified Threshold

Modifying the Legend in a Report

Editing the Display Properties of Traces

Zoom in or out.

Fit contents in the view window.

Showing Objects

Hiding Objects

Delta Markers in 2D Reports

Discarding Report Values Below a Specified Threshold

To prevent real small numbers from skewing a plot, you can discard small values (below a specifiable threshold).

1. Double-click on the X or Y axis of interest on an open plot display.

   This opens the **Properties** window for the Axis
2. Under the **Axis** tab, use the scroll bar to find the **Specify Discard Values** property.
3. Click the checkbox to enable the property.
4. Enter a value in the **Discard Below** field. Units specified elsewhere in the Axis property are applied to this value. The Discard Below text box is inactive if the Specify Discard Values checkbox is not enabled.
5. Click **OK** to apply the Discard Values to the report.

Related Topics

Working with Traces

Removing Traces

Editing the Display Properties of Traces

Modifying Background Properties of a Report

Modifying Reports

Add Trace Characteristics

Add Trace Characteristics

You can add or clear additional characteristics to a selected trace. To add additional characteristics to a selected trace:

1. Select a trace in a report plot or legend.
2. Click **Report 2D>Trace Characteristics**, or right-click on the selected trace to display the short cut menu.
3. Select **Trace Characteristics>Add...**.

   This displays the **Add Trace Characteristics** dialog.
4. Select the **Category**, and then an associated Function to apply. The available categories depend on the plot, and Category enables the display of associated functions.

Category	Functions for the Category
Math	max, min, pk2pk, rms, avg, integ, integabs, avgabs, rmsAC, ripple, pkavg, XatYMin, XatYMax, XatYVal
PulseWidth	pulsefall9010, pulsefront9010, pulsefront3090, pulsemax, pulsemaxtime, pulsemin, pulsemintime, pulsetail50, pulsewidth5050, pw_plus, pw_plus_max, pw_plus_min, pw_plus_avg, pw_plus_rms, pw_minus_max, pw_minus_min, pw_minus_avg, pw_minus_rms
Overshoot, Undershoot	overshoot, undershoot.
TR & DC	crestfactor, formfactor, distortion, fundamentalmag, delaytime, risetime, deadtime, settlingtime,
Error	iae, ise, itae, itse
Period	per, pmax, pmin, prms
Radiation	xdb10bandwidth, xdb20bandwidth, lSidelobeX, lSidelobeY, rSidelobeX, rSidelobeY

   Given a selected Function, and Category, the **Add Trace** dialog displays a text field that explains the Purpose of the function. For a full list of functions and their definitions, see Selecting a Function.
5. Some categories and functions call for you to specify one or two additional values in a table.

You can save these values using the Default button.

6. Click the **Add** button to add the specified characteristics to the Trace.

To remove existing trace characteristics:

1. Select a trace in a report plot or legend.
2. Click **Report 2D>Trace Characteristics**, or right-click on the selected trace to display the short cut menu.
3. Select **Trace Characteristics>Clear All**

   Trace characteristics are clear from the selected trace.

Related Topics

Working with Traces

Selecting a Function

Adding Data Markers to Traces

Removing Traces

You can remove traces from the traces list in the following ways:

To *remove one trace* from the report:

- Select the trace you want to remove from the Project tree, and then click **Delete**.

To *remove all traces* from the report:

- Select all the traces and click **Delete**.

Related Topics

Working with Traces

Editing the Display Properties of Traces

Copy and Paste of Report and Trace Definitions

You can copy and paste report and individual trace definitions within a single design or across designs. The report or trace definition will be evaluated within the context of the target design or report.

Note If the report or trace definition contains properties that do not exist in the target design (for example, a port name) an error will be posted that indicates a solution does not exist for this trace

Note You must copy and paste trace definitions between the same report types. For example, you cannot copy a trace from a Modal Solution Data report and paste it in a Far Fields report.

To copy a Report Definition:

Right click on the report name in the project tree and select **Copy Definition** from the shortcut menu.

To paste the Report Definition:

Right click on Results in the project tree of the target design and select **Paste**.

A new report is created and it contains the copied definitions.

To copy an individual Trace Definition(s):

Right click on the trace or traces under a report name in the project tree and select **Copy Definition**.

To paste the Trace Definition(s):

Right click on the report in the target design to which you would like to copy the trace or traces and select **Paste**.

A new trace(s) is added to the report and it contains the copied trace definition(s).

Note If you copy and paste a report or trace definition to a design which contains a definition with the same name, then an incremented number is appended to the pasted report or trace name.

Related Topics

Copying to the Clipboard as Images

Copy and Paste of Report and Trace Data

Copy and Paste of Report and Trace Data

You can copy and paste report and individual trace data within a single design or across designs. The report and trace definitions and all underlying data within the report or trace are copied and pasted to the target design or report.

To copy all data from a report:

Right click on the report name in the project tree and select **Copy Data**, or use the menu bar **Edit>Copy Data**, or right click within a plot to display a shortcut menu with **Copy Data**.

To paste copied report data:

Right click on Results in the project tree of the target design and select **Paste**.

To copy data from an individual trace(s) in a report:

Right click on the trace or traces under a report name in the project tree and select **Copy Data**.

To paste copied trace data:

Right click on the report in the target design to which you would like to copy the trace data and select **Paste**.

Note If you copy and paste report or trace data which contains the same name definition as a report or trace in the target design then an incremented number will be appended to the pasted name.

Related Topics

Copying to the Clipboard as Images

Copy and Paste of Report and Trace Definitions

Sweeping a Variable in a Report

In HFSS, a swept variable is a variable that typically has more than one value. You can plot any calculated or derived quantity against one or more of the swept variable's values.

For large projects or projects with many variables, you may obtain faster post processing before generating a solution by selecting which variables function as Sweep variables. Only the variables with Sweep enabled are indexed for post processing. See Adding a Design Variable and Adding a Project variable.

To specify the swept variable values to plot a selected quantity against:

1. In the **Report** dialog, select the variable from the X (Primary Sweep) pulldown menu.
2. To modify the values that will be plotted for a variable:
   a. Click the ellipsis **[...]** button on the **X (Primary Sweep)** line of the **Report** dialog to displays a popup list of the possible values.
   b. Select **All Values** or click the Edited button to display a dialog that lets you specify the sweeps to use.

      All of the selected variable's values will be plotted.

Related Topics

Sweeping Values Across a Distance

Sweeping Values Across a Sphere

Sweeping Values Across a Distance

1. If you are plotting a field quantity along a line, define a polyline object in the problem region.

   If you are plotting a near-field quantity along a line, set up a near-field line.
2. In the **Report** dialog box, click the line geometry of interest in the **Geometry** list.
3. Specify the quantities you want to plot along the axes.
4. For the **X (Primary Sweep)**, select the **Distance** variable.

   The values at which the selected quantity or quantities will be plotted are listed to the right. By default, a post-processing polyline object is divided into 100 equally spaced points.

5. For Near field, to plot the selected quantity or quantities at every point on the line, select **All Values**.

   For Near field, to plot the selected quantity or quantities at specific points on the line, clear the **All Values** option, and then select the point values on which you want to plot.

Note All maximum near-field data calculated by HFSS is at their maximum over the selected line object; if you plot the parameter over a sweep of values, the parameter will have the same value at each point on the plot.

Related Topics

Sweeping a Variable in a Report

Sweeping Values Across a Sphere

1. Set up a near-field sphere or a far-field infinite sphere.
2. In the **Report** dialog box, click the sphere geometry of interest in the **Geometry** list.
3. For the **Sweeps** variable corresponding to **phi**, select the ellipsis [...] button.

   This displays a small dialog.
4. Clear the Use all values checkbox to enable selection and editing of the sweep values.

   All of the possible values for the phi variable are listed in the dialog. The values are the result of the range of phi you specified during the infinite sphere's setup. To modify the values of phi to be plotted across the sphere, do the following:

   a. Click **Edit Sweep**.

   b. Specify the following information:

      Step or Count Whether to sweep by steps, or by linear count, decade count, octave count, or exponential count.

      Start Value The point where the rotation of phi begins.

      End Value The point where the rotation of phi ends.

      Step or Count The number of values between the start value and the end value.

   c. Click **Update Values**, and then click **OK**.

      The values listed are updated to reflect the new number of points.
5. To plot the selected quantity or quantities at every value of phi, select **All Values**.

   To plot the selected quantity or quantities at specific values of phi, clear the **All Values** option, and then select the phi values at which you want to plot.
6. For the **Sweeps** variable corresponding to **theta**, follow steps 4 and 5 for modifying the values of theta, if necessary, and specifying the theta values at which to plot the selected quantity or

quantities.

Note All antenna parameters and maximum far-field data calculated by HFSS is at their maximum over the selected object; if you plot the parameter over a sweep of values, the parameter will have the same value at each point on the plot.

Selecting a Function for a Plot

The value of a quantity being plotted depends upon its mathematical function, which you select from the **Trace** tab **Function** list in the **Report** dialog box. The available, valid functions depend on the type of quantity (real or complex) that is being plotted. The function is applied to the quantity which is implicitly defined by all the swept and current variables. For example, "S(11)" is the value of the S-parameter for every swept combination of variables (e.g., "height", "frequency" and so forth). (A smaller set of functions appears for the Function list in the Output Variables dialog.)

These functions can also be applied to previously specified Quantities and Functions as **Range Functions** when using the Set Range Function dialog.

Some of these functions can operate along an entire curve. These are: deriv, min, max, integ, avg, rms, pk2pk, cang_deg and cang_rad. These functions have syntax as follows:

- deriv(*quantity*) implicitly implies derivative over the primary sweep
- deriv(*quantity*, *SweepVariable*) explicitly means derivative over the sweep variable specified in the second argument (such as "Freq").

You can select from the following functions in the **Trace** tab **Function** list or type them directly into the Yor X field, if necessary.

abs	Absolute value of the simulation quantity which results in a number that is always positive.
acos	Arc cosine i.e. the inverse function of a cosine.
acosh	Inverse hyperbolic arc cosine.
ang	Magnitude of an angle.
ang_deg	Angle (phase) of a complex number, cut at +/-180.
ang_rad	Angle in radians.
arg	Argument of a complex number. It is the angle the complex number makes with the positive x axis. Same as **ang_deg**.
asin	Arc sine i.e. inverse function of sine.
asinh	Inverse hyperbolic sine.
atan	Arc tangent i.e. the inverse function of a tan.
atanh	Inverse hyperbolic tan.
atan2	Two argument function. For non-0 x,y, the function returns the angle between the + x-axis and the given x,y coordinates.

avg	Returns the average of the values of the selected quantity. avg = (Area between the curve and the X-axis) / (X length of the curve)
avgabs	Returns the mean of the absolute value of the selected quantity.
bandwidth	Returns the 3dB bandwidth of the selected simulation quantity. For bandwidth, the calculation is based on 3dB below the maximum peak.
cang_deg	Cumulative angle (phase) of the first parameter (a complex number) in degrees, along the second parameter (typically sweep variable). Returns a double precision value cut at +/-180.
cang_rad	Cumulative angle of the first parameter in radians along a second parameter (typically a sweep variable). Returns a double precision value.
cmplx(*re*, *im*)	A complex number, where *re* is the real part and *im* is the imaginary part.
conjg	Conjugate of the complex number.
cos	Cosine.
cosh	Hyperbolic cosine.
crestfactor	Returns the crest factor (peak/RMS) for the selected quantity.
dB(x)	20*log10(|x|) to base 10.
dBc	Decibels relative to the carrier. It is the power ratio of the signal to a carrier signal. Gives the relative signal strength.
dBm(x)	10*log10(|x|) +30.
dBW(x)	10*log10(|x|).
dB10	10*log(|x|) to base 10.
dB10normalize	10*log [normalize(mag(x))].
dB20	20*log(x) to base 10.
dB20normalize	20*log [normalize(mag(x))].
deadtime	Obtains the latest time when the qtyl is within a tolerance of zero.
delaytime	Obtains the time from zero to 50% of the target point.
degel	Conversion from degrees electrical to seconds with respect to Hz.
deriv	Derivative of first parameter with respect to second parameter.
distortion	Returns the total distortion for the selected simulation quantity and an additional argument frequency, which is the frequency in Hz at which to calculate the fundamental RMS of the simulation quantity.
even	Returns 1 if integer part of the number is even; returns 0 otherwise.

exp	Exponential function (the natural anti-logarithm) of the simulation quantity.
formfactor	Returns the form factor (RMS/Mean Absolute Value) for the selected quantity.
fundamentalmag	Returns the RMS value of the fundamental frequency for the selected quantity, and an additional argument, Frequency, which specifies the fundamental frequency.
gaincrossover	Returns the gain crossover frequency (where the gain is 0 dB) of the selected simulation quantity in Hz.
gainmargin	Returns the gain margin in dB at the phase crossover frequency of the selected simulation quantity. It also requires a reference simulation quantity to which the measured quantity is compared and the AC magnitude and phase angle of the reference quantity. These are entered as the arguments Reference Channel, Base Source Magnitude, and Base Source Angle.
iae	Returns the integral of the absolute deviation of the selected quantity from a target value that is entered via the additional argument.
if	if(cond_exp,true_exp, false_exp).
im	Imaginary part of the complex number.
int	Truncated integer function.
integ	Integral of the selected quantity. Uses trapezoidal area.
integabs	Absolute value of integral.
ise	Returns the integral of the squared deviation of the selected quantity from a target value that is entered via an additional argument.
itae	Returns the time-weighted squared deviation of the selected quantity from a target value that is entered via an additional argument.
itse	Returns the time-weighted squared deviation of the selected quantity from a target value that is entered via an additional argument. To use this function, you need to open the Add Trace Characteristics dialog and select the Error category.
j0	Bessel function of the first kind (0^{th} order).
j1	Bessel function of the first kind (1^{st} order).
jn	Bessel function of the first kind (nth order).
ln	Natural logarithm.
log	Natural logarithm (same as ln).

log10	Logarithm base 10.
lowercutoff	Returns the lower 3dB frequency of the selected simulation channel in Hertz.
lsidelobeX	The 'x' value for the left side lobe: the next highest value to the left of the max value.
lsidelobeY	The 'y' value for the left side lobe: the next highest value to the left of the max value.
mag	Magnitude of the complex number.
max	Returns maximum value of the simulation quantity.
max_swp	Returns maximum value of a sweep.
max2	Maximum value of the two simulation quantities. For example, **max2(a,b)** will plot maximum of **a** and **b** for a particular instance.
mean	Returns the average in the set of quantities selected. mean = sum(all y-value) / (number of y-values)
min	Returns the minimum value of the simulation quantity.
min_swp	Returns the minimum value of a sweep.
min2	Minimum value of the two simulation quantities. For example, **min2(a,b)** will plot minimum of **a** and **b** for a particular instance.
mod	Returns the modulus or absolute value of the simulation quantity.
nint	Nearest integer.
none	Returns null value.
normalize	Divides each value within a trace by the maximum value of the trace. ex. normalize(mag(x)).
odd	Returns 1 if integer part of the number is odd; returns 0 otherwise.
overshoot	Calculates peak overshoot given a threshold value and number of evenly spaced points over entire time range.
peakgain	Returns the peak value of gain of the selected simulation quantity in dB.
peakgainfreq	Returns the frequency in Hz at which the peak gain of the selected simulation quantity occurs.
polar	Coverts the complex number in rectangular co-ordinates to polar co-ordinates.
per	Returns the period of a simulation quantity.
phasecrossover	Returns the phase crossover frequency, at which the phase is -180 degrees, in Hz for the selected simulation quantity.

phasemargin	Returns the phase angle in degrees at the gain crossover frequency of the selected simulation quantity.
pk2pk	Peak to peak. Difference between max and min of the first parameter over the second parameter. Returns the peak-to-peak value for the selected simulation quantity.
pkavg	Returns the ratio of the peak to peak-to-average for the selected quantity.
pmax	Maximum period of the selected simulation quantity.
pmin	Minimum period of the selected simulation quantity.
pow	Raises x to the power of y; pow(x,y).
prms	Period Root Mean Square.
pulsefall9010	Returns the pulse fall time of the selected quantity according to the 90%-10% estimate.
pulsefront1090	Returns the pulse front time of the selected quantity according to the 10%-90% estimate.
pulsefront3090	Returns the pulse front time of the selected quantity according to the 30%-90% estimate.
pulsemax	Returns the pulse maximum from the front and tail estimates for the selected quantity.
pulsemaxtime	Returns the time at which the maximum pulse value of the selected quantity is reached.
pulsemin	Returns the pulse minimum from the front and tail estimates for the selected quantity.
pulsemintime	Returns the time at which the minimum pulse value of the selected quantity is reached.
pulsetail50	Returns the pulse tail time of the selected quantity from the virtual peak to 50%.
pulsewidth5050	Returns the pulse width of the selected quantity as measured from the 50% points on the pulse front and pulse tail.
pwl	Piecewise Linear.
pwl_periodic	Piecewise Linear for periodic extrapolation on x.
pwlx	Piecewise Linear x with linear extrapolation on x.
pw_minus	Pulse width of the first negative pulse.
pw_minus_avg	Returns the average of the negative pulse width input stream.
pw_minus_max	Returns the maximum pulse width of the negative pulse of input stream.

pw_minus_min	Returns the minimum pulse width of the negative pulse of input stream.
pw_minus_rms	RMS of the negative pulse width input stream.
pw_plus	Pulse width of the first positive pulse.
pw_plus_avg	Average of the positive pulse width input stream.
pw_plus_max	Max. Pulse width of the positive pulse of input stream.
pw_plus_min	Min. Pulse width of the positive pulse of input stream.
pw_plus_rms	RMS of the positive pulse width input stream.
re	Real part of the complex number.
rect	Converts the complex number in polar to rectangular co-ordinates.
rem	Fractional part of the selected simulation quantity i.e. remainder.
ripple	Returns the ripple factor (AC RMS/Mean) for the selected quantity.
risetime	Obtains the time taken to go from 10% to 90% of target point.
rms	Returnns the root mean square value of the selected quantity.
rmsAC	Returns the AC RMS for the selected quantity.
root	nth root function.
rSidelobeX	Returns the X value of right side-lobe occurrence.
rSidelobeY	Returns the Y value of right side-lobe occurrence.
settlingtime	Returns the latest time at which the value of the selected simulation quantity fell outside its tolerance band. The target value of the quantity and the +/- bandwidth of the tolerance band are the additional args.
sgn	Sign extraction.
sin	Sine.
sinh	Hyperbolic sine.
slidingmean	Returns the moving average value of the selected simulation quantity (specified by the first argument). The average is calculated over a period (specified by the second argument).
slidingrms	Returns the moving RMS value of the selected simulation quantity (specified by the first argument). The RMS value is calculated over a period (specified by the second argument).
sqr	Square of the selected simulation quantity.
sqrt	Square root of the selected simulation quantity.
stddev	Returns the standard deviation of given values.
sum	Returns the sum of the given values.

tan	Tangent.
tanh	Hyperbolic tangent.
undershoot	Calculates peak undershoot given a threshold value and number of evenly spaced points over entire time range.
uppercutoff	Returns the upper 3dB frequency of the selected simulation channel in Hz.
variance	Calculates the variance of the given values.
XAtYMax	Threshold crossing time: report first time (x value) at which an output quantity crosses YMax.
XAtYMin	Threshold crossing time: report first time (x value) at which an output quantity crosses a user definable threshold.
XAtYVal	Returns the X value at the first occurrence of Y value.
XWidthAtYVal	Returns the X width between the first 2 occurrence of Y value.
xdb10beamdwidth	Width between left and right occurrences of values 'x' db10 from max. Takes 'x' as argument (3.0 default). To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.
xdb20beamwidth	Width between left and right occurrences of values 'x' db20 from max. Takes 'x' as argument (3.0 default). To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.
YAtXMax	Returns the X value at maximum value of Y.
YAtXMin	Returns the Y value at minimum value of X.
YAtXVal	Returns the Y value at the first occurrence of X value.
y0	Bessel function of the second kind (0^{th} order).
y1	Bessel function of the second kind (1^{st} order).
yn	Bessel function of the second kind (nth order).

Related Topics

Add Trace Characteristics

Set Range Function

Selecting Solution Quantities to Plot

When you create a report of Modal or Terminal solution data, each trace in the report includes a quantity that is plotted along an axis. The quantity being plotted can be a value that was calculated by HFSS such as S_{11}, a value from a calculated expression, or an intrinsic (inherent) variable value such as frequency or theta. The valid categories available depend on the type of quantity (real or complex) that is being plotted, the setup, the solution type, and the plot domain.

To select an S-parameter quantity to plot:

1. In the **Report** dialog box, **Trace** tab, select one of the following categories:

Variables	Intrinsic variables, such as frequency or theta, or user-defined project variables, such as the length of a quarter-wave transformer.
Output Variables	User defined expressions applied to derive quantities from the original field solution.
S-parameter	S-parameters from the S-matrix. For designs which include a Frequency Selective Surface (FSS)-referenced radiation boundary, S_{11} and S_{21} represent the extracted reflection and transmission coefficients, respectively.
Y-Parameter	Admittance matrix parameters computed from the S-parameters and port impedances.
Z-Parameter	Impedance matrix parameters computed from the S-parameters and port impedances.
VSWR	Voltage standing wave ratio, calculated from the equation. $$\frac{1 + \left\|S_{ij}\right\|}{1 - \left\|S_{ij}\right\|}$$
Gamma	Propagation constants for the S-parameters.
Port Z_o	Characteristic port impedances.
Lambda	Guided wavelength.
Epsilon	Effective permittivity.
Group Delay	Quantity calculated as rate of change of the total phase shift with respect to angular frequency, $$\frac{d(\phi)}{d(\omega)}$$
TDR Impedance	TDR (Time-Domain Reflectometry) impedance for non-terminal problems. The idea behind TDR is to excite a structure with a step function, and inspect the reflections as a function of time. If you select the Time Domain for the plot, the Category list includes the TDR Impedance and the TDR options button is enabled. Selecting the TDR Impedance category displays the (TDRZ) of every terminal or mode in the ports. The list of available Functions includes those that can operate on the TDRZ values.

Active S-parameter

Active S, Y, Z, VSWR is supported only for driven modal projects.

Given a driven model project with a total of N port modes, let a_k denote the complex excitation fro the k-th mode specified in the Edit Sources dialog. Also let S denote the computed N x N scattering matrix. If $a_m \neq 0$ define active-S_m, m = 1..., N by

$$active\text{-}S_m \equiv \sum_{n=1}^{N} S_{mn} \frac{a_n}{a_m} \qquad m = 1, ..., N$$

If S is renormalized or deembedded, the interpretation is that the stimulations are applied to the renormalized external transmission lines at the plane of deembedding.

The other relative active quantities are simply transformations on the active S_m.

Active Y-parameter

$Y_0^{(M)}$ is the port admittance of the m -th mode.

$$active\text{-}Y_m \equiv Y_0^{(m)} \frac{1 - active\text{-}S_m}{1 + active\text{-}S_m} \qquad m = 1, ..., N$$

Active Z-parameter

$Z_0^{(m)}$ is the impedance of the m -th mode.

$$active\text{-}Z_m \equiv Z_0^{(m)} \frac{1 + active\text{-}S_m}{1 - active\text{-}S_m} \qquad m = 1, ..., N$$

Active VSWR

Active Voltage standing wave ratio is supported only for driven modal projects.

$$active\text{-}VSWR_m \equiv \frac{1 + |active\text{-}S_m|}{1 - |active\text{-}S_m|} \qquad m = 1, ..., N$$

Passivity

A scalar quantity based on the matrix Q = I - conjugate(transpose(S)) * S For every frequency, the value must be no larger than 1.

Design	Enables you to plot or tabulate properties of objects, such as their volumes.
Expression Cache	The values of expressions listed in the Expression cache of the Solution setup can be plotted, for example, as a function of adaptive pass to monitor their convergence.
Expression Convergence	This is intended to plot convergence, as a function of adaptive pass, of expressions in the Expression cache of the Solution setup. In defining the report, for "Context", select Solution: SetupN:Adaptive Pass. ExprDelta will show the change in the value of the expression as a function of adaptive pass, while ExprGoal will show, for comparison, the convergence goal for this expression, as defined in the Analysis Setup under the panel Expression Cache.

2. Select a quantity to plot from the **Quantity** list. The available quantities will depend upon the selected category and the setup of the design.

Selecting a Field Quantity to Plot

When plotting field quantities, the quantity can be a value that was automatically calculated by HFSS such as the magnitude of S_{11}, a value from a calculated expression, or an intrinsic (inherent) variable value such as frequency or phase.

To select a field quantity to plot:

1. When you create the report, specify the Report Type as "Fields" and the plot type (for example, radiation pattern.)
2. In the **Report** dialog, select Geometry for the Context, unless you are plotting scalar (for example, integration). For example, to plot near-field values across a sphere, you select the sphere object from the **Geometry** list in the **Traces** dialog box when you create a report.
3. In the **Report** dialog, select one of the following categories:

Variables	Intrinsic variables, such as frequency or phase, or user-defined project variables, such as the length of a quarter-wave transformer.
Output Variables	User defined expressions applied to derive quantities from the original field solution.
Calculator Expressions	Includes scalar and vector field quantities automatically calculated by HFSS, as well as derived field quantities that are defined by calculated expressions you set up in the Fields Calculator.

4. Select a quantity to plot from the **Quantity** list. The available quantities will depend upon the selected category and the setup of the design. See Field Quantities list for definitions.

Selecting a Far-Field Quantity to Plot

When plotting far-field quantities, the quantity can be a value that was calculated by HFSS such as antenna gain, a value from a calculated expression, or an intrinsic (inherent) variable value such as frequency or theta.

To select a far-field quantity to plot:

1. When you create the report, specify the Report Type as "Far Fields."
2. In the **Report** dialog box, select one of the following **Categories** for the field setup:

Variables	Intrinsic variables, such as frequency or theta, or user-defined project variables, such as the length of a quarter-wave transformer.
Output Variables	User defined expressions applied to derive quantities from the original field solution.
rE	The selected component of the radiated electric field, which is multiplied by the radial distance, *r*.
Gain	Gain is four pi times the ratio of an antenna's radiation intensity in a given direction to the total power accepted by the antenna.
Directivity	Directivity of the antenna.
Realized Gain	Realized gain is four pi times the ratio of an antenna's radiation intensity in a given direction to the total power incident upon the antenna port(s).
Axial Ratio	Axial ratio of the electric field.
Polarization Ratio	Polarization ratio of the electric field.
Antenna Params	HFSS-calculated quantities that include peak directivity, radiated power, accepted power, radiation efficiency including the total and component values at Phi and Theta, max U, and array factors. For far-field setups, the decay factor for lossy materials is calculated as a constant for all far fields.
Normalized Bistatic RCS	*For designs with plane incident waves. RCS is not supported for other types of incident waves.* The normalized radar cross section. $\frac{\sigma}{\lambda_0^2}$ where λ_0 is the wavelength of free space.

Radar Cross-Section (Bistatic RCS)

For designs with Plane Incident Waves. (RCS is not supported for other types of incident waves).

The radar cross-section (RCS) or echo area, σ, is measured in meters squared and represented for a bistatic arrangement (that is, when the transmitter and receiver are in different locations as shown in the linked figure). This is represented by

$$\sigma = \frac{4\pi r^2 \left|\boldsymbol{E}_{scat}\right|^2}{\left|\boldsymbol{E}_{inc}\right|^2}$$

where

- E_{scat} is the scattered E-field.
- E_{inc} is the incident E-field.

Complex (Bistatic) RCS

For designs with Plane Incident Waves. (RCS is not supported for other types of incident waves)

The equation for complex (bistatic) RCS is calculated as:

$$\sqrt{\sigma} = 2\sqrt{\pi} R \frac{E_{scat}}{\left|\boldsymbol{E}_{inc}\right|}$$

where

- E_{scat} is the scattered E-field.
- E_{inc} is the incident E-field.

This form retains the phase information.

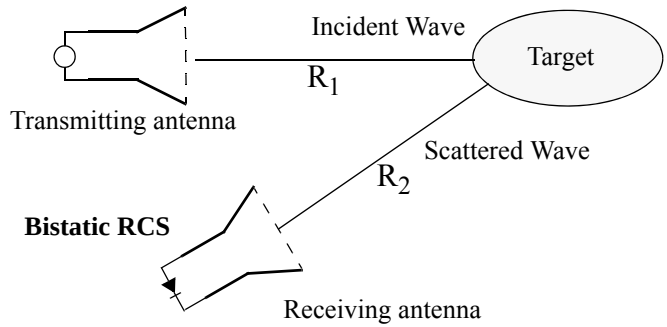
Monostatic RCS

For designs with Plane Incident waves. (RCS is not supported for other types of incident waves) A proper incident angle sweep should exist at the incident wave source setup before HFSS can plot Monostatic RCS.

The radar cross-section (RCS) or echo area when the transmitter and receiver are at the same location.

For Monostatic RCS, you need not be concerned with the Theta and Phi values defined in the radiation sphere. Only the incident wave Theta and incident wave Phi are used in calculating a Monostatic RCS plot.

The following diagram shows the bistatic RCS concept, with separate transmitting and receiving antennas.



Each **Category** item that you select causes the **Quantity** list to offer quantities appropriate to selected category. **Category** selection for a Variable of an Output Variable lists those available in each case. Selecting Antenna Parameters as Category causes the Quantity list to show Antenna parameters.

3. Select the Quantity to apply to the selected Category.

   If the **Category** item you select is rE, Gain, Directivity, or Realized Gain, you will need to specify the polarization of the electric field by selecting from the **Quantity** list. This ability to plot the gain of certain vector components (polarizations) of the electric field allows you to evaluate how well your antenna radiates in desired polarizations.

Total	The combined magnitude of the electric field components.
Phi	The phi component.
Theta	The theta component.
X	The x-component.
Y	The y-component.
Z	The z-component.
LHCP	The dominant component for a left-hand, circularly polarized field.
RHCP	The dominant component for a right-hand, circularly polarized field.
CircularLHCP	The polarization ratio for a predominantly left-hand, circularly polarized antenna.
CircularRHCP	The polarization ratio for a predominantly right-hand, circularly polarized antenna.
SphericalPhi	The polarization ratio for a predominantly ϕ-polarized antenna.

SphericalTheta	The polarization ratio for a predominantly θ-polarized antenna.
L3X	The dominant component for an x-polarized aperture using Ludwig's third definition of cross polarization.
L3Y	The dominant component for a y-polarized aperture using Ludwig's third definition of cross polarization.

The plot's Y axis field shows the combined selections.

For example, if you select Gain as the Category, and RHCP as the Quantity, HFSS evaluates the equation as follows:

$$Gain_{RHCP} = \frac{4\pi U_{RHCP}(\theta,\phi)}{P_{acc}} = \frac{4\pi |rE_{RHCP}(\theta,\varphi)|^2}{2\eta P_{acc}}$$

$$= \frac{4\pi \left| r\vec{E}(\theta,\varphi) \bullet \left(\frac{\hat{\theta} - j\hat{\phi}}{\sqrt{2}} \right) \right|^2}{2\eta P_{acc}}$$

4. You can also select a function to apply to the your selections for the **Category** and **Quantity** (for example, mag).

   As you make selections in the **Report** dialog for Category, Quantity, and Function, the Y field shows the combined calculation they describe.

5. Click **New Report** to create the Report.

   The new report based on your selections is displayed.

Related Topics

Selecting a Function

Setting a Range Function

Working with Traces

Creating Reports

Modifying Reports

Technical Notes: Antenna Parameters

Technical Notes: Polarization of the Electric Field

Technical Notes: Spherical Cross-Sections

Plotting Vertical Cross-Sections of Far Fields

When plotting far fields, a vertical cross-section plot results from holding phi fixed and sweeping theta through a range of values.

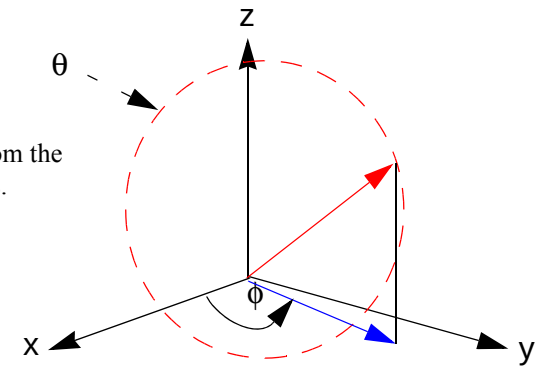
1. Open the **Report** dialog box.
2. Click the ellipsis **[...]** button for the sweep variable corresponding to phi.

   This displays a dialog listing all values for the phi variable. The values are the result of the range of phi you specified during the infinite sphere's setup.
3. Select the fixed value that phi should take in the plot.

   HFSS will display values for the vertical cross-section at selected phi cuts of the problem region at a set of theta rotations.

The figure shown below demonstrates the orientation of the vertical cross-section when ϕ is the fixed variable:

θ values are an infinite radial distance away from the origin for far-field plots.



Plotting Horizontal Cross-Sections of Far Fields

When plotting far fields, a horizontal cross-section results from holding theta fixed and sweeping phi through a range of values.

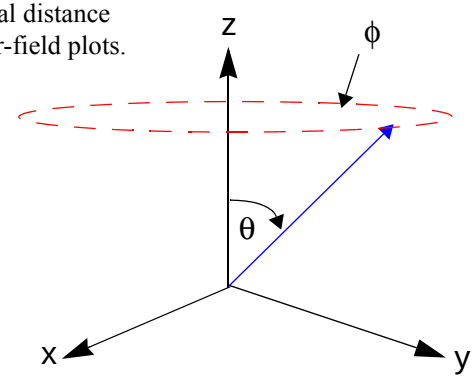
1. Click the ellipsis **[...]** button for the sweep variable corresponding to theta.

   To the right, all of the possible values for the theta variable are listed. The values are the result of the range of theta you specified during the infinite sphere's setup.
2. Select the fixed value that theta should take in the plot.

   HFSS will display values for the horizontal cross-section at selected theta cuts of the problem region at a set of phi rotations.

The figure shown below demonstrates the orientation of the sphere on which the field is computed when θ is the fixed variable:

ϕ values are an infinite radial distance away from the origin for far-field plots.



Selecting a Near-Field Quantity to Plot

When plotting near-field quantities, the quantity can be a value that was calculated by HFSS, a value from a calculated expression, or an intrinsic (inherent) variable value such as frequency or theta.

To select a near-field quantity to plot:

1. When you create the report, specify the Report Type as "Near Fields."
2. In the **Report** dialog box, select one of the following categories:

Variables	Intrinsic variables, such as frequency or theta, or user-defined project variables, such as the length of a quarter-wave transformer.
Output Variables	User defined expressions applied to derive quantities from the original field solution.
Near E	The radiated electric field in the near region.
Max Near Field Params	The maximum radiated electric field in the near region.
Near Normalized Antenna	The resultant plot is: field quantity / (maximum field quantity value over the entire infinite sphere).

3. If you selected the **Near E** category, specify the polarization of the electric field by selecting one of the following types of quantities from the **Quantity** list:

NearETotal	The combined magnitude of the electric field components.
NearEPhi	The phi component of the electric field.
NearETheta	The theta component of the electric field.
NearEX	The x-component of the electric field.

NearEY	The y-component of the electric field.
NearEZ	The z-component of the electric field.
NearELHCP	The dominant component for a left-hand, circularly polarized electric field.
NearERHCP	The dominant component for a right-hand, circularly polarized electric field.
NearECircularLHCP	The polarization ratio for a predominantly left-hand, circularly polarized antenna.
NearECircularRHCP	The polarization ratio for a predominantly right-hand, circularly polarized antenna.
NearEL3X	The dominant component for an x-polarized aperture using Ludwig's third definition of cross polarization.
NearEL3Y	The dominant component for a y-polarized aperture using Ludwig's third definition of cross polarization**.**

If a Near-field plot takes a long time to plot, be sure to perform **File>Save** when the plot is displayed. This saves the calculated data and permits fast display on subsequent viewings of the plot.

Related Topics

Technical Notes: Polarization of the Electric Field

Selecting an Emission Test Quantity to Plot

1. When you create the report, select Emission Test.
2. In the **Report** dialog box, select one of the following categories and apply an appropriate Quantity.

Variables	Intrinsic variables, such as frequency or theta, or user-defined project variables, such as the length of a quarter-wave transformer.
Output Variables	User defined expressions applied to derive quantities from the original field solution.
Sphere	A sphere of 1, 3, 10, or 30 meters, or of the same dimensions and RBS Simple or RBS exact (where RBS is random binary [bit] sequence).
Cylinder	A cylinder of 3 or 10 meters, or of the same dimensions and PRBS Simple or PRBS exact.

3. Select a **Function** for the quantity from the function list.
4. For Emission Test, the **Report** dialog also contains a button for specifying the digital signal options. The default values are a rise time of 0 seconds, and a hold time of 1 second. To specify other values, click **Digital Signal...**.

   This displays the **Digital Signal Options** dialog. It contains fields for the rise time and hold time.

5. **OK** the specified values or Cancel, Use Defaults, or Save As Default as appropriate.

Plotting Imported Solution Data

1. In the **Solution** pull-down list in the **Report** dialog box, click the imported data you want to plot.
2. Follow the procedure for creating a report.

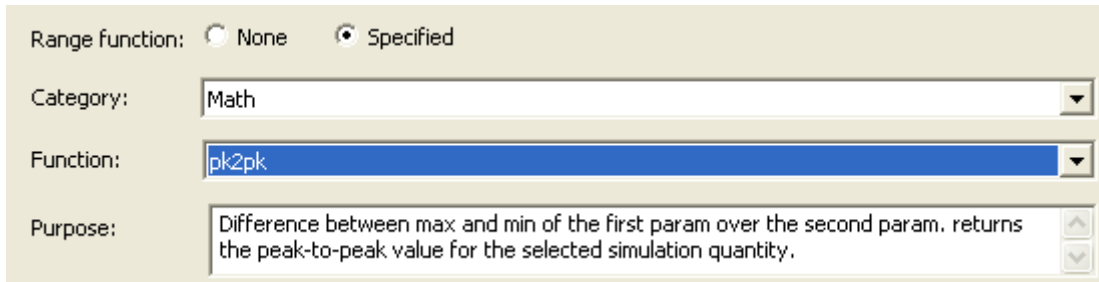
Setting a Range Function

To apply a range function to the Y, Z, or Mag component of a trace:

1. Click the **Range Function** button in the **Reports** dialog.

   This opens the **Set Range Function** dialog. The functions available are the same as described in the Selecting a Function section, with the exception of those for the Eye Measurements category.
2. To enable Range function selection, click the Specified radio button.

   Selecting the None radio button disables the Range Function fields.
3. Select the **Category**, and then an associated Function to apply. The available categories depend on the plot, and Category enables the display of associated functions.



   Given a selected Function, and Category, the **Set Range Function** dialog displays a text field that explains the Purpose of the function. See figure above.

   Selecting a function causes the display of a description in the **Purpose** field. If the function requires a value (such as the XatYVal Math function or the pw_minus_max Pulse Width function), the table below the function field displays the name, editable value field, unit, and description.
4. Use the **Over Sweep** drop down menu to select from available sweeps.
5. To select from available Sweeps, or to edit them, use the ellipsis [**...**] button and uncheck **Use All Sweeps**.

   This enables a list of the sweeps. The sweep(s) you select is displayed on the Over Sweep line. You can use the buttons to **Clear All Selections** or **Select All** sweeps.
6. Select the Sweeps **Default** or **Edited** radio buttons to specify whether to accept the default or edited sweeps.
7. To edit the sweeps further, select the ellipsis button to display an **Edit Sweep** dialog.

   For frequency variables, this lets you specify a single value, linear step, linear count, decade

count, octave count, or exponential count. You can **Add** legal values to the list of sweep values, **Update** the list for changes, or **Delete** selected entries.

8. Click **OK** to apply the range function.

Related Topics

Range Functions

Selecting a Function

Eye Measurement Range Function Parameters

Range Functions

The following table shows the **Functions** according to their **Categories**. The most commonly used range categories are **Math** and **Radiation**. Other functions could be used if needed. Use the category links to navigate to tables with definitions of functions.

Category	Functions
Math	max, min, pk2pk, rms, sum, mean, variance, stddev, integabs, avgabs, rmsAC, ripple, pkavg, XatYMin, XatYMax, YAtXMin, YAtXMax, XAtYVal, YAtXVal, XWidthAtYVal
PulseWidth	pulsemin, pulsemax, pulsemintime, pulsemaxtime, pulsefall9010, pulsefront1090, pulsefront3090, pulsetail50, pulsewidth5050, pw_plus, pw_minus, pw_plus_avg, pw_minus_avg, pw_plus_max, pw_minus_max, pw_plus_min, pw_minus_min, pw_plus_rms, pw_minus_rms
Overshoot, Undershoot	overshoot, undershoot.
TR & DC	crestfactor, formfactor, distortion, fundamentalmag, delaytime, risetime, deadtime, settlingtime
Error	iae, ise, itae, itse
Period	per, pmax, pmin, prms
AC	gainmargin, phasemargin, gaincrossover, phasecrossover, lowercutoff, uppercutoff, bandwidth, peakgain, peakgainfreq.
Radiation	xdb10bandwidth, xdb20bandwidth, lSidelobeX, lSidelobeY, rSidelobeX, rSidelobeY
Eye Measurements	EyeLevelZero, EyeLevelOne, EyeAmplitude, EyeHeight, EyeSignalToNoise, EyeOpeningFactor, EyeWidth, EyeJitterP2P, EyeJitterRMS, EyeRiseTime, EyeFallTime, MinEyeWidth, MinEyeHeight **Note:** Refer to the SI Wave or Nexxim online help for more information. The Purpose field offers brief descriptions of each.

Math Functions

***avg**	Returns the average of the values of the selected quantity. avg = (Area between the curve and the X-axis) / (X length of the curve)
avgabs	Returns the mean of the absolute value of the selected quantity.
integabs	Absolute value of integral.
max	Returns maximum value of the simulation quantity.
mean	Returns the average in the set of quantities selected. mean = sum(all y-value) / (number of y-values)
min	Returns the minimum value of the simulation quantity.
rms	Returns the root mean square value of the selected quantity.
rmsAC	Returns the AC RMS for the selected quantity.
ripple	Returns the ripple factor (AC RMS/Mean) for the selected quantity.
pkavg	Returns the ratio of the peak to peak-to-average for the selected quantity.
pkp2pk	Peak to peak. Difference between max and min of the first parameter over the second parameter. Returns the peak-to-peak value for the selected simulation quantity.
sum	Returns the sum of the given values.
stddev	Returns the standard deviation of given values.
variance	Calculates the variance of the given values.
XAtYMax	Threshold crossing time: report first time (x value) at which an output quantity crosses YMax.
XAtYMin	Threshold crossing time: report first time (x value) at which an output quantity crosses a user definable threshold.
XAtYVal	Returns the X value at the first occurrence of Y value.
XWidthAtYVal	Returns the X width between the first 2 occurrences of Y value.
YAtXMax	Returns the X value at maximum value of Y.
YAtXMin	Returns the Y value at minimum value of X.
YAtXVal	Returns the Y value at the first occurrence of X value.

Radiation Functions

lsidelobeX	The 'x' value for the left side lobe: the next highest value to the left of the max value.
lsidelobeY	The 'y' value for the left side lobe: the next highest value to the left of the max value.
rSidelobeX	Returns the X value of right side-lobe occurrence.
rSidelobeY	Returns the Y value of right side-lobe occurrence.
xdb10beamdwidth	Width between left and right occurrences of values 'x' db10 from max. Takes 'x' as argument (3.0 default). To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.
xdb20beamwidth	Width between left and right occurrences of values 'x' db20 from max. Takes 'x' as argument (3.0 default) To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.

Pulse Width Functions

Note: In this table, the functions with the asterisk (*) do not appear on the Range Function drop down menu. They can still be used via text entry.

pulsefall9010	Returns the pulse fall time of the selected quantity according to the 90%-10% estimate.
pulsefront1090	Returns the pulse front time of the selected quantity according to the 10%-90% estimate.
pulsefront3090	Returns the pulse front time of the selected quantity according to the 30%-90% estimate.
pulsemax	Returns the pulse maximum from the front and tail estimates for the selected quantity.
pulsemaxtime	Returns the time at which the maximum pulse value of the selected quantity is reached.
pulsemin	Returns the pulse minimum from the front and tail estimates for the selected quantity.
pulsemintime	Returns the time at which the minimum pulse value of the selected quantity is reached.
pulsetail50	Returns the pulse tail time of the selected quantity from the virtual peak to 50%.

pulsewidth5050	Returns the pulse width of the selected quantity as measured from the 50% points on the pulse front and pulse tail.
***pwl**	Piecewise Linear.
***pwl_periodic**	Piecewise Linear for periodic extrapolation on x.
***pwlx**	Piecewise Linear x with linear extrapolation on x.
pw_minus	Pulse width of the first negative pulse.
pw_minus_avg	Returns the average of the negative pulse width input stream.
pw_minus_max	Returns the maximum pulse width of the negative pulse of input stream.
pw_minus_min	Returns the minimum pulse width of the negative pulse of input stream.
pw_minus_rms	Returns the rms of the negative pulse width input stream.
pw_plus	Returns the pulse width of the first postive pulse.
pw_plus_avg	Returns the average of the positive pulse width input stream.
pw_plus_max	Returns the maximum pulse width of the positive pulse of input stream.
pw_plus_min	Returns the minimum pulse width of the positive pulse of input stream.
pw_plus_rms	Returns the rms of the positive pulse width input stream.

Overshoot/Undershoot

Overshoot	Calculates peak overshoot given a threshold value and number of evenly spaced points over entire time range.
Undershoot	Calculates peak undershoot given a threshold value and number of evenly spaced points over entire time range.

TR & DC Functions

crestfactor	Returns the crest factor (peak/RMS) for the selected simulation quantity.
formfactor	Returns the form factor (RMS/Mean Absolute Value) for the selected quantity.
distortion	Returns the total distortion for the selected simulation quantity and an additional argument frequency, which is the frequency in Hz at which to calculate the fundamental RMS of the simulation quantity.
fundamentalmag	Returns the RMS value of the fundamental frequency for the selected quantity, and an additional argument, Frequency, which specifies the fundamental frequency.
delaytime	Obtains the time from zero to 50% of the target point.

risetime	Obtains the time taken to go from 10% to 90% of target point.
deadtime	Obtains the latest time when the qtyl is within a tolerance of zero.
settlingtime	Returns the latest time at which the value of the selected simulation quantity fell outside its tolerance band. The target value of the quantity and the +/- bandwidth of the tolerance band are the additional arguments.

Error Functions

iae	Returns the integral of the absolute deviation of the selected quantity from a target value that is entered via the additional argument.
ise	Returns the integral of the squared deviation of the selected quantity from a target value that is entered via an additional argument.
itae	Returns the time-weighted squared deviation of the selected quantity from a target value that is entered via an additional argument.
itse	Returns the time-weighted squared deviation of the selected quantity from a target value that is entered via an additional argument. To use this function, you need to open the Add Trace Characteristics dialog and select the Error category.

Periodic Functions

per	Returns the period of a simulation quantity.
pmax	Max period of the selected simulation quantity.
pmin	Minimum period of the selected simulation quantity.
prms	Period Root Mean Square.

AC Functions

gainmargin	Returns the gain margin in dB at the phase crossover frequency of the selected simulation quantity. It also requires a reference simulation quantity to which the measured quantity is compared and the AC magnitude and phase angle of the reference quantity. These are entered as the arguments Reference Channel, Base Source Magnitude, and Base Source Angle.
gaincrossover	Returns the gain crossover frequency (where the gain is 0 dB) of the selected simulation quantity in Hz.
phasecrossover	Returns the phase crossover frequency, at which the phase is -180 degrees, in Hz for the selected simulation quantity.

phasemargin	Returns the phase angle in degrees at the gain crossover frequency of the selected simulation quantity.
lowercutoff	Returns the lower 3dB frequency of the selected simulation channel in Hz..
uppercutoff	Returns the upper 3dB frequency of the selected simulation channel in Hz.
bandwidth	Returns the 3dB bandwidth of the selected simulation quantity. For bandwidth, the calculation is based on 3dB below the maximum peak.
peakgain	Returns the peak value of gain of the selected simulation quantity in dB.
peakgainfreq	Returns the frequency in Hz at which the peak gain of the selected simulation quantity occurs.

Related Topics

Selecting a Function

Range Functions

Eye Measurement Range Function Parameters

Eye Measurement Range Function Parameters

The Eye Measurement category of range functions provide the means to evaluate several characteristics of eye diagrams. Each of the Eye Measurement functions includes the following parameters. Specify the Value by editing the Value text field for the parameter name. Specify the unit for the parameter by selecting from the Unit drop down menu.

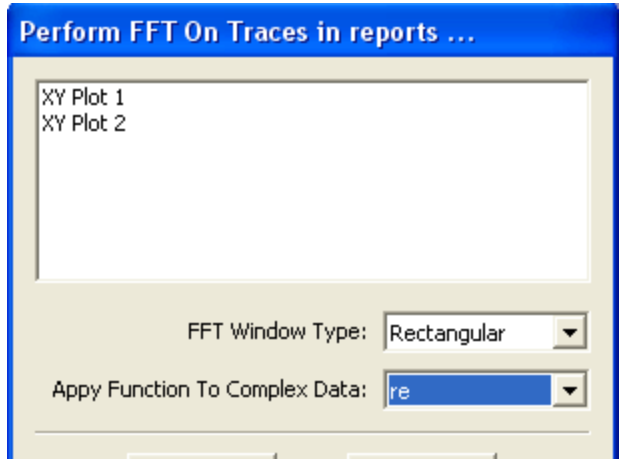
Name	Default Value	Default Unit	Description
Unit Interval	0	ns	Unit interval of signal
Start Offset	0	ns	Offset at beginning of signal
End Offset	0	ns	Offset at end of signal
Autocrossing amplitude	1		Nonzero number means that crossing amplitude is calculated automatically.
Crossing amplitude	0	mV	Specify crossing amplitude used for eye measurement data computation.

Related Topics

Setting a Range Function

Perform FFT on a Report

You can perform FFT on an existing 2D plot by using the **Results>Perform FFT** command. This opens the **Perform FFT on Traces in Reports** dialog.



1. Select the report you want from the list in the dialog.
2. Select the FFT Window type from a drop down list.

   Select the window type to apply. Windowing functions cause the FFT of the signal to have non-zero values away from ω. Each window function trades off the ability to resolve comparable signals and frequencies versus the ability to resolve signals of different strengths and frequencies.
3. Select the function to apply to complex data.

The new report displays and appears in the Project tree. The new report name prefixes FFT to the name of the original report. Trace names are also prefixed with FFT.

Related Topics

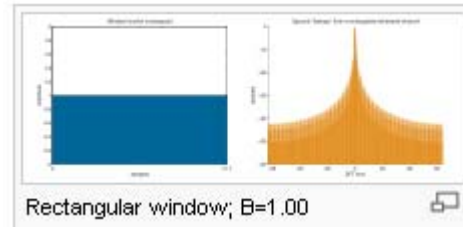
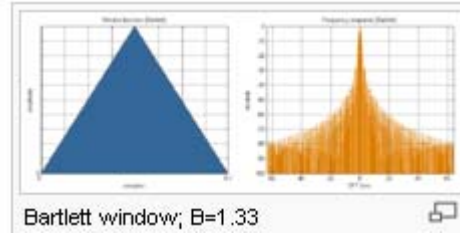
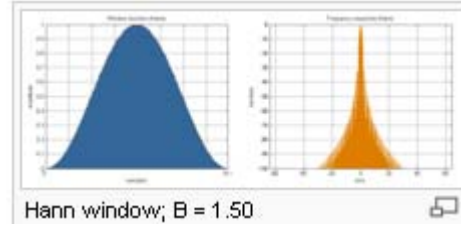
FFT Window Functions

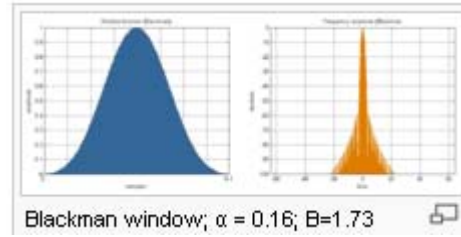
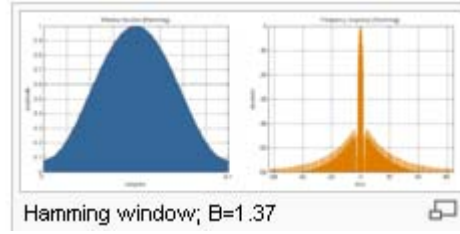
Apply FFT to Report Functions

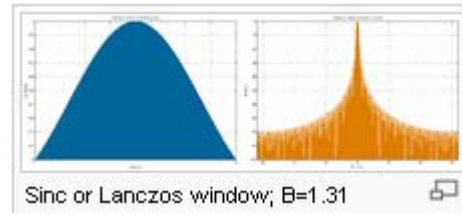
Perform TDR on a Report

FFT Window Functions

The window type list for **Perform FFT on Report** includes:

Window Function	Preferred Use
Rectangular	A low dynamic range function offering good resolution for signals of comparable strength. Poor when signals have very different amplitudes. $w\ (n)=1$. 
Tri	A Bartlett window with the endpoints valued at zero. 
Van Hann	A moderate dynamic range function, designed for narrow band applications. $$w(n) = 0.5\left(1-\cos\frac{2\pi n}{N-1}\right)$$ 

Window Function	Preferred Use
Blackman	A high dynamic range function, with lower resolution, designed for wide band applications. $$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right)$$ where $a_0 = (1-\alpha)/2$; $\alpha_1 = 1/2$; $\alpha_2 = \alpha/2$ 
Hamming	A moderate dynamic range function, designed for narrow band applications. It minimizes the maximum sidelobe. $$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$$ 

Window Function	Preferred Use
Lanczos	The Lanczos window offers a windowed form of the infinite sinc filter, providing the central lobe of a horizontally-stretched sinc, sinc(x/a) for −a ≤ x ≤ a.
	
Weber	
Welch	This approach applies a parabola-shaped window to the frequency domain data. It is based on the Bartlett method but splits the signal into overlapping segments, which are then windowed. The intent is to balance the influence of data in the center of the function.

Related Topics

Perform FFT on a Report

FFT Window Functions

Apply FFT to Report Functions

Perform TDR on a Report

Apply FFT to Report Functions

The choices include:

ang_deg	Angle (phase) of a complex number, cut at +/-180
ang_rad	Angle in radians
arg	
cang_deg	Cumulative angle (phase) of the first parameter (a complex number) in degrees, along the second parameter (typically sweep variable). Returns a double precision value cut at +/-180.
cang_rad	Cumulative angle of the first parameter in radians along a second parameter (typically a sweep variable) Returns a double precision value.
dB(x)	20*log10(|x|)
dB 10normalize	10*log [normalize(mag(x))]
dB 20normalize	20*log [normalize(mag(x))]
dBc	

im	Imaginary part of the complex number
mag	Magnitude of the complex number
normalize	Divides each value within a trace by the maximum value of the trace. ex. normalize(mag(x))
re	Real part of the complex number

Related Topics

FFT Window Functions

Perform FFT on a Report

Perform TDR on a Report

Perform TDR on Report

You can perform TDR on an existing 2D plot by using the **Results>Perform TDR on Report** command. This opens a **Perform TDR on Traces in reports** dialog.

1. Select the report you want from the list in the dialog.
2. Specify the input signal as Step or Impulse and give the rise time.
3. Select the TDR Window type from a drop down list.

   Select the window type to apply. Windowing functions cause the FFT of the signal to have non-zero values away from ω. Each window function trades off the ability to resolve comparable signals and frequencies versus the ability to resolve signals of different strengths and frequencies.

   You may choose to specify a window width as a percentage.

   If you select the Kaiser function, you can specify a Kaiser number.

The new report displays and appears in the Project tree. The new report name prefixes TDR to the name of the original report. Trace names are also prefixed with TDR.

Related Topics

Perform FFT on a Report

Plotting in the Time Domain

Specifying Output Variables

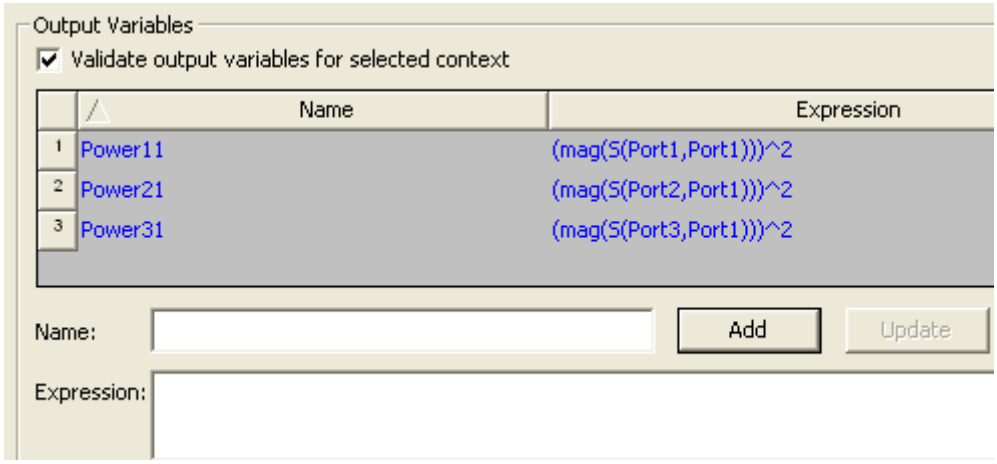
You can define output variables to use calculated expressions of results as adaptive convergence goals and for reports. You can use output variables in the Expression Cache tab of the **Solution Setup** dialog and select them as Categories in the **Reports** dialog, as well as the **Output Variables** window. You can access the **Output Variables** window in several ways.

- Click **HFSS or HFSS-IE>Results>Output Variables**.
- In the Project tree, right-click on **Results** and select **Output Variables** from the short-cut menu.
- In the **Solution Setup** dialog, select the **Expression Cache** tab, click the **Add...** button to display the **Add to expression cache** dialog, and click the **Output Variables** button.
- Click the **Output Variables** button the **Reports** dialog.

The **Output Variables** window contains four sections:

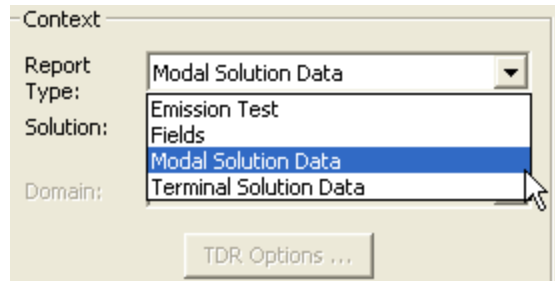
- **Output Variables** section, where you can specify the name and expression for a new output variable. Existing variables appear in the list at the top of the window. Clicking the triangle in the Name bar inverts the sort order.

  At top of the **Output Variables** window, you can use a checkbox to **Validate output variable for selected context**.

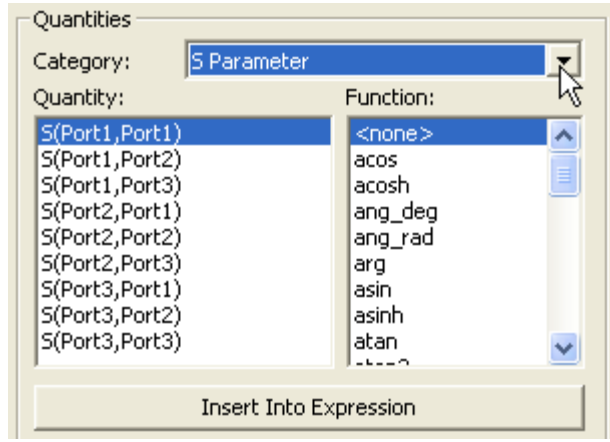


- **Context section,** where you specify the Report type, the Solution, and for appropriate report types, the Domain. Changing the Report type to Near Fields, or Far Fields causes a Geometry menu to appear. Selecting Emission tests require a digital signal. All selections affect the Func-

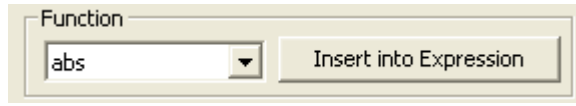
tions listed.



- **Expressionsection**, where you can insert quantities into the Expression area of the **Output Variables** section.



- **Function section**, where you can insert completed expressions into the Expression area of the **Output Variables** section.



Related Topics

Adding a New Output Variable

Building an Output Expression Using Existing Quantities

Viewing the Output Variable Convergence

Getting Started Guides: HFSS Transient: A Ball Grid Array

Adding a New Output Variable

The **Output Varibles** window can be accessed in several ways. To add an output variable:

- Click **HFSS or HFSS-IE>Results>Output Variables**.

- In the Project tree, right-click on **Results** and select **Output Variables** from the short-cut menu.
- In the **Solution Setup** dialog, select the **Expression Cache** tab, click the **Add...** button to display the **Add to expression cache** dialog, and click the **Output Variables** button.
- Click the **Output Variables** button the **Reports** dialog.

  Existing variables appear in the list at the top of the window.

1. In the **Output Variables** section, enter a name for the new variable in the **Name** box.
2. To enter an expression, do one or both of the following:
   a. Type part or all of the expression directly in the **Expression** area. Valid functions appear in blue. Invalid functions appear in red, which usually means that you need to provide quantities for a function.
   b. Insert part or all of the expression using the options in the Calculation and Function sections.
3. Click **Add** to add the new variable to the list.
4. Repeat to add additional variables.
5. When you are finished adding output variables, click **Done** to close the **Output Variables** window.

Related Topics

Deleting Output Variables

Building an Output Variable Expression Using Existing Quantities

Viewing the Output Variable Convergence

Building an Output Variable Expression Using Existing Quantities

When you are entering an expression for a new output variable, you can insert part or all of the expression using the options in the **Calculation** and **Function** sections of the **Output Variables** window.

The **Output Variables** window can be accessed in several ways. To open the window:

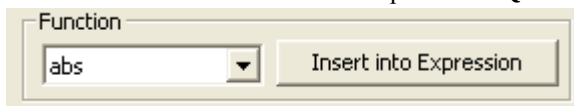
- Click **HFSS or HFSS-IE>Results>Output Variables**.
- In the Project tree, right-click on **Results** and select **Output Variables** from the short-cut menu.
- In the **Solution Setup** dialog, select the **Expression Cache** tab, click the **Add...** button to display the **Add to expression cache** dialog, and click the **Output Variables** button.
- Click the **Output Variables** button the **Reports** dialog.

  The **Output Variables** window appears. Existing variables appear in the list at the top of the window.

To add an input variable by inserting part or all of the expression:

1. In the **Output Variables** section, enter a name for the new variable in the **Name** box.
2. Specify the Context
   a. From the **Report Type** pull-down list, select the type of report from which you want to select the quantity. The Report types listed can vary with the design, but can include Emission Test, Fields, Far Fields, Modal Solution Data, or Terminal Solution Data.
   b. From the **Solution** pull-down list, select the solution from which you want to select the quantity. In the case of Transient, you may also select Spectral or Transient.
   c. In some cases the Report Type selection enables additional selections. For example, a Fields Category calls for for Geometry and possibly Points. A Transient project may offer selections of Domain as Sweep or Time. A Emission Test Report enables a Digital Signal button.
3. Specify Quantities and optionally apply functions.
   d. From the **Category** list, select the type of quantity you want to enter. The Report type selections cause the list to provide applicable categories.
   e. From the **Quantity** list, select the quantity, variable, or the geometry, as applicable
   f. From the **Function** list, select a ready-made function to apply to the select quantity.
   g. Click **Insert Into Expression**.

   The selected quantity is entered into the **Expression** area of the **Output Variables** section.
4. To insert a function that does not depend on a **Quantity** selection into the Expression area.



   a. In the **Function** section, select a ready-made function from the pull-down list.
   b. Click **Insert Function into Expression.**

   The function appears in the **Expression** area of the **Output Variables** section.
5. When you are finished defining the variable in the **Expression** area, click **Add** to add the new variable to the list.

   Legal expressions display in blue. Incomplete expressions, that require additional input, display in red.
6. Repeat to add additional variables.
7. Click **Done** to close the **Output Variables** window.

Note Remember the evaluated value of an expression is always interpreted as in SI units. However, when a quantity is plotted in a report, you have the option to plot values in units other than SI. For example, the expression "1+ang_deg(S_{11})" represents an "angle" quantity evaluated in radians though plotted in degrees units. To represent an angle quantity in degrees, you would specify units as "1 deg + ang_deg(S_{11})".

Related Topics

Adding a New Output Variable

Building an Expression Using Existing Quantities

Function List for Output Variables

Function List for Output Variables

The Output Variables dialog includes a second function list containing functions to enter directly into the Expression field. These functions can also be applied to previously specified Quantities and Functions.

Some of these functions can operate along an entire curve. These are: deriv, min, max, integ, avg, rms, pk2pk, cang_deg and cang_rad. These functions have syntax as follows:

- deriv(*quantity*) implicitly implies derivative over the primary sweep
- deriv(*quantity*, *SweepVariable*) explicitly means derivative over the sweep variable specified in the second argument (such as "Freq").

You can select from the functions in the **Output Variables** dialog **Function** list or type them directly into the Expression field, if necessary: The functions in the Output Variables dialog list and the ones in the Function list on the Report dialog box are defined in Table

Related Topics

Add Trace Characteristics

Set Range Function

Deleting Output Variables

To delete output variables:

1. Remove all references to the output variable in the project.
2. Save the project to erase the command history.
3. Click **HFSS>Results>Output Variables** or, in the Project Tree, right-click on **Results** and select **Output Variables** from the short-cut menu.

   This opens the **Output Variables** dialog.
4. Select the variable and click the **Delete** button.
5. Click **OK** to close the dialog.

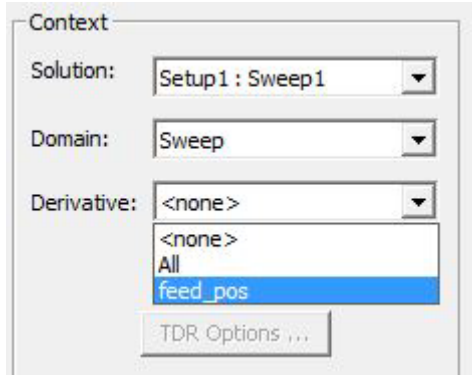
Related Topics

Adding a New Output Variable

Derivative Tuning for Reports

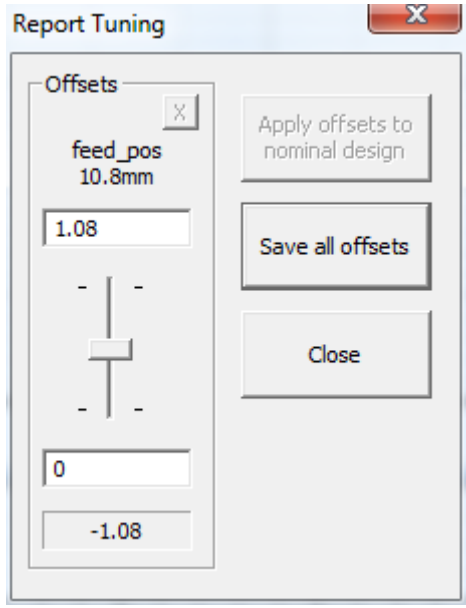
The Derivative Tuning feature available in the results menu is based on the derivatives that you can request for selected variables in the solution setup. It is limited to quantities like S-parameters. Frequencies, and local quantities like fields cannot be tuned.

A common way to use this feature is to produce, for an output quantity of interest, two curves in one plot. One curve is produced with the **Report** dialog selection for the Context Derivative as <none>. The other has the All or specific variable selection.



This gives you a plot with initially two identical curves, one on top of the other. You can then right-click on Results in the Project tree and select **Tune Reports**. The Report Tuning window appears. You can use the slider to tune one curve interactively while the other curve stays to provide a refer-

ence. This way you can see interactively how small changes in variables affect the result. You can then apply those offsets to the original variable values and re-solve the design.



The overall procedure for Report Tuning follows.

1. Generate a solution with one or more variables for which you select Use on the Derivatives tab of the solution setup.
2. Use the Reporter to create one curve with Derivative selection in the Context panel of the report dialog set to None.
3. Then create another curve, but in the Context pane. select for Derivatives, All or the variable of interest. This gives you two identical curves, one top of the other.
4. Click the **HFSS>Results>Tune Reports** or right-click on **Results** in the Project tree and click **Tune Reports** from the short cut menu. The menu item is disabled if no variables have been selected in the Derivatives tab of the solution setup.

   This displays a **Report Tuning** dialog which lists the variables available for tuning. The example above shows a **Report Tuning** dialog for a design with one variable.
5. You can use the slider to adjust the value of each available variable. When you move a slider, the **Apply offsets to nominal design** button and the **Save all offsets** buttons are enabled. The dialog displays the change to the variable selected. For example, if the variable is $length with a value of 1mm, and the slider shows 0.1mm, then the effective value of $length for the purpose of derivatives is 1.1mm. If you exit the dialog by applying the offset or offsets (click **Apply offsets to nominal design**), $length is assigned a value of 1.1mm. You can then re-solve, and get results based on the derivative's prediction. The **Close** button lets you close the dialog.

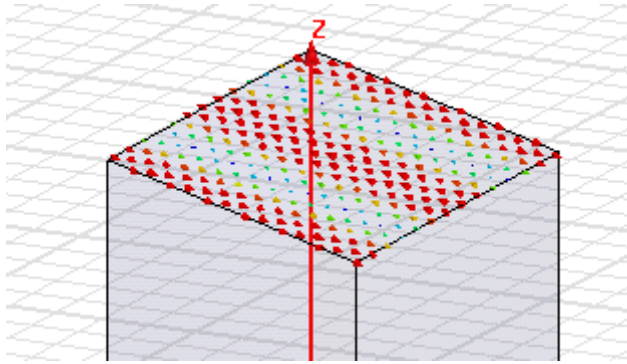
Related Topics

Produce Derivatives for Selected Variables

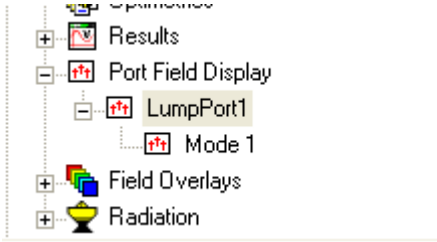
Examples: Tune a Coax Fed Patch

Port Field Display

The Port Field Display feature lets you see a visualization of the electric field patterns belonging to the modes of the 2D port solution. To have an available solution," Save Fields" must be checked for the Solution setup Advanced tab, or for a fast or discrete sweep. If you select a port mode listed in the **Project** tree, the **Modeler window** displays the a visualization of the selected mode for the port field.

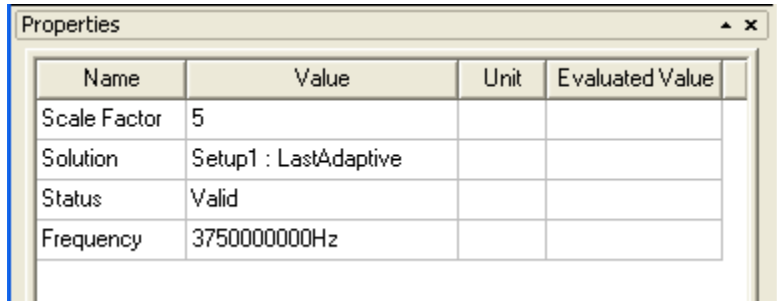


If an HFSS design has ports, they are listed Port Field Display icon in the Project tree. The modes for each port appear under the port name.



Select a port name to display a **Properties** dialog with the properties for that port, or for individual Modes listed under that port. These are used as subsequent plot defaults for the port field display. Your control of port field displays is purely through the docked **Property** window. You can edit the

Scale Factor field, which governs the sizes of the arrows. If multiple solutions or frequencies are available, you can select from a dropdown menu.



If no 2D solutions exist for the selected port or mode, the Status line of the **Properties** dialog will say "No solutions available."

Depending on the view you begin with, and the location and size of the mode in the design, you may want to right click on the selected mode to display a **Zoom to Region** command.

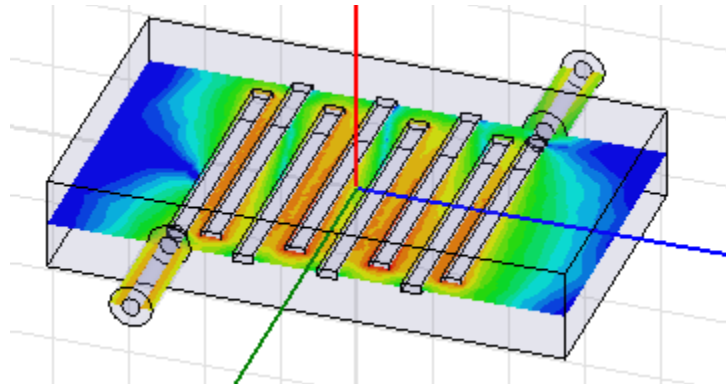
The difference between the Port Field Display and the Field Overlay is that the Port Field Display gives the pure 2D port solution. A Field Overlay provides a the 3D field solution on the port, which is the sum of the excitation in the port and the reflections by the 3D structure. The Port Field display lets you examine the field patterns with which the 3D structure is excited.

Related Topics

Plotting Field Overlays

Plotting Field Overlays

Field overlays are representations of basic or derived field quantities on surfaces or objects for the current design variation. You can set the design variation via the **Set Design Variation** dialog. This dialog box is accessible from the **Solution Data** window via by clicking the ellipsis button on the right of the Design Variation field, and via the **HFSS** or **HFSS-IE>Results>Apply Solved Variation** command.



You can also overlay existing 3D Polar Plots of near or far fields on the model window by using the **HFSS** or **HFSS-IE>Fields>Plot Fields>Radiation** command, or by right-clicking on Field Overlays in the Project tree and selecting **Plot Fields>Radiation Field**. You can also create animations of field plots.

To plot a basic field quantity:

1. Select a point, line, surface, cutplane, or object to create the plot on or within.
2. Click **HFSS** or **HFSS-IE>Fields**>**Plot Fields**., or right-click on Field Overlay icon in the Project tree and select **Plot Fields**, or right click in the modeler window, and select **Plot Fields** from the context menu.
3. On the **Plot Fields** menu, click the field quantity you want to plot.

   The available selections depend on the solved solution. For definitions of the usual quantities, see the list under Quantity command.

   If you select a scalar field quantity, a scalar surface or volume plot will be created. If you select a vector field quantity, a vector surface or volume plot will be created. If you select a vector quantity, you will be able to specify a Streamline plot. If the quantity you want to plot is not listed, see Named Expression Library.

   For projects with Temperature dependent materials, the **HFSS** or **HFSS-IE>Fields>Plot Fields>Other...** menus selections include **Temperature**.

   After you select the field quantity to plot, the **Create Field Plot** dialog box appears.

   The Specify Name field shows a name based on the field quantity you selected, and the Quantity list shows the field quantity selected.
4. To specify a name for the plot other than the default, select **Specify Name**, and then type a new

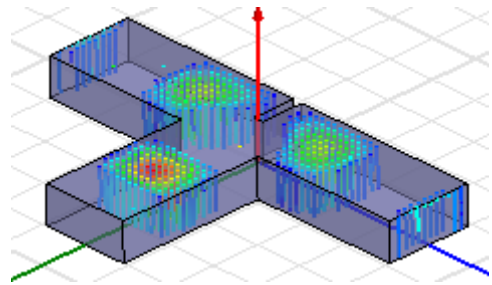
name in the **Name** text box.

5. Select the solution to plot from the **Solution** pull-down list.
6. To specify a folder other than the default in which to store the plot, select **Specify Folder**, and then click a folder in the **Plot Folder** pull-down list, or type the name you wish to use. Plot folders are listed under **Field Overlays** in the project tree. Plot folders let you group plots with the same quantity together. All field plots under the same folder share the same color key.
7. Under **Intrinsic Variables**, select the frequency and phase angle at which the field quantity is evaluated.
8. If desired, you can select a different field quantity to plot from the **Quantity** list.
9. Select the volume (region) in which the field will be plotted from the **In Volume** list.

   This selection enables you to limit plots to the intersection of a volume with the selected object or objects. You can select and deselect any items in the **In Volume** list. You can mix model objects with non-model boxes. For example you might want to see a plot from part of two model objects by restricting the region to a non-model box overlapping those parts.

Note Multiple selection should be used when there is a discontinuous field on a surface. If not, the field on both sides of the surface is plotted and each interferes with the other.

10. If you selected a vector quantity, you can use the checkbox to select **Streamline** plot. Streamlines are often used to indicate magnetic flux lines, etc. in plots. See Setting Field Plot Attributes for adjusting the streamline display and Setting Fields Reporter Options for setting Streamline defaults.

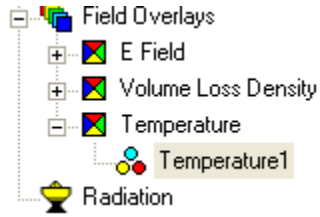


11. Click **Done**.

    The field quantity is plotted on the surfaces or within the objects you selected. The plot uses the attributes specified in the **Plot Attributes** dialog box.

    The new plot appears in the view window. It is listed in the specified plot folder in the project tree. If you have created a field plot on a simulation in progress, the field plot is updated after the last adaptive solution. Each category of plots (such as Temperature) are listed separately in

the Project Tree.



If you want to update the field overlay before then, to view progress in the solution, select the Field icon in the Project tree that contains the field plot of interest, right-click to display the short cut menu, and select **Update Plots**.

To turn off the display of the plot, right click on the plot and select **Plot Visibility** from the short-cut menu. Unchecking **Plot Visibility** turns off the plot display.

Related Topics

Plotting Derived Field Quantities

Overlaying 3D Polar Plots on Models

Creating Animations

Select Objects.

Select Faces.

Creating an Object List

Selecting the Face or Object Behind

Using the Fields Calculator

Port Field Display

Setting a Plot's Visibility

Example Projects

Working with Scalar Field Plot Markers

Technical Notes: Field Overlays

Technical Notes: Field Quantities

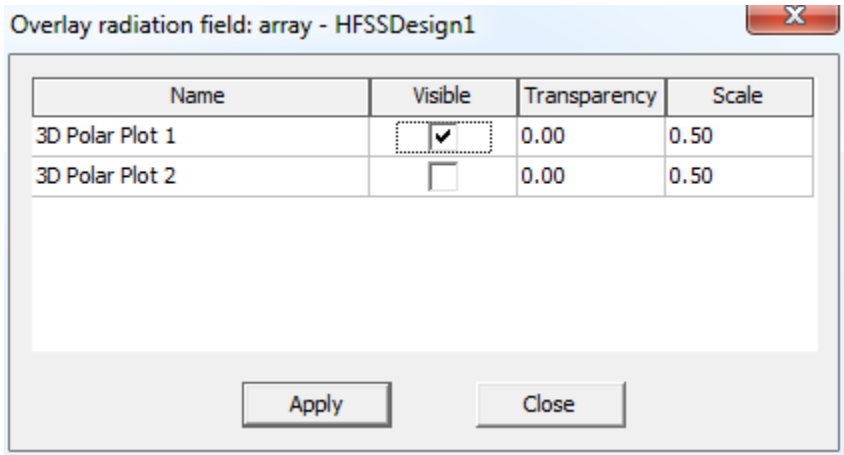
Technical Notes: Specifying the Phase Angle

Overlaying 3D Polar Plots on Models

You can overlay existing 3D Polar Plots or near or far fields on the model window by using the **HFSS** or **HFSS-IE>Fields>Plot Fields>Radiation Field...**command, or by right-clicking on Field Overlays in the Project tree and selecting **Plot Fields>Radiation Field**. You can edit the visibility, transparency and scale of the polar plot by using a dialog.

1. Create one or more 3D Polar Plots for your model. This enables the Radiation Field.. command.
2. Click the **HFSS** or **HFSS-IE>Fields>Plot Fields>Radiation Field** command, or by right-clicking on Field Overlays in the Project tree and selecting **Plot Fields>Radiation Field.**

This displays a dialog listing any existing 3D polar plots.



3. Check the Visible box and click **Apply** to cause that plot to appear in the model window. You can also edit the Transparency and Scale. Other properties of the 3D plot are controlled in its properties window.

Related Topics

Plotting Field Overlays

Creating 3D Polar Plots

Example Projects: Helical Antenna

Example Projects: Pyramidal Horn

Plotting Derived Field Quantities

Derived field quantities are field quantity representations that have been deduced from the original field solution using the Fields Calculator.

1. Select a point, line, surface, or object to create the plot on or within.
2. On the **HFSS** or **HFSS-IE** menu, or right-click on the **Field Overlays** icon in the Project tree, and point to **Fields>Named Expression**.
3. Select the derived quantity you want to plot, and then click **OK**.

   The **Create Field Plot** dialog box appears.
4. To specify a name for the plot other than the default, select **Specify Name**, and then type a new name in the **Name** text box.
5. Select the solution to plot from the **Solution** pull-down list.
6. To specify a folder other than the default in which to store the plot, select **Specify Folder**, and then click a folder in the **Plot Folder** pull-down list, or type the name you wish to use. Plot folders are listed under **Field Overlays** in the project tree.

7. Under **Intrinsic Variables**, select the frequency and phase angle at which the field quantity is evaluated.
8. Select the derived field quantity to plot from the **Quantity** list.
9. Select the volume, or region, in which the field will be plotted from the **In Volume** list.

   This selection enables you to limit plots to the intersection of a volume and the selected object.
10. Click **Done**.

    The derived field quantity you created in the Fields Calculator is plotted on the surfaces or objects you selected. The new plot is listed in the project tree under **Field Overlays**.

Related Topics

Using the Fields Calculator

Technical Notes: Field Quantities

Technical Notes: Specifying the Phase Angle

Add Trace Characteristics

Creating Scalar Field Plots

A scalar plot uses shaded colors or contoured lines to illustrate the magnitude of field quantities on surfaces or volumes.

1. Do one of the following:
   a. To create a scalar surface plot, select the faces on which you want to plot the fields.
   b. To create a scalar volume plot, select the objects within which you want to plot the fields.
2. Click **HFSS** or **HFSS-IE**>**Fields**>**Plot Fields**.
3. On the **Plot Fields** menu, click the scalar field quantity you want to plot.

   The **Create Field Plot** dialog box appears.
4. Follow the procedure for plotting field overlays.

The plot uses the attributes specified in the **Plot Attributes** dialog box.

The new plot will be listed in the specified plot folder in the project tree.

Related Topics

Modifying Field Plot Attributes

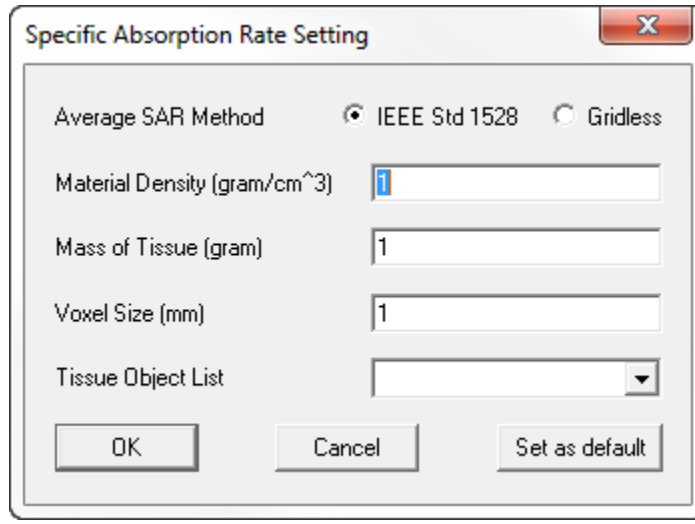
Modifying SAR Settings

HFSS uses default specific absorption rate (SAR) settings when creating a local SAR or average SAR field overlay plot. It does use mass density for each material if defined for each material. Oth-

erwise (that is, if the mass density for that library material is 0), the density in the **Specific Absorbtion Rate Setting** dialog will be used. To change the default settings:

1. Click **HFSS**>**Fields**>**SAR Setting**.

   The **Specific Absorption Rate Setting** dialog box appears.



2. To select the Average SAR Method, select the radio buttons for IEEE Standard P 1528.4 or Gridless, which is the legacy Average SAR algorithm. Selecting Gridless disables the Voxel Size and Tissue Object list fields.
3. In the **Material Density** text box, enter the mass density of the dielectric material in g/cm^3. This provides a default mass density if not specified in the material definition.
4. In the **Mass of Tissue** text box, enter the mass of the material that surrounds each mesh point. This can be a value between 1 and 10.
5. In the **Voxel Size** field, specify a size. The units are millimeters.

   The voxelization process takes all elements as rectilinear. So it is better to solve projects with curvilinear off. See Specifying Initial Mesh Settings.

   Since the number of voxels depends on the size of the model and the size of the voxel you can expect that computational complexity increases in O(n^3) or more as the model size increases and the voxel size decreases. Below are a few suggestion:

   - For large models you can set up an average SAR plot before you solve so that the data from the first phase could be generated at solve time. With the tissue setting you can also perform SAR calculation over partial model.
   - Set the solver order to 1 for best performance.
   - The algorithm takes full advantage of multi-cores. Set the number of processors to use in the desktop performance tab in **General Options**. If your machine is hyper threaded, you should only use half the number of cores.

6. If the Tissue Object List is empty, all conductors are treated as tissues. You can create object lists for the menu by selecting objects and clicking, **Modeler>List>Create>Object List**. After saving the Project, you can then select from available object lists. When you select a list, all objects in the list are treated as tissues.
7. Click **OK**.

Related Topics

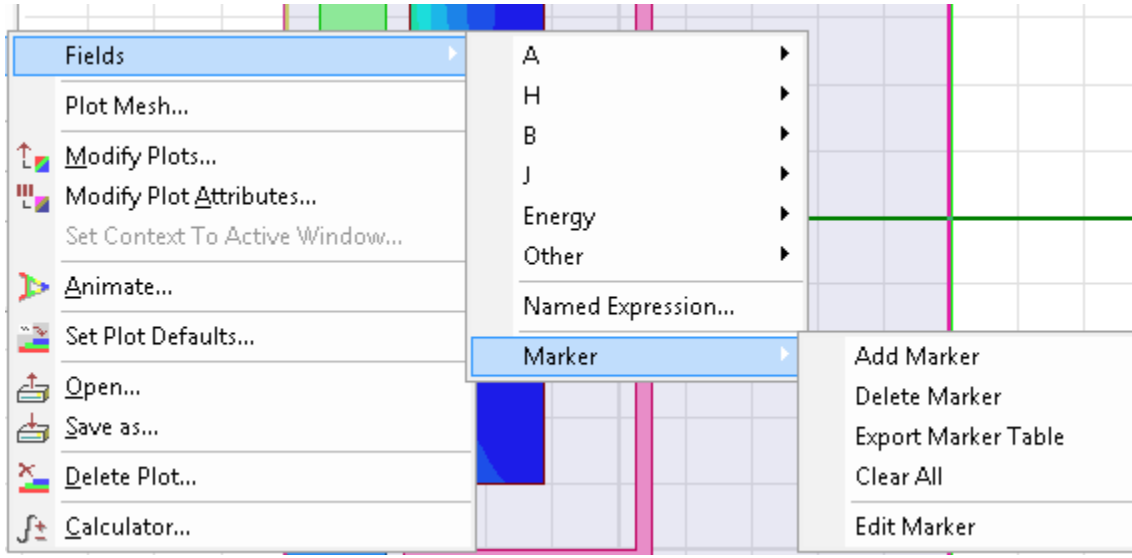
Viewing and Modifying Material Attributes

Technical Notes: Calculating the SAR

Working with Scalar Field Plot Markers

The field overlay plot marker feature enables you to create a marker at selected points in the scalar field overlay plot geometry, and to obtain the field value at that point. The fields Marker sub-menu enables you to:

- Add Marker
- Delete Marker
- Export Marker Table
- Clear All markers
- Edit Marker



Adding a Field Plot Marker

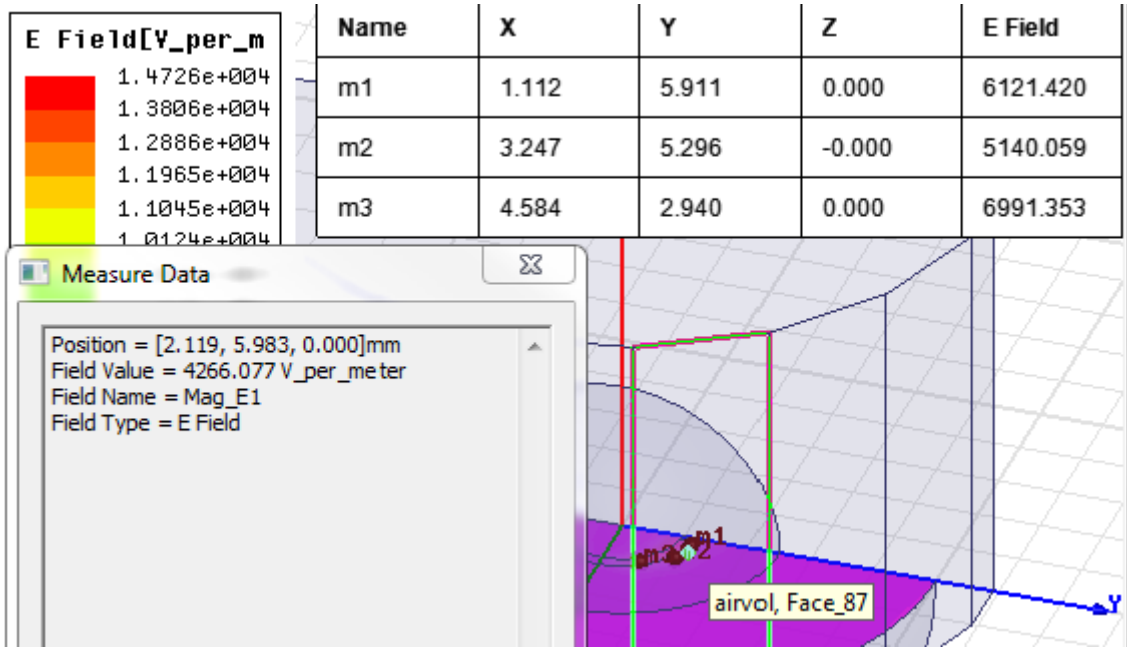
To add one or more field plot markers to a scalar field:

1. On the main menu click **HFSS** or **HFSS-IE** and then select

Fields>Fields>Marker>AddMarker. Alternatively you can right-click anywhere in the modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>AddMarker**.

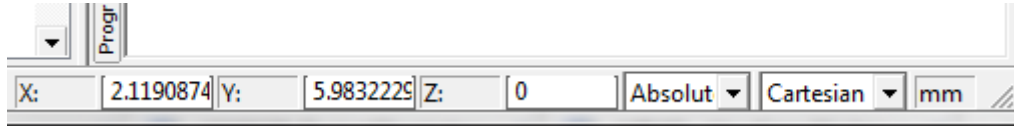
The **Measure Data** dialog box opens and a round dot appears at the tip of the cursor.

2. Drag the dot over the spot on the field overlay plot where you want to add a marker. The Measure Data dialog shows detailed information for the spot currently under the dot.



Name	X	Y	Z	E Field
m1	1.112	5.911	0.000	6121.420
m2	3.247	5.296	-0.000	5140.059
m3	4.584	2.940	0.000	6991.353

3. Click the desired point in the field overlay plot to add the marker at that location. Alternatively you can select the position of the marker by entering the values manually in the edit fields at the bottom of the window as shown below.



A table showing the marker coordinates and associated field value is also created and added to the modeler window.

4. Repeat as desired to add additional markers.
5. Press the **Esc** key when finished adding markers.

Deleting a Field Plot Marker

To delete a field plot marker:

1. Click on the marker you want to delete to select it. The row corresponding to the selected marker will be highlighted in the marker table.

   Press and hold the **Ctrl** key and click to select multiple markers.

2. On the main menu click **HFSS** or **HFSS-IE** and then select **Fields>Fields>Marker>Delete Marker** to delete the selected marker(s).

   - Alternatively you can right-click anywhere in the modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>Delete Marker**.

   - You can also simply press the **Delete** key to delete the selected marker(s).

Exporting a Field Plot Marker Table

You can export a field plot marker table to either a comma- or tab-delimited file as follows:

1. On the main menu click **HFSS** or **HFSS-IE** and then select **Fields>Fields>Marker>Export Marker Table**.

   Alternatively you can right-click anywhere in the modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>Export Marker Table**.

2. In the **Export As** dialog box, choose the export format, either **.csv** or **.tab**, and save the file in the desired location.

   The exported file can then be imported into another application such as a spreadsheet.

	A	B	C	D	E
1	Name	X	Y	Z	H[A_per_m]
2	m1	2	22.935	1.517	1234
3	m2	1.517	17.373	1.517	123
4	m3	2.488	23.516	1.517	234
5	m4	2.751	23.828	1.517	1324

Clearing All Field Plot Markers

To clear all field plot markers in the active modeler window do one of the following:

- On the main menu click **HFSS** or **HFSS-IE** and then select **Fields>Fields>Marker>Delete All**.
- Alternatively you can right-click anywhere in the modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>Delete All**.

Deleting all the markers also removes the marker table.

Editing Field Plot Markers

To edit field plot markers:

1. On the main menu click **HFSS** or **HFSS-IE** and then select **Fields>Fields>Marker>Edit Marker**.

   Alternatively you can right-click anywhere in the modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>Edit Marker**.
2. Click on the marker you wish to edit. The row corresponding to the selected marker is highlighted in the marker table. The properties of the marker are displayed in the Properties window.
3. You can change **Position** of the selected marker by editing its coordinate values. The marker table changes to reflect the new value. You can also click the **Color** value bar to select a new color for the marker.
4. Optionally, press the **Delete** key to delete the marker and its corresponding entry in the marker table.
5. When finished editing markers, press **Esc** to exit the marker editing function.

Creating Vector Field Plots

A vector plot uses arrows to illustrate the magnitudes of the x-, y-, and z-components of field quantities. Vector plots can be created on surfaces or volumes.

1. Do one of the following:
   a. To create a vector surface plot, select the faces on which you want to plot the fields.
   b. To create a vector volume plot, select the objects within which you want to plot the fields.
2. Click **HFSS** or **HFSS-IE>Fields>Plot Fields**.
3. On the **Plot Fields** menu, click the vector field quantity you want to plot.
4. Follow the procedure for plotting field overlays.

If you select a vector quantity, you can also check Streamline for the plot.

Related Topics

Modifying Field Plot Attributes

Plotting Field Overlays

Modifying Field Plots

1. Click **HFSS** or **HFSS-IE>Fields>Modify Plots**  , or in the Project tree, select the **Field Overlays** icon, right-click, and select **Modify Plots** or use the "m" hotkey.

   The **Select Field Plot(s)** dialog box appears.

   Optionally, you can right-click on an existing plot listed under the Field Overlays in the Project tree, bypass the **Select Field Plot** dialog and go to step 3
2. Select the plot you want to modify in the **Select** column, and then click **OK**.

The **Modify Field Plot** dialog opens.

- Optionally, click the Specify Name check box to enable the name field.
- Optionally, click the Specify Folder field enable the **Plot Folder** drop down.
- Optionally, select a different Solution from the dropdown menu.
- Optionally, select a different field type, if available.

3. Under **Intrinsic Variables**, specify the frequency and phase at which the field quantity will be evaluated.
4. Optionally, select a different field quantity to plot from the **Quantity** list.
   - To choose a calculated expression, select **Calculator** from the **Category** pull-down list. If you choose Calculator, click the Fields Calculator button to display the calculator.
   - To choose a predefined field quantity, select **Standard** from the **Category** pull-down list. Select from the Quantity list.
5. Select the volume, or region, in which the field will be plotted from the **In Volume** list.

   This selection enables you to limit plots to the intersection of a volume and the selected object.
6. Click **Apply** to make the changes and leave the dialog open, or click **Done** to apply the changes and close the dialog.

   The field quantity is plotted on the surfaces or within the objects you selected. The modified plot is listed in the specified plot folder in the project tree.

   The plot uses the attributes specified in the **Plot Attributes** dialog box.

Related Topics

Setting a Plot's Visibility

Technical Notes: Specifying the Phase Angle

Add Trace Characteristics

Setting Field Plot Attributes

Setting Field Plot Attributes

After creating a mesh or field overlay on a surface or volume, you can modify its appearance by changing the settings in the **Plot Attributes** dialog box. You will modify the settings for a plot folder and all plots in that folder will use the same attributes.

1. Click **HFSS** or **HFSS-IE**>**Fields**>**Modify Plot Attributes**  , or in the Project Manager window, select the **Field Overlays** icon, and select **Modify Attributes** or use the "a" hotkey.

   The **Select Plot Folder** window appears.
2. In the **Select Plot Folder** window, select the plot you want to modify, and then click **OK**. (You can also select the specific plot in the Project tree, and select Modify Attributes from the right click menu.

   A dialog box with attribute settings for the selected plot (whether for an E Field plot or a Mesh Overlay plot) appears.
3. For an E Field Plot, under the following tabs in the dialog box, you can control the following

plot attributes: For Mesh plot attributes, see below.

Color map	The number of colors used and how they are displayed.
Scale	The scale of field quantities, including the number of divisions in the scale, and whether to use a linear or log scale.
Marker/Arrow	• The appearance of points (for scalar point plots). • The appearance of arrows (for vector plots).
Plots (if not streamline)	• The plot selected. • To display or hide the mesh on the plot's surface or volume. • The type of isovalue display (for scalar plots.) • The transparency based on solution value. • Whether to add a grid (that is, a mesh overlay), and to set the grid color. • Specify the plot resolution as **Coarse**, **Normal**, **Fine**, or **Very Fine**. This affects the use of memory for animating plots. For large plots with more frames to animate, use **Coarse** or **Normal** to reduce memory requirements and improve performance. For smaller plots with few frames, if higher resolution is required, use **Fine** or **Very Fine**. • The spacing of arrows (for vector plots).
Plots (if streamline is checked)	• The plot selected • The linestyle as solid or cylinder from dropdown menu. • Line width, specified using a slider. • Whether to show marker on streamline. • Seeds density spacing. This affects the number of stream lines used to represent the quantity in the plot. Moving the slider to the left decreases the spacing and increases the number of stream lines. Moving the slider to the right increases the spacing and decreases the number of lines used to represent the quantity. • Min. and Max. values represented.

a. Under each tab, click **Save as default** if you want the tab's settings to apply to field overlay plots created after this point.

b. Select **Real time mode** if you want the changes to take effect immediately in the view window.

c. If this option is cleared, click **Apply** when you want to see the changes.

Related Topics

Setting a Plot's Visibility

Plotting the Mesh

Plotting Field Overlays

Modifying Field Plot Colors

1. Click **HFSS** or **HFSS-IE**>**Fields**>**Modify Plot Attributes**  , or in the Project tree, right click on the Field Overlays icon and select **Modify Plots** from the short-cut menu, or use the "m" hotkey.

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Color Map** tab.
4. Select one of the following color types:

Uniform	Field quantities are plotted in a single color. Choose the plot color from the **Color** palette.
Ramp	Field quantities are plotted in shades of a single color. Choose the plot color from the **Color** palette. The shade of the color corresponds to its field value.
Spectrum	Field quantities are plotted in multiple colors. Choose a color spectrum from the pull-down list. The values are Rainbow, Temperature, Magenta, and Grey. Each field value is assigned a color from the selected spectrum.

   You can choose **Save as Default**, if you want to use the current settings.

   Select **Real time mode** if you want these, or subsequent changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
5. Click the **Scale Tab**.
6. In the **Num. Divisions** field, enter the number of colors to use in the plot.

   You can choose **Save as Default**, if you want to use the current settings.

   Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
7. Click **Close** to dismiss the dialog box.

Related Topics

Setting a Plot's Visibility

Setting the Color Key Visibility

Moving the Color Key

Keyboard Shortcuts for HFSS General Purposes

Custom Keyboard Shortcuts

Setting the Color Key Visibility

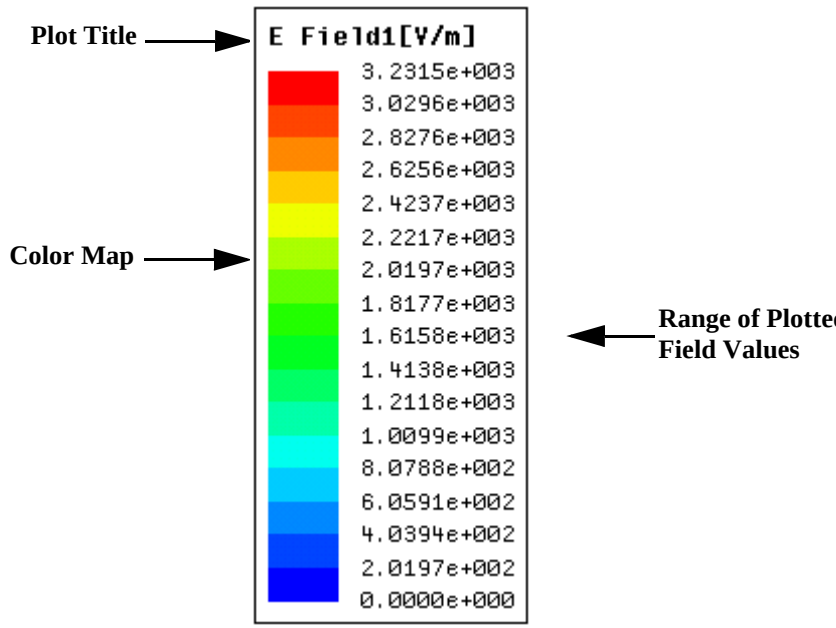
The color key (shown below) displays the range of plotted field values for a field overlay plot. It displays the colors that correspond to the range of field values on the plot.

1. Click **View**>**Active View Visibility**  .

   The **Active View Visibility** dialog box appears.
2. Click the **Color Keys** tab.
3. In the **Visibility** column, select the field overlay or mesh plots in which you want to display the color key. Clear the plots in which you want to hide the color key from view.
4. Click **Done** to dismiss the dialog box.

Alternatively, to hide the color key, right-click on the color key in the view window, and then click **Hide** from the shortcut menu.

Only the color keys in the selected plots will be visible.



Related Topics

Modifying Field Plot Colors

Moving the Color Key

Moving the Color Key

Click on the active field overlay plot's color key and drag it to a new location.

Related Topics

Setting the Color Key Visibility

Modifying the Field Plot Scale

To change how field quantities are scaled on the field overlay plot:

1. Click **HFSS** or **HFSS-IE**>**Fields**>**Modify Plot Attributes**  or in the Project tree, right click on the Field Overlays icon and select **Modify Plots** from the short-cut menu, or use the "m" hotkey.

   The **Select Plot Folder** window appears.

2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.

3. Click the **Scale** tab.

4. Optionally, to change the number of divisions in the field plot scale, set the Num. Division field to a new value. You can click **Save as Default**, if desired.

5. Select one of the following scale options:

Auto	The full range of field values will be plotted on the selected surface or volume. Selecting **Auto** enables the **Auto Scale Options** and disables the **Min** and **Max** fields. By default, precision is not limited and auto-min is the actual computed min on the plotted geometry.
Use Limits	Only the field values between the minimum and maximum values will be plotted. Field values below or above these values will be plotted in the colors assigned to the minimum or maximum limits, respectively. Selecting **Use Limits** enables the **Min** and **Max** fields and disables the **Auto Scale Options**. Field values have a precision of at most 6 decimal places (field solution files are saved in floating precision), so Min/Max numbers are displayed to this precision.
Specify Values	This enables a **Scale Values** button..

6. Optionally, use the **Units** drop down menu to select the default unit of measure for the plot.

   The units specified here appear on the Color map for the fields plot, and for the properties dialog for the field quantities.

7. If you selected **Use Limits**, enter the lowest field value to be plotted in the **Min.** text box and the highest field value to be plotted in the **Max.** text box.

   If you selected **Auto**, the **Auto Scale Options** are enabled. You should only changed for cases where auto-min is a small number. Use the 'Limits Max/Min precision to' checkbox to enable setting the drop down menu for the precision limit. The auto-min is the greater of the following:

   - Actual computed Min

- Max/pow(10, num digits of field precision)

If you selected Specify Values, you can click the **Scale Values** button. This opens a dialog with an editable, scrollable list of the current scale values. To apply the changes you make, click the **OK** button. To close the dialog without make changes, click **Cancel**.

8. If you selected **Auto** or use **Limits**, you can select one of the following options:

Linear	Field values are plotted on a linear scale.
Log	Field values are plotted on a logarithmic scale. If field plots have negative and positive values and when auto-scale is selected, the log-scale choice automatically sets the Min value as the Max/Min Ratio. (If field plots have all negative values, **Log** is not allowed.)

9. Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.

10. Optionally, you can use the **Save As Default** button to save the following to registry:
    - Whether to limit field precision,
    - The number of digits of field precision,
    - Whether to use log/linear scale.

    Auto scale is the default for new plots. For scalar-in-volume plots, iso-surface (rather than cloud) is the default display

11. Click **Close** to dismiss the window.

Modifying Vector Field Plot Arrows

To change the appearance of a vector field plot's arrows:

1. Click **HFSS** or **HFSS-IE>Fields>Modify Plot Attributes**  .

   The **Select Plot Folder** window appears.

2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.

3. Click the **Marker/Arrow** tab.
4. Under **Arrow Options**, select one of the following arrow types:

Line	The arrows are displayed as 2D/flat.
Cylinder	The arrow tails are displayed as cylinders. The arrowheads are displayed as 3D/round.
Umbrella	The arrow tails are displayed as 1D lines. The arrowheads are displayed as 3D/round.

5. Use the **Size** slider to increase (move to the right) or decrease (move to the left) the length and dimensions of the arrows. The arrows are resized relative to the size of the model geometry.

6. Select **Map Size** to scale the size of the arrows to the magnitude of the field quantity being plotted.
7. Select **Arrow tail** to include tails on all arrows.
8. Click the **Plots** tab.
9. HFSS plots arrows on a grid that is superimposed on the surface or object you selected for the plot. Under **Vector plot**, use the **Spacing** slider to increase (move to the right) or decrease (move to the left) the distance between arrows (grid points.)
   - Select **Uniform** if you want the arrows to be spaced equally.
10. Select **Real time mode** if you want the changes to take effect immediately in the view window.

    If this option is cleared, click **Apply** when you want to see the changes.
11. Click **Close** to dismiss the window.

Setting the Mesh Visibility on Field Plots

To display or hide the mesh on field plots, or change the mesh's color:

1. Click **HFSS** or **HFSS-IE**>**Fields**>**Modify Plot Attributes**  .

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Plots** tab.
4. Select **Add Grid** to display the mesh.
5. Optionally, select a color for the mesh from the **Color** palette.
6. Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
7. Click **Close** to dismiss the window.

Related Topics

Plotting the Mesh

Setting a Plot's Visibility

Modifying Scalar Field Plot Isovalues

1. Click **HFSS** or **HFSS-IE**>**Fields**>**Modify Plot Attributes**  .

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.

3. Click the **Plots** tab.
4. If the plot is a scalar surface plot, do the following:
   a. Select one of the following isosurface display types in the **IsoValType** pull-down list:

Line	Lines are drawn along the isovalues.
Fringe	Color is constant between isovalues.
Tone	Color varies continuously between isovalues.
Gourard	Color varies continuously across the plot.

   b. Optionally, if you selected **Fringe** or **Tone**, select **Outline** to add a border line between isovalues.
5. If the plot is a scalar volume plot, do the following:
   a. Select one of the following display types:

IsoValSurface	Color is drawn on the isovalues.
Cloud	Field values are represented by points that illustrate the spatial distribution of the solution. The higher the solution value, the greater the cloud density.

   b. Optionally, if you select **Cloud**, use the **Cloud density** slider to increase or decrease the number of points that represent the density on the volume.
   c. Optionally, if you select **Cloud**, enter a point size for the clouds in the **Point size** text box
6. Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
7. Click **Close** to dismiss the window.

Mapping Scalar Field Plot Transparency to Field Values

1. Click **HFSS** or **HFSS-IE**>**Fields**>**Modify Plot Attributes**  .

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Plots** tab.
4. Use the **Map transp.** slider to increase (move to the right) or decrease (move to the left) the transparency of the plot.
   - If you select **Map transp.**, the transparency of field values increases as the solution values decrease.
5. Select **Real time mode** if you want the changes to take effect immediately in the view window.

If this option is cleared, click **Apply** when you want to see the changes.

6. Click **Close** to dismiss the window.

Modifying Markers on Point Plots

For scalar point plots, a marker is used to represent a field quantity at a selected point. (For vector point plots, arrows are used.) Modify the shape and size of markers in the plot attributes window.

1. Click **HFSS** or **HFSS-IE>Fields>Modify Plot Attributes**  .

   The **Select Plot Folder** window appears.

2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.

3. Click the **Marker/Arrow** tab in the plot attributes window.
4. Under **Marker options**, select one of the marker types to represent the field quantity at the point:
   - **Sphere**
   - **Box**
   - **Tetrahedron**
   - **Octahedron**
5. Use the **Size** slider to increase (move to the right) or decrease (move to the left) the size of the marker.
6. Select **Map size** to scale the size of the marker to the magnitude of the quantity being plotted.
7. Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.

8. Click **Close** to dismiss the window.

Related Topics

Drawing a Point

Modifying Line Plots

Field quantities can be plotted directly on a line object. Scalar quantities are plotted as 3D color-shaded lines. Vector quantities are plotted as arrows that are based on the line.

To modify the appearance of line plots:

1. Click **HFSS** or **HFSS-IE>Fields>Modify Plot Attributes**  .

   The **Select Plot Folder** window appears.

2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.

3. Click the **Plots** tab.

4. Select one of the following isosurface display types in the **IsoValType** pull-down list:

Fringe	Color is constant between isovalues.
Tone	Color varies continuously between isovalues.
Gourard	Color varies continuously across the plot.

5. Select one of the following styles for the line object in the **Line style** pull-down list:

Cylinder	The line object is shaped like a cylinder.
Solid	The line object is a 3D solid.
Dash-Dash	The line object is represented by dashed black line segments.
Dot-Dot	The line object is represented by a series of dots.
Dash-Dot	The line object is represented by a a series of alternating dashed black line segments and dots.

6. Use the **Line width** slider to increase (move to the right) or decrease (move to the left) the thickness of the line.
7. By default, a polyline object is divided into 100 equally spaced points for post processing. To modify the number of points on the line, type a new value in the **Number of points** text box.
8. Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
9. Click **Close** to dismiss the window.

Related Topics

Drawing a Polyline

Setting a Plot's Visibility

To turn off the display of the plot, right click on the plot and select **Plot Visibility** from the shortcut menu. Unchecking **Plot Visibility** turns off the plot display.

To display or hide a field overlay or mesh plot from view in the **3D Modeler** window:

1. Click **View**>**Active View Visibility**  . Alternatively, you can select the **Active View Visibility** icon from the toolbar.

   The **Active View Visibility** dialog box appears.
2. Click the **FieldsReporter** tab.
3. In the **Visibility** column, select the field overlay or mesh plots you want to display. Clear the plots you want to hide from view.

   Only the selected plots will be visible.

Related Topics

Plotting the Mesh

Setting a Plot's Visibility

Saving a Field Overlay Plot

Field overlay and mesh plots are saved in the project file; however, you can save a plot to HFSS Field Plot File format (.dsp) and then open it in HFSS.

To save field overlay or mesh plot data to a .dsp file:

1. In the project tree, click the plot you want to export.
2. Click **HFSS** or **HFSS-IE>Fields>Save as**  .

   The **Select Field Plot(s)** dialog box appears.
3. Select the plots you want to export by checking the **Select** box, and then click **OK**.

   The file browser appears. **Field Plot Files (.dsp)** is the selected file type.
4. Specify the name of the .dsp file and the location in which to save it.
5. Click **Save**.

   The plot is exported to the specified .dsp file.

The file you created can be opened in HFSS version 9 and later. Simply click **HFSS >Fields>Open**.

Related Topics

Exporting Animations

Opening a Field Overlay Plot

To open a field overlay or mesh plot that you have saved to HFSS Field Plot File format (.dsp) in HFSS version 9 and later:

1. Click **HFSS** or **HFSS-IE>Fields>Open**  .
2. The file browser appears. **Field Plot Files (.dsp)** is the selected file type.
3. Browse to the location of the .dsp file you want to open, and then click the file name.
4. Click **Open**.

   The plot appears in the view window. It is listed under **Field Overlays** in the project tree.

Deleting a Field Overlay Plot

1. Click **HFSS** or **HFSS-IE>Fields>Delete Plot**  .

   The **Delete Plots** dialog box appears.
2. Select the plots you want to delete by checking the **Delete** check box.
3. Click **OK**.

   The selected plots are deleted.

Alternatively, click the plot in the project tree that you want to delete, and then press **Delete**  .

Setting Field Plot Defaults

Each new field plot uses the default plot settings specified in the **Set Plot Defaults** dialog box.

To modify the default plot settings:

1. If a plot folder has not been created, click **Field Overlays** in the project tree.
2. Click **HFSS** or **HFSS-IE>Fields>Set Plot Defaults**  .

   The **Set Plot Defaults** dialog box appears.
3. Select the solution to plot from the **Solution** pull-down list.
4. Select the plot folder in which new plots will be stored from the **Quantity type** pull-down list. Choose one of the following options:

New Folder	Each new plot will be stored in a separate folder in the project tree.
Automatic	Each new plot will be stored in a folder determined by HFSS as the most appropriate based on the plotted field quantity. For example, all surface magnitude E plots will be stored in the same folder.
An existing folder	Select the existing folder in which you want to store new plots.

Note Plots stored in the same folder will use the same color key. The **Auto** scale setting will be based on the maximum field solution value present in a plot.

5. Under **Intrinsic Variables**, specify the frequency and phase angle at which the field quantity is evaluated.
6. Click **OK**.

Related Topics

Technical Notes: Specifying the Phase Angle

Using the Fields Calculator

The Fields calculator enables you to perform computations using basic field quantities. The calculator will compute derived quantities from the general electric field solution; write field quantities to files, locate maximum and minimum field values, and perform other operations on the field solution.

The calculator does not perform the computations until a value is needed or is forced for a result. This makes it more efficient, saving computing resources and time; you can do all the calculations without regard to data storage of all the calculated points of the field. It is generally easier to do all the calculations first, then plot the results.

A online help Fields Calculator Cookbook provides examples of the following:

- Calculating Numerical Quantities
- Calculating Quantities for 2D (Line) Plot Outputs
- Calculating Quantities for 3D (Surface or Vector) Plot Outputs
- Calculating Quantities for 3D (Volume) Plot Outputs
- Calculating Quantities for Animated Outputs
- Creating User Defined Named Expressions Library

The chm version is here:

. in html format.

In addition to the chm format version, you can also view a pdf format ***HFSS Fields Calculator Cookbook:*** *A Brief Primer and Collection of Step-byStep Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

Related Topics

Opening the Fields Calculator

Context Area

Calculator Stack

Registers

The Stack Commands

Input Commands

General Commands

Scalar Commands

Vector Commands

Output Commands

Calculating Derived Output Quantities

Named Expression Library

Opening the Fields Calculator

To open the Fields Calculator, do one of the following:

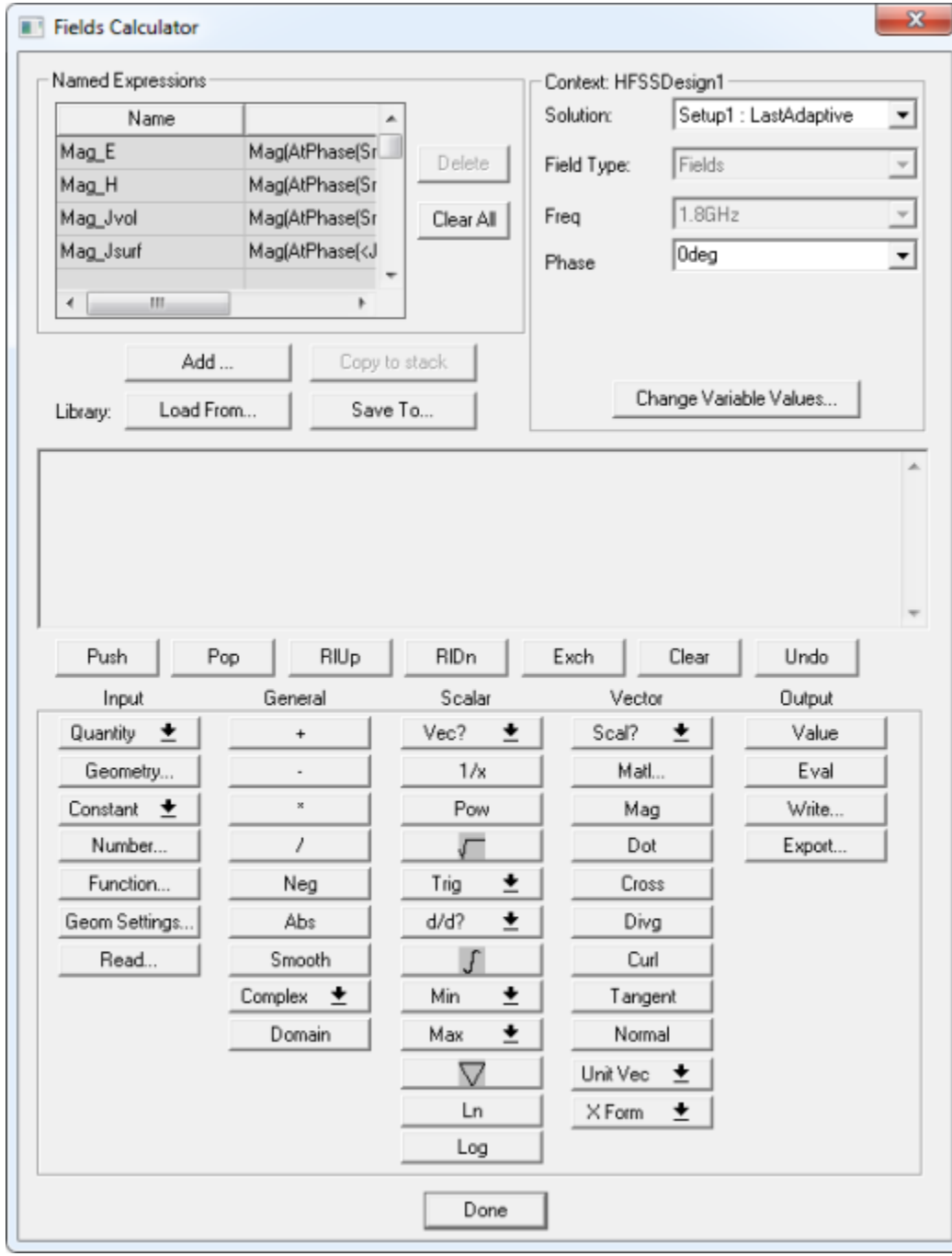
- Click **HFSS** or **HFSS-IE>Fields>Calculator** 

or

- Right-click **Field Overlays** in the project tree, and then click **Calculator** on the shortcut menu.

  The Fields Calculator window appears.

To view information on a command or screen area, click over the button or screen area on the illustration below.



Context Area

The panel at the upper right of the window identifies the context to be used for the calculations. The top line identifies the design. Depending on the design, text entry boxes allow you to select a **Solution**, **Field Type**, **Freq**, **Phase**, **IWavePhi** and **IWaveTheta**. The **IWavePhi** and **IWaveTheta** are available only for incident wave projects in which the wave is defined with spherical coordinates.

The Field Type here is not related to the edit sources. This is a general term among ANSYS EM products (HFSS, Maxwell, and Q3D). Some products have more than one field type for different solution types. If only one Field Type is available the box is grayed out. If the design contains an IE Region, you can select **IE Surface Fields** for J and Q input quantities.

The **Change Variable Values** button opens a **Set Variable Values** dialog. By default it has Use Nominal Design checked. Unchecking the box lets you select another variable value. OK the dialog to accept the selection.

Related Topics

Opening the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

The Calculator Stack

The calculator is made up of a stack of registers. Registers are displayed in the register display area at the center of the calculator window. Each register can hold:

- Field quantities such as the H-field or E-field.
- Functional or constant scalars and vectors.
- Geometries — points, lines, surfaces, or volumes — on which a field quantity is to be evaluated.

To perform a computation on the field solution, you must first load a basic field quantity into a register on the stack. Once a quantity is loaded into a register, it can be:

- Manipulated using mathematical operations such as curls, gradients, cross products, divergences, and dot products.
- Integrated over lines, surfaces, or subvolumes of the solution region — either predefined surfaces, volumes, and lists, or lines, surfaces, and volumes that were defined using the **Draw** commands.
- Exported to a file, allowing you to superimpose saved solutions.

Related Topics

Registers

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook:*** *A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Registers

Calculator registers hold field quantities, numbers, vectors, and geometries. No registers are created until you load something into the calculator; therefore, this part of the window is initially blank. As items are loaded into the calculator, it creates new registers to hold them.

Each register is labeled with its contents as follows:

Vec	Vector quantities, which have both direction and magnitude at each point in space. The x-, y-, and z-components of these quantities are stored in the register.
Scl	Scalar quantities, which have a magnitude only.
Cvc	Complex vector quantities.
Csc	Complex scalar quantities.
Pnt	Points.
Lin	Lines.
Srf	Surfaces.
Vol	Volumes.
SclLin	Scalar value on a line.
VecLine	Vector value on a line.
SclSrf	Scalar value on a surface.
VecSrf	Vector value on a surface.

When examining calculator registers, keep the following in mind:

- To move or delete calculator registers, use the stack commands.
- To save a register to a disk file, use the **Write** command.

Related Topics

Enlarging the Register Display Area

Units of Measure

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Enlarging the Register Display Area

If there are too many registers to fit into the display area, do one of the following:

- Use the scroll bars to view the hidden registers.
- Enlarge the calculator window using the window's borders.

Related Topics

Registers

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Units of Measure

Unless you are prompted specifically for the unit of measure, all measurements should be assumed to be in SI base units, not model units.

Related Topics

Registers

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Stack Commands

Use these commands to manipulate the registers in the calculator stack.

Push

Reloads the quantity in the top register onto the top of the stack, creating a new register. The contents of the top two registers are identical.

Pop

Deletes the top register from the stack.

RlUp

Rolls the top register to the bottom of the stack, moving the other registers up the stack.

RlDn

Rolls the bottom register to the top of the stack, moving the other registers down the stack.

Exch

Exchanges the top two registers in the stack.

Clear

Clears the contents of the stack.

Undo

Use this command to undo the effect of the last operation you performed on the contents of the top register. Successive **Undo** commands act on any previous operations.

Note You cannot undo a simple operation such as loading a field quantity, constant, function, or geometry into the calculator. Instead, use the **Pop** or **Clear** commands to delete these items from the calculator stack.

Related Topics

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Input Commands

Use the following commands to load data onto the top of the calculator stack:

Quantity	Basic field quantities, such as E and H, and simple derived quantities such as volume current. For designs with IE Region, and with IE Surface Fields selected as the Context, you can select J and Q.
Geometry	Geometries such as planes, points, polylines, face lists, and volumes
Constant	Predefined constants such as π, ε_0, and conversion factors between various units of measurement.
Number	Vector and scalar constants, including complex numbers.
Function	User-defined or intrinsic variables
Geom Settings	Number of equally spaced points used to integrate fields and other quantities on a line.
Read	Previously-saved calculator registers containing field quantities.
Output Vars	This button appears only for eigenmode problems. Freq is the only value there. Evaluation for Freq returns a complex value.

These quantities can be manipulated using the **Stack** commands, **General** commands, **Scalar** commands, and **Vector** commands. The results of these calculations can then be examined using the **Output** commands.

Related Topics

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook:*** *A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Quantity Command

The **Input** command loads a field quantity into the top register of the calculator. Phasors in the calculator are peak phasors. The **Poynting** command in the calculator therefore implements the Poynting vector for peak phasors. Calculations which compute either average or instantaneous time domain quantities must adhere to the peak phasor conventions.

The available quantities are:

E	The electric field, E
H	The magnetic field, H
Jvol	The volume current density, J_{vol}
Jsurf	The surface current density, J_{surf}
Poynting	The Poynting vector, defined as 0.5E x H*
LocalSAR	The local Specific Absorption Rate
AverageSAR	The average Specific Absorption Rate
Certification SAR	IEEE standard Specific Absorption Rate certification number. To calculate certification SAR on a specific object (rather than the whole model) proceed as follows: 1. In the Calculator Input area, click the Quantity button and select Certification SAR. Certification SAR is displayed in the calculator stack. 2. In the Calculator, click the Geometry button to display the Geometry dialog. The Geometry dialog displays with the Volume radio button selected, and the available geometries listed. 3. Select the Geometry of interest. This enables the OK button. 4. Click the OK button. 5. This adds the selected Volume geometry to the calculator stack. 6. In the Calculator Output area, press the Value button This prepares the calculation for the selected quantity and volume. 7. Press the Eval button to evaluate. Both the value and location will be shown on the calculator stack.
SurfaceLossDensity	This contains the surface impedance (if any) loss at every node in every triangle. This is calculated as: $$p_s = Re(S \cdot n)$$ where p_s is the surface impedance loss density, S is the Poynting vector on the boundary, and n is the out unit normal of the boundary. To export a REG file containing the surface loss density, place the SurfaceLossDensity in the top register and use the **Write...** command, selecting Reg format.

VolumeLossDensity	The volume loss density *p* is calculated as: $$p_v = \frac{1}{2}Re(E \cdot \tilde{J} + j\omega B \cdot \tilde{H}) = \frac{1}{2}Re(E \cdot \tilde{J} - curlE \cdot \tilde{H})$$ where E is the electric field, $\tilde{J}$ is the conjugate of the volumentric current density, B is the magnetic flux density, and H is the conjugate of the magnetic field. To export a Reg file containing the volume loss density, place the VolumeLossDensity into the top register, and use the **Write...**.command, selecting the Reg format.
SurfaceForceDensity	Surface Forces exist when one side is conductor, but the other is not, or finite conductivity and layered impedance boundary. This is mainly for the purpose of mapping surface force density in HFSS to Workbench Mechanical. For details on the calculations, see the technical notes.
Temp	Temperature.
Displacement	This is value is for use with Workbench when exploring stress feedback.

Related Topics

Using the Fields Calculator

Input Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Geometry Command

The Fields Calculator **Geometry** command opens a dialog that lets you select a geometry to load into the top register of the calculator. Do this to:

- Find the value of derived field quantities on any point, line, surface, or volume.
- Plot quantities directly from the calculator.
- Display a previously defined isosurface, maximum or minimum field point using the **Draw** command.

The following types of geometries are available:

Point - See drawing a point object. Points you draw are listed in the history tree, and in the Calculator **Geometry** dialog when you select Point.

Line- See drawing a line object. Lines you draw are listed in the history tree, and in the Calculator **Geometry** dialog when you select Line. To set the number of points on a line, see Geom Settings .

Surface - Sheet objects and face lists which you can make, (for example of radiation boundaries) are listed under surface in the history tree and in the Calculator **Geometry** dialog when you select Surface.

Due to the ambiguity of the normal vector of a sheet, the result may require a multiplication by (1) or (-1).

Volume - 3D objects, Regions, and object lists of 3D objects including AllObjects are available in the Calculator **Geometry** dialog when you select Volume.

Coord - Coordinate systems are available in the Calculator **Geometry** dialog when you select Coord.

To load a geometry into the calculator:

1. In the Fields Calculator, click **Geometry**.

   The **Geometry** dialog box appears.
2. Select a geometry type.

   A list of all applicable geometries appears.
3. Click the geometry.
4. Click **OK** to load the geometry.

Note Consider a box (Box2) that is completely enclosed in a bigger box (Box1), so that no faces of Box2 are touching any faces of Box1.

Box2 is actually implicitly subtracted from Box1 as is done in our solvers. So Box1 is used as if Box2 were already subtracted from Box1. Volume(Box1) is Box1 excluding Box2, and Surface(Box1) contains faces from both Box1 and Box2.

Related Topics

Using the Fields Calculator

Input Commands

Domain command

Export Command

Geom Settings

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Constant Command

The **Constant** command loads one of these four predefined constants, or conversion constant into the top register of the calculator:

Pi	π
Epsi0	The permittivity of free space, $\varepsilon_0 = 8.85418782 \times 10^{-12}$ C^2/Nm^2
Mu0	The permeability of free space, $\mu_0 = 4\pi \times 10^{-7}$ Wb/Am
c	The speed of light in vacuum, $c = 2.99792458 \times 10^8$ m/s
conversion constant	Displays the **Enter Units Conversion Factor** dialog. This lists a range of Quantities (such as frequency, resistance, and others) along with a list of Units (Hz to Thz, and rps) to convert From and To. The ratio of the Units From to the Units to is displayed for the selected values as **Conversion Factor**.

Related Topics

Using the Fields Calculator

Input Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Number Command

The **Number** command enters one of the following into the top register of the calculator:

Scalar A scalar constant. To enter a constant scalar number:

1. Click **Number**.

   The **Input Number** dialog box appears.
2. Select **Scalar**.
3. Type the scalar value in the **Value** text box.
4. Click **OK** to load the number into the top register.

Vector A vector constant.

To enter a constant vector:

1. Click **Number**.

   The **Input Number** dialog box appears.
2. Select **Vector**.
3. Enter the x-, y-, and z-components of the vector.
4. Click **OK** to load the vector into the top register.

Complex A complex constant. Complex constants are entered in the form $C=A+jB$, where A represents the real part of the constant and B represents the imaginary part.

1. Click **Number**.

   The **Input Number** dialog box appears.
2. Select **Scalar** or **Vector**.
3. Select **Complex**.
4. Enter the real and imaginary components of the number.
5. Click **OK** to load the number into the top register.

Related Topics

Using the Fields Calculator

Input Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Function Command

Any functions you use must be defined prior to using this operation.

Enters one of the following into the top register of the calculator:

Scalar

A scalar function.

To enter a function:

1. Click **Function**.

   The **Function** dialog box appears.
2. Select **Scalar**.
3. Select the function from the list.
4. Click **OK** to load the functional scalar into the top register.

Vector

A vector function, in which the values of the vector's x-, y-, and z-components are given by functions.

To enter a functional vector:

1. Click **Function**.

   The **Function** dialog box appears.
2. Select **Vector**.
3. Select the function from the list.
4. For each component of the vector, click **SetX**, **SetY**, and **SetZ**.
5. Click **OK** to load the functional vector into the top register.

Note

The predefined variables **X**, **Y**, **Z**, **RHO**, **THETA**, **R**, and **PHI** and any functions that you created can be used to define functional scalar and vector quantities.

Use of the Global Coordinate System is assumed. Local coordinate systems are not used.

Related Topics

Using the Fields Calculator

Input Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook:*** *A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Geom Settings Command

Clicking the **Geom Settings** button opens the **Geometric Settings** dialog box. The dialog box allows you to specify the line discretization, the number of equally-spaced points used to integrate fields and other quantities on a line. The default is 1000 points.

Related Topics

Using the Fields Calculator

Input Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Read Command

This command copies the contents of a disk file into the top register. The register must be one that has been saved using the Write output command.

To read in a register:

1. Click **Read**.
2. Use the file browser to specify the register's file name and directory path. A .reg extension is automatically assumed for register files.
3. Click **OK**.

   The contents of the file are copied to the top register in the stack.

Related Topics

Using the Fields Calculator

Input Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

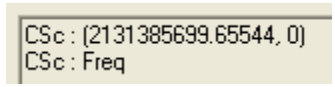
. in pdf format.

The chm version is here:

. in html format.

Output Vars [Input for Eigenmode problems]

This button appears in the **Inputs** column of the Fields calculator only for Eigenmode problems. Freq is the only value listed. After you push Freq to the stack, you can click **Eval** to return a complex value.



Related Topics

Setting the Solution Type

Using the Fields Calculator

Input Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

General Commands

Use these Fields Calculator commands to perform operations on both vector and scalar quantities.

+ (Add)	/ (Divide)	Smooth
-- (Substract)	Neg	Complex
* (Multiply)	Abs	Domain

+ (Add)

Adds the quantities in the top two registers of the calculator.

- (Subtract)

Subtracts the quantity in the top register from the quantity in the second register. The two registers must hold the same type of quantity (both scalar or both vector). You cannot subtract a scalar from a vector (or vice versa).

*** (Multiply)**

Multiplies the quantity in the top register by the quantity in the second register. One of the two registers must contain a scalar value; the other register can be either a scalar or a vector.

/ (Divide)

Divides the quantity in the second register by the quantity in the top register. The second register must contain a scalar value; the top register can be either a scalar or a vector.

Neg

Changes the sign of the quantity in the top register.

Abs

Takes the absolute value of the quantity in the top register.

Smooth

Smooths the quantity in the top register. Because of the numerical solution technique used, field values are not always continuous across the boundaries of the individual elements that make up the finite-element mesh. Smoothing makes the values continuous by taking a weighted average from all of a node's neighboring elements. The weights are based on angles, so elements with larger angles provide larger contributions. In general, use smoothing before plotting a quantity.

Complex

These commands perform operations on a complex quantity in the top register. Complex quantities are indicated by a C at the beginning of the register label. They can be represented in terms of real and imaginary components, or in terms of magnitude and phase:

$$C = A + jB = Me^{j\phi}$$

where:

- A is the real part of the complex number.
- B is the imaginary part of the complex number.
- M is its magnitude, which is equal to $\sqrt{A^2 + B^2}$.
- ϕ is its phase, which is equal to $\text{atan}(B/A)$.

The **Complex** commands let you do the following:

Real	Takes the real part of the complex quantity (A).
Imag	Takes the imaginary part of the complex quantity (B).
CmplxMag	Takes the magnitude of the complex quantity (M). Due to interpolation issues, the sequence of calculations may cause a loss of accuracy. It is best to define the points , separately obtain the value of the real part, then the imaginary part, and use those values to calculate the magnitude and and phase. For the sequence for using the Fields Calculator to obtain the real and imaginary parts, see the procedure here.
CmplxPhase	Takes the phase of the complex quantity (ϕ).
Conj	Takes the complex conjugate of the quantity in the top register. If a complex number is given by $C = A + jB$, its complex conjugate is given by $C^* = A - jB$.

AtPhase	Lets you specify the phase angle, θ, at which an field quantity is evaluated. These quantities can be represented in the form $$\boldsymbol{A}(x, y, z, t) = \boldsymbol{A}(x, y, z)\cos(\omega t + \theta)(x, y, z).$$ where • ω is the angular frequency at which the quantities are oscillating, specified during the solution. • θ(x,y,z) is the phase angle (the offset from a cosine wave that peaks at t=0). Entering the phase angle lets you compute the real part of the field's magnitude at different points in its cycle.
CmplxReal	Converts the real scalar of the top register to the real part of a complex number.
CmplxImag	Converts the real scalar of the top register to the imaginary part of a complex number.
CmplxPeak	Calculates the peak value of a given complex vector. Intuitively, this calculates the maximum magnitude of the equivalent real vector in a waveform.

Domain

This limits a calculation to the volume you specify. The domain filter works for scalars, vectors, complex scalars and complex vectors. This operation requires the top two entries of the stack to be a volume geometry and a numeric field quantity. To do this:

1. Load the field quantity into the top register, and perform any necessary operations on it.
2. Load the volume using the **Geometry** command.
3. Click **Domain**.

The **Domain** command is often used to limit a calculation or plot to the intersection of a surface and an object or group of objects. If you export a domain filtered numeric, points that are filtered out by the domain will not be written out.

Related Topics

Using the Fields Calculator

Export Command

Steps for Calculating the Complex Vector Electric Field

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Steps for Calculating the Complex Vector Electric Field

These are the field calculator steps to obtain the real part, the imaginary part, and the magnitude of the x-directed, y-directed, and z-directed components of the phasor electric field. For each of these vector components, the magnitude should be equal to sqrt(real^2+imag^2), but the need to interpolate values and the calculation sequence means that HFSS does not give this value unless the specified location is directly on a mesh element node.

1. Calculate real part of complex vector electric field (in x, y, and z directions):
   a. Qty > E
   b. Complex > Real
   c. Geometry > Point > fieldcalc_point
   d. Value
   e. Eval
2. Calculate imaginary part of complex vector electric field (in x, y, and z directions):
   a. Qty > E
   b. Complex > Imag
   c. Geometry > Point > fieldcalc_point
   d. Value
   e. Eval

Use the real and imaginary components to manually calculate the magnitude as the sqrt (Real^2+imag^2).

Related Topics

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Scalar Commands

Use these commands to perform operations on scalar quantities.

Vec?	Makes the scalar quantity in the top register a vector component.
1/x	Takes the inverse of the scalar quantity in the top register.
Pow	Raises a scalar quantity to the power you specify.
√ (Square Root)	Takes the square root of the quantity in the top register.

Trig	Takes a selected trigonometric value of the value in the top register of the calculator stack
d/d?	Takes the partial derivative of the quantity in the top register.
∫ (Integral)	Takes the integral of a scalar quantity over a volume, surface, or line.
Min	Computes the minimum of a scalar field quantity on a line, surface, or volume.
Max	Computes the maximum of a scalar field quantity on a line, surface, or volume.
∇ (Gradient)	Takes the gradient of the scalar quantity in the top register.
ln	Takes the natural logarithm (base e) of the scalar quantity in the top register.
log	Takes the logarithm (base 10) of the scalar quantity in the top register

Related Topics

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook:*** *A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Vec? Command

Makes the scalar quantity in the top register a vector component. Choose from the following:

VecX	The x-component of a vector.
VecY	The y-component of a vector.
VecZ	The z-component of a vector.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook:*** *A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

1/x (Inverse) Command

Takes the inverse of the scalar quantity in the top register.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Pow Command

Raises a scalar quantity to the power you specify.

To raise a scalar quantity to a power:

1. Enter the quantity into the calculator.
2. Enter the exponent to which it is to be raised into the calculator.
3. Click **Pow**.

   The results are displayed in the top register.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

 (Square Root) Command

Takes the square root of the quantity in the top register.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Trig

Takes one of the following trigonometric values of the value in the top register of the calculator stack:

Sin	Sine.
Cos	Cosine.
Tan	Tangent.
Asin	Arcsine.
Acos	Arccosine.
Atan	Arctangent.
Atan2	Arctangent squared.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

d/d? (Partial Derivative) Command

Takes the partial derivative of the quantity in the top register:

d/dx	Takes the partial derivative of the quantity with respect to x.
d/dy	Takes the partial derivative of the quantity with respect to y.
d/dz	Takes the partial derivative of the quantity with respect to z.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

∫ (Integral) Command

Takes the integral of a scalar quantity over a volume, surface, or line. The top register must contain a geometry and the second register must contain the scalar quantity to be integrated.

To perform an integration:

1. Load a quantity into the top register of the calculator, and perform any required operations on it.
2. Use one of the **Geometry** commands to load the line, surface, or volume over which the quantity is to be integrated.

Note If you computed the tangent or normal of the quantity to be integrated, you do not have to load a geometry onto the calculator stack. HFSS integrates the tangential or normal component of the quantity over the line on which you computed its tangent, or the surface on which you computed its normal.

3. Choose the ∫ command to integrate the scalar quantity over the geometry.

To find the numerical results of an integration, use the **Eval** command.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Min Command

Computes the minimum of a scalar field quantity on a line, surface, or volume. Two options are available:

Value	Finds the magnitude of the minimum value of the field.
Position	Finds the point where the minimum field value occurs. You can then: • Plot the minimum field value at the point. • Plot basic field quantities at the point. • Load the point into the calculator. • Change the point's location.

These commands operate in the same way as the **Max** commands. Use the **Eval** command to display the actual minimum field value or the coordinates of the point where it occurs.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Max Command

Computes the maximum of a scalar field quantity on a line, surface, or volume. Two options are available:

Value	Finds the magnitude of the maximum value of the field.
Position	Finds the point where the maximum field value occurs. You can then: • Plot the maximum field at the point. • Plot field quantities at the point. • Load the point into the calculator. • Change the point's location.

To compute the maximum field value:

1. Load a field quantity into the calculator, and perform any necessary operations on it. Keep the following in mind:
   - You cannot find the maximum value of a vector quantity. Therefore, make sure that the result is a scalar.

- Before computing the maximum value of a complex quantity, you must find the real part of the quantity using the **Cmplx/Real** or **Cmplx/AtPhase** commands.

2. Load a point, line, or volume into the calculator using one of the **Geometry** commands.
3. Do one of the following:
   - Choose **Max/Value** to compute the maximum field value on the geometry.
   - Choose **Max/Position** to identify the point at which this value occurs.

Use the **Eval** command to display the actual maximum field value or the coordinates of the point where it occurs.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

∇ (Gradient) Command

Takes the gradient of the scalar quantity in the top register.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Ln Command

Takes the natural logarithm (base e) of the scalar quantity in the top register.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Log Command

Takes the logarithm (base 10) of the scalar quantity in the top register.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Vector Commands

Use these commands to perform operations on vector quantities.

Scal?	Replaces the vector in the top register with a scalar quantity whose value is a component of the vector.
Matl	Multiplies or divides the vector field quantity in the top register by a material property, or if you select MassDensity as the material property, produces a scalar that operates like a named variable.
Mag	Takes the magnitude of the vector quantity in the top register. The magnitude of a complex vector is defined to be the length of the real vector resulting from taking the modulus of each component of the original complex vector.
Dot	Takes the dot product of the vector quantities in the top two registers.
Cross	Takes the cross product of the vector quantities in the top two registers.
Divg	Takes the divergence of the vector quantity in the top register.
Curl	Takes the curl of the vector quantity in the top register.

Tangent	Computes the tangential component of a vector quantity along a line
Normal	Computes the normal component of a vector quantity on a surface such as a cutplane or object surface.
Unit Vec	Computes the normal or tangent unit vector. The unit vector is a "wild card" entry. The context is specified at the time of plotting, integrating, or report generation.

Related Topics

Using the Fields Calculator

Scalar Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Scal? Command

Replaces the vector in the top register with a scalar quantity whose value is a component of the vector. Choose from the following:

ScalarX	Returns the x-component of the vector.
ScalarY	Returns the y-component of the vector.
ScalarZ	Returns the z-component of the vector.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Matl Command

Operates on the vector field quantity in the top register of the Fields Calculator based on a material property. At each tetrahedron, the field quantity is operated on by the value of the selected material property — taking the different material attributes of each object into account.

To operate on a vector quantity by a material property:

1. Click **Matl**.

   The **Material Operation** window appears.

2. Select a material property. Available properties are:

Permittivity (epsi)	The permittivity, ε_r.
Permeability (mu)	The permeability, μ_r.
Conductivity	The conductivity, σ.
Omega (w)	The angular frequency, ω. The angular frequency is equal to $2\pi f$, where f is the frequency at which the solution was generated.
MassDensity	This is based on the value of the Mass Density material property. MassDensity is treated like a named expression. Selecting MassDensity disables the Operation radio buttons for Multiply or Divide in the **Material Operations** dialog.

3. For Permittivity, Permeability, Conductivity or Omega, select an operation — **Multiply** or **Divide**. Selecting MassDensity disables these operations. The MassDensity scalar can be used like any other named expression.
4. Choose **OK** to operate on the field quantity by a material property or **Cancel** to stop the operation. If you selected MassDensity and click **OK**, a scalar named expression MassDensity is pushed onto the stack.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook:*** *A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Mag Command

Takes the magnitude of the vector quantity in the top register. The magnitude of a complex vector is defined to be the length of the real vector resulting from taking the modulus of each component of the original complex vector.

With a complex vector on the calculator stack, the **Mag** button returns a nonnegative scalar. In previous software versions, this command returned a complex scalar.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Dot Command

Takes the dot product of the vector quantities in the top two registers.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Cross Command

Takes the cross product of the vector quantities in the top two registers.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Divg Command

Takes the divergence of the vector quantity in the top register.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Curl Command

Takes the curl of the vector quantity in the top register.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

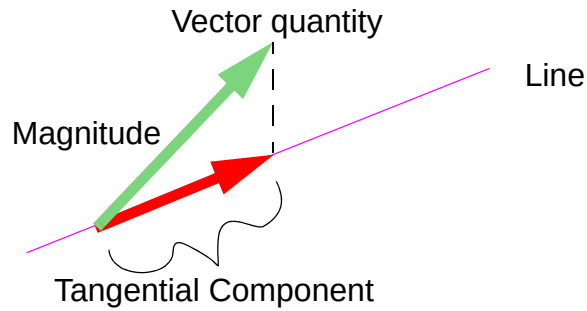
. in pdf format.

The chm version is here:

. in html format.

Tangent Command

.



To take the tangent of a vector:

1. Load a vector quantity into the top register.
2. Load a line into the top register using the **Geometry/Line** command.
3. Click **Tangent**.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

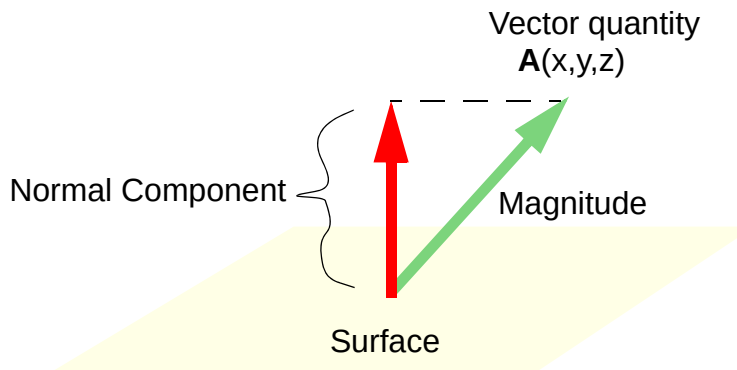
. in html format.

Normal Command

Computes the normal component of a vector quantity on a surface such as a cutplane or object surface. This is the equivalent of taking the dot product of the quantity with the surface's unit normal

$$Normal = \boldsymbol{A}(x, y, z) \bullet \hat{\boldsymbol{n}}$$

vector:



To take the normal of a vector:

1. Load a vector quantity into the top register.
2. Load a surface into the top register using the **Geometry/Surface** command.
3. Click **Normal**.

Note Because surface normals of sheets are not well defined the fields calculator can produce incorrect results if an expression is evaluated on a sheet. To enforce the correct direction of the surface normal of a sheet, a faceted 3D object (such as a box) can be defined such that one of its planar faces is coincident with the sheet. Because surface normals of a valid object are always defined in an outward direction in HFSS, the fields calculator uses the surface normal of the face of the 3D object that is coincident with the sheet.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Unit Vec Command

Computes the normal or tangent unit vector. The unit vector is a "wild card" entry. The context is specified at the time of plotting, integrating, or report generation.

Select from the following:

Tangent	Computes the unit vector tangent to the line specified at the time of plotting, integrating, or report generation based on the context.
Normal	Computes the unit vector normal to the surface specified at the time of plotting, integrating, or report generation based on the context.

CoordSys(X)	Computes the unit vector in the X-dimension of the relative coordinate system in the top register of the calculator stack. Add the relative CS as a geometric object using the **Geometry/Coord** command.
CoordSys(Y)	Computes the unit vector in the Y-dimension of the relative coordinate system in the top register of the calculator stack. Add the relative CS as a geometric object using the **Geometry/Coord** command.
CoordSys(Z)	Computes the unit vector in the Z-dimension of the relative coordinate system in the top register of the calculator stack. Add the relative CS as a geometric object using the **Geometry/Coord** command.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook:*** *A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

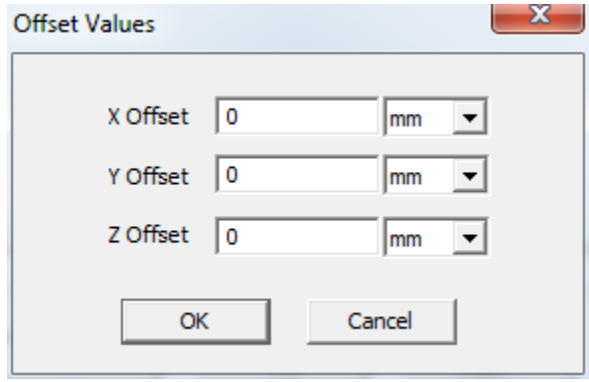
. in pdf format.

The chm version is here:

. in html format.

X Form

Computes the offset based on your selection of Cylindrical or Spherical coordinates and your input of X, Y, and Z offsets for the origin of the new coordinate system. Selecting X Form displays a selection menu for the coordinate system (ToCyclindrical or To Spherical), after which the Offsets dialog displays.



Pressing OK will apply the respective coordinate transformation onto the (complex) vector quantity on the calculator stack. The new expression will be pushed onto the stack. It will be something like CVc : ToCylindrical(<Ex,Ey,Ez>,offset<0mil,0mil,0mil>) or CVc : ToSpherical(<Ex,Ey,Ez>,offset<0mil,0mil,0mil>) respectively.

Related Topics

Using the Fields Calculator

Vector Commands

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Output Commands

Use these commands to compute or evaluate expressions and to output the data in the calculator.

Value command	Computes the value of a field quantity at a point.
Eval command	Numerically evaluates and displays the results of calculator operations.
Write command	Saves the contents of the top register to a disk file.
Export command	Saves field quantities in a format that can be read by other modeling or post-processing software packages.

Related Topics

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Value Command

This computes the value of a field quantity at a point. Use it to find:

- The magnitude of a scalar field quantity at that point.
- The x-, y-, and z-components of a vector field quantity at that point.

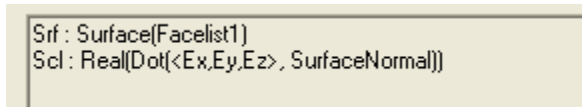
To find the value of a field quantity at a point:

1. Load the field quantity into the top register, and perform any needed operations on it.
2. Load the appropriate point into the calculator using the **Geometry/Point** command.
3. Click **Value**.

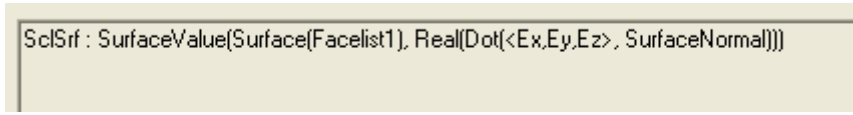
To view the numerical results of this operation, use the **Eval** command.

You can also use the **Value** command to access the intermediate SurfaceValue function.

For example, after inputting an expression for a quantity, such as an E field, and then selecting a surface geometry, the calculator stack displays something like this.



Clicking the **Value** command changes the display to the following, showing the intermediate SurfaceValue function.



In this case SurfaceValue provides the x, y, z, coordinates of the FEM mesh and Lagrangian points so you can use **Write** to generate an .fld file containing an evaluated scalar quantity at those points. Two of the examples in Using the Fields Calculator pdf also show the intermediate SurfaceValue function in practical use.

In general for **Value**:

1. Enter any quantity onto the stack.
2. Enter a volume / surface / line / point onto the stack.
3. Press the **Value** button and you will get an appropriate geometry value on the stack.

Now you can perform suitable operations such as Write, Integrate, etc. For PointValue you can also do **Eval**.

Related Topics

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook:*** *A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Eval Command

This command numerically evaluates and displays the results of calculator operations such as integrations, maximum or minimum field computations, field values at points, and so forth. The quantity to be evaluated must be in the top register. The **Eval** command computes the numerical results of the operation, which replace the contents of the register.

For instance, to find the current around a loop, you must numerically evaluate the following integral for that loop: $\boldsymbol{I} = \oint \boldsymbol{H} \bullet dl$.

Since **H** and **I** are complex quantities, you will need to evaluate the real part of **H** to obtain the real part of **I**, then evaluate the imaginary part of **H** to obtain the imaginary part of **I**. To do this:

1. Load **H** into the calculator using the **Qty** command.
2. Take the real part of **H** using the **Cmplx/Real** command.
3. Load the rectangular loop using the **Geom/Line** command. Create the loop, a closed polyline, to integrate over.
4. Click **Tangent** to get the component of **H** along the line.
5. Take the integral around the loop using the $\int$ command.
6. Click **Eval** to evaluate the integral. The real part of **I** appears in the top register.
7. Repeat this process using the imaginary part of **H** (found with the **Cmplx/Imag** command) to obtain the imaginary part of **I**.

Related Topics

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Write Command

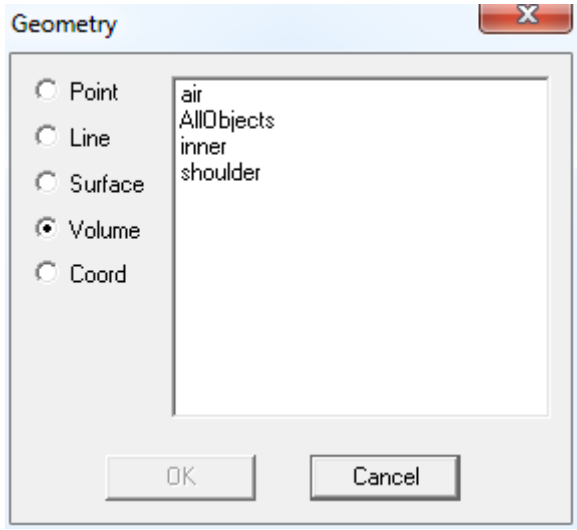
This command saves the contents of the top register to a disk file. Use this command to:

- Save registers for use during a later post-processing session.
- Save a field quantity for use when post processing a different model.

Note Use of the Global Coordinate System is assumed. Local coordinate systems are not used.

To save a register:

1. Click **Write**.
2. If the register includes numeric with a constrained quantity (such as jsurf), you see a dialog that gives a choice of constraining geometries. For example:



3. Select the geometry of interest, and select **OK**.

   This displays a file browser.

4. Use the file browser to specify the register's file name and directory path. A .reg extension is automatically assigned to register files and a .fld extension is assigned to field files. You can choose to save both .reg and .fld files, or either one.

5. Click **OK**.

   The contents of the register are saved to the file you specified.

Related Topics

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Export Command

This command opens the **Export Solution** dialog, from which you can export the field quantity in the top register to a file, mapping it to a grid of points. Use this command to save field quantities in

a format that can be read by other modeling or post-processing software packages. Two options are available for defining the grid points on which to export:

Input grid points from file	Maps the field quantity to a customized grid of points. Before using this command, you must create a file containing the points and units.
Calculate grid points	Maps the field quantity to a three-dimensional cartesian grid. You specify the dimensions and spacing of the grid in the cartesian, cylindrical, or spherical coordinates, with units that you specify. The initial units are taken from the model.

Note Use of the Global Coordinate System is assumed. Local coordinate systems are not used.

To export a field quantity to a customized grid:

1. Load the quantity into the top register for the fields calculator, and perform any operations on it.
2. If desired, load a volume using the **Geometry** command.

   You can use the Domain command to limit the calculation to the volume you specify. If you export a Domain filtered numeric, points that are filtered out by the domain will not be written out.
3. Click the **Export** button in the Fields Calculator.

   This opens the **Export Solution** dialog.
4. Type or select the name of the file in which the field quantity is to be saved in the **Output File Name** text box. You can use the file icon to open the file browser to specify the file name and directory path. A **.reg** extension is automatically assigned to this file.
5. Click either the **Input grid points from file** radio button if you have a created a .pts file containing the grid points, or click the **Calculate grid points** radio button.
   - If you select **Input grid points from file**, either type the name and directory of the file containing the points on which the field is to be mapped, or, click on the file icon and use the file browser to locate the point file (**.pts** extension).

Note The **.pts** file should contain the units to use for the export as shown in this file stub:

```
Unit=mm
-5.5 -5.5 -5.21475
-5.5 -5.5 -5.14425
-5.5 -5.5 -5.07375
-5.5 -5.5 -5.021
```

   - If you select **Calculate grid points** button, you can specify the coordinate system as Cartesian, Cylindrical, or Spherical.

Cartesian: for each grid dimension on **X**, **Y**, and **Z**, enter the Minimum, Maximum, and grid point spacing.

Cylindrical: for each dimension Rho, Phi, and Z, enter the Minimum, Maximum, and grid point Spacing. You can also specify an origin of Offset.

Spherical: for each dimension R, Theta, and Phi, enter the Minimum, Maximum, and grid point Spacing. You can also specify an origin of Offset.

Note When you export fields on a 1D or 2D line/surface from the field calculator, the start and stop values must be the same for one or two of the coordinate system start/stop ranges. If you specify a zero spacing for a dimension, the export uses only the minimum value.

The default coordinate system will be Cartesian. The default offset will be all zeroes. The length units will default to model unit and default angle unit will be degree. At the start the minimum/maximum/Spacing entries are blank. The user entered values are not remembered when the dialog is closed.

6. For larger files, you may want to uncheck the **Include points in output file** box. If you uncheck the box, the file header will include minimum, maximum and spacing information from which you can recalculate the grid points.
7. Click **OK** to export the file.

   The field quantity is mapped to the grid and saved to the file you specified (**.reg** extension.).

Related Topics

Using the Fields Calculator

Input Commands

Domain command

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Calculating Derived Field Quantities

The **Named Expressions** panel displays expressions that can be included in register definitions by name. You can add additional expressions to the Named expression list by creating the expression in the register display area, and the clicking the Add button. This lets you add to the Named expression library.

Click on a named expression to select it. When a named expression has been selected, the **Copy to Stack** button is activated. Click **Copy to Stack** to push the expression on the top of the stack.

When an HFSS design is open and a Solution Setup has been performed, the following predefined named expressions are available:

Expression Name	Expression Definition
Mag_E	Mag(AtPhase(Smooth(<Ex,Ey,Ez>),Phase))
Mag_H	Mag(AtPhase(Smooth(<Hx,Hy,Hz>),Phase))
Mag_Jvol	Mag(AtPhase(Smooth(<JVx,JVy,JVz>),Phase))
Mag_Jsurf	Mag(AtPhase(Smooth(<Jsurfx,Jsurfy,Jsurfz>),Phase))
ComplexMag_E	Mag(CmplxMag(Smooth(<Ex,Ey,Ez>))
ComplexMag_H	Mag(CmplxMag(Smooth(<Hx,Hy,Hz>))
ComplexMag_Jvol	Mag(CmplxMag(Smooth(<JVx,JVy,JVz>))
ComplexMag_Jsurf	Mag(CmplxMag(Smooth(<Jsurfx,Jsurfy,Jsurfz>))
Vector_E	AtPhase(Smooth(<Ex,Ey,Ez>),Phase)
Vector_H	AtPhase(Smooth(<Hx,Hy,Hz>),Phase)
Vector_Jvol	AtPhase(Smooth(<JVx,JVy,JVz>),Phase)
Vector_Jsurf	AtPhase(Smooth(<Jsurfx,Jsurfy,Jsurfz>),Phase)
Vector_RealPoynting	Real(Poynting)
Local_SAR	LocalSAR
Average_SAR	AverageSAR
Surface_Loss_Density	SurfaceLossDensity. See further discussion here.
Volume_Loss_Density	VolumeLossDensity See further discussion here.
Surface_Force_Density	Surface Forces exist when one side is conductor, but the other is not, or finite conductivity and layered impedance boundary. This is mainly for the purpose of mapping surface force density in HFSS to Workbench Mechanical. For details on the calculations, see the technical notes.
Mag_Displacement	Mag(Smooth(<Ux,Uy,Uz>)). Magnitude of displacement, used with Workbench in projects exploring stress feedback.
Displacement_Vector	Smooth(<Ux,Uy,Uz>). Used with Workbench in projects exploring stress feedback.

Related Topics

Named Expression Library

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Named Expression Library

The named expression library in the Fields Calculator provides a way to conveniently calculate frequently used quantities. The library comes with several predefined expressions. You can combine calculator Input commands in any legal fashion, including complex quantities, to produce new named expressions.

- Adding named expressions to the Fields Calculator expression library
- Copying named expressions to the Calculator Stack
- Saving named expressions to a Personal Library
- Loading named expressions
- Deleting named expressions that you added

To add a named expression of your own to the Fields Calculator list:

1. In the register display area, create the expression by using the calculator Input commands.

   You can combine input commands in any legal fashion, including the use of complex quantities. If you select an input command that is not legal for a current operation, you receive an error message.
2. When you finish creating the expression, click **Add** in the **Named Expressions** panel.

   The **Named Expression** dialog box appears.
3. Type a name for the expression in the **Name** text box.

   The new expression is added to the list of named expressions.

To copy a named expressions to the Calculator Stack

- You can scroll through the list, select any desired named expression, and click **Copy to Stack** to move it to the calculator stack, where you can use it to generate calculated outputs.

To delete named expressions that you added:

When the **Named Expression** list contains one or more user-defined expressions, the **Delete** and **Clear All** buttons are active (you cannot delete or clear the predefined named expressions.)

- To delete the selected user-defined named expression, click **Delete**. To delete all user-defined named expressions, click **Clear All**.

To save one or more named expressions for the Fields Calculator to a personal Library:

1. Click the **Save To** button on the Fields Calculator.

   The **Select Expressions for Saving** dialog displays.

2. If any new named expressions exist, you can select one or more to save to a file.
3. Give a file name, and click **OK** to save the file.

To load named expressions for the Fields Calculator from a personal library:

1. From the Fields Calculator, click **Load From**.

   This displays a file browser that you can use to search for existing .clc files.
2. Select the library to load and click **OK**.

   This loads the expression file you have selected.

Related Topics

Calculating Derived Field Quantities

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Exiting the Fields Calculator

Click **Done** to exit the Fields Calculator.

Related Topics

Using the Fields Calculator

In addition to the online help, you can also view a pdf format ***HFSS Fields Calculator Cookbook****: A Brief Primer and Collection of Step-by Step Calculator Recipes for Use in Fields Post Processing*.

. in pdf format.

The chm version is here:

. in html format.

Radiated Fields Post Processing

To analyze the radiated fields associated with a design, define a radiation surface over which the fields will be calculated. Such a surface can be a boundary radiation surface, or a custom radiation surface which you define as a face list. The values of the fields over this surface are used to compute the fields in the space surrounding the device. This space is typically split into two regions — the near-field region and the far-field region. The near-field region exists at less than a wave length from an energy source. The far field is where radiation occurs. See Radiated Fields for the specific equations used in HFSS for calculating the near and far field regions.

You can define a spherical surface over which to analyze the near or far fields by specifying a range and step size for phi and theta. This defines the spherical direction in which radiated fields will be evaluated. You can also draw a line along which to calculate the near fields.

You also may need to edit the Global Material Environment in consideration of the far fields calculation.

Optionally, after defining the radiation surface, HFSS can compute antenna array radiation patterns and antenna parameters for designs that have analyzed a single array element. HFSS models the array radiation pattern by applying an "array factor" to the single element's pattern when far fields are calculated. You set up the array factor information by defining either a finite, 2D array geometry of uniformly spaced, equal-amplitude elements (a regular array) or an arbitrary array of identical elements distributed in 3D space with individual complex weights (a custom array.)

HFSS can also compute antenna parameters, such as the maximum intensity, peak directivity, peak gain, and radiation efficiency. For near-field analysis, HFSS can also compute maximum parameters, such as the maximum of the total E-field and the maximum E-field in the x-direction.

Note When computing near and far fields, keep in mind that you must have defined at least one radiation or PML boundary in the design. At any time you may change the radiation surfaces that HFSS uses when calculating the radiated fields without needing to re-solve the problem, but the radiation-type boundary is still required.

Related Topics

Setting Up a Near-Field Sphere

Setting Up a Near-Field Line

Drawing Non-Model Objects

Computing Maximum Near Field Parameters

Setting up a Far-Field Infinite Sphere

Defining Antenna Arrays

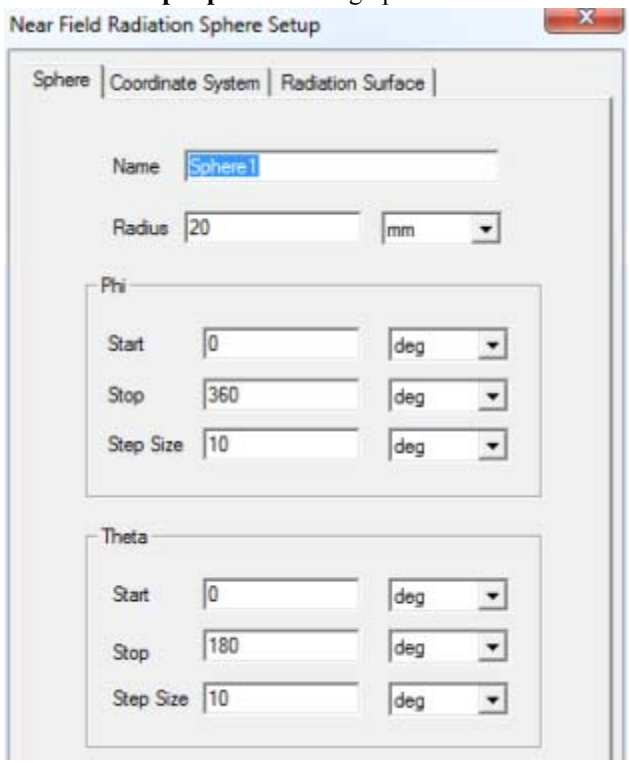
Computing Antenna Parameters

Technical Notes: Radiated Fields

Setting up a Near-Field Sphere

To evaluate near fields on a spherical surface, set up a near-field sphere. Near fields are evaluated at a fixed finite distance from the radiating device. You can use the near-field sphere setup to create near-field reports.

1. To create a near-field sphere setup, click **HFSS>** or **HFSS-IE>Radiation>Insert Near Field Setup>Sphere** to bring up the **Near Field Radiation Sphere Setup** dialog.



2. Use the **Sphere** tab define the **Name**, **Radius**, and sampling of **Phi** and **Theta** for the near-field sphere. The radius is measured from the origin of the sphere's coordinate system, which you specify under the **Coordinate System** tab. You can assign a variable to the radius, and a post-processing variable will often make sense in this context.

   Specify the sphere's sampling in terms of **Start**, **Stop**, and **Step Size** angles given in radians or degrees. To verify your settings, use the **View Sweep Points** button to display a list of the theta and phi sweep points. See Spherical Cross-Sections in the **Technical Notes** for guidelines for setting phi and theta.

   You can use **Save as Defaults** to set the current values as the default for new near-field sphere setups.

3. Use the **Coordinate Systems** tab to specify the orientation of the sphere.

   Use global coordinate system is selected by default, but in some cases the orientation of the

antenna requires the use of a local coordinate system. In this case, select **Use local coordinate system**, and choose a local coordinate system that you created previously in the modeler.

4. Use the **Radiation Surface** tab to select the solved surface from which to calculate radiated fields.

   Use Boundary Radiation Surfaces is selected by default, indicating that the radiated fields will be calculated using the assigned radiation or PML surface. For some models you may find it it more efficient and/or accurate to use an interior surface. In this case, select **Use Custom Radiation Surface**, and choose a face list that you previously created in the modeler.

Notes

You must have defined at least one radiation or PML boundary in the design for HFSS to compute near-field quantities, regardless of which radiation surfaces you instruct HFSS to use when calculating the near fields. You do not need to re-solve the problem if you modify radiation surfaces in the **Near Field Radiation Sphere Setup** dialog.

Related Topics

Selecting Faces

Creating a Face List

Radiated Fields Post Processing

Assigning PML Boundaries

Assigning Radiation Boundaries

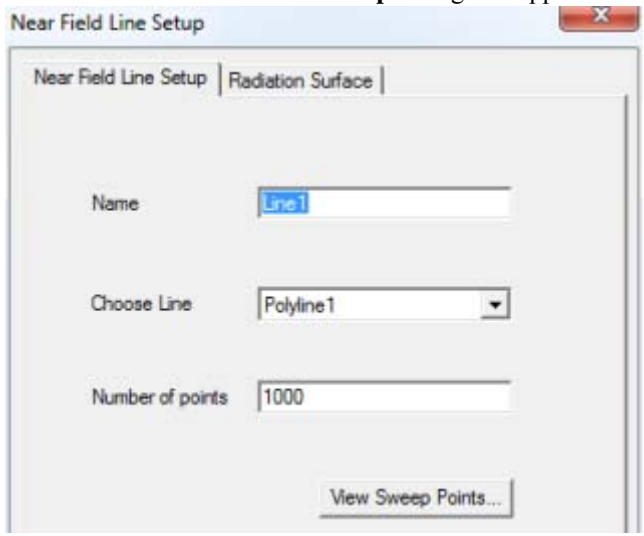
Technical Notes: Spherical Cross-Sections

Setting up a Near-Field Line

To evaluate the near field along a line, set up a near-field line. The near-field line can be a polyline with one or more segments. To plot near-field values along the line, you will select the line object from the **Geometry** list in the **Traces** dialog box when you create a report.

1. Draw a polyline in post-processing mode.
2. Click **HFSS** or **HFSS-IE>Radiation>Insert Near Field Setup>Line**.

The **Near Field Line Setup** dialog box appears.



3. Under the **Near Field Line Setup** tab, type a name for the line in the **Name** text box.
4. Select the polyline along which you want to evaluate the near fields from the **Choose Line** list.
5. Specify the **Number of points** in the line.

   This is the total number of equally spaced points on the line. Specifying points on the line will enable you to plot the near-field values across a normalized distance, that is, to create a value versus distance plot of a near-field quantity on the line.

   You can click the **View Sweep Points** button to view a dialog that lists the points.
6. Click the **Radiation Surface** tab.

   By default, the **Use Boundary Radiation Surfaces** radio button is selected.

   To specify a surface other than an assigned radiation or PML boundary over which to integrate the radiated fields, you must first create a face list. To create a facelist see Creating a Face List. The face list cannot include a face that lies on a PML object.

   If you have created one or more face lists, the **Use Custom Radiation Surface** radio button is enabled.

   a. Select **Use Custom Radiation Surface**.

      This enables the **Choose from existing face list** field.

   b. Select a defined face list from drop down menu.

   HFSS will use the surfaces in the face list as the radiating surfaces when calculating the near fields.
7. Click **OK**.

You must have defined at least one radiation or PML boundary in the design for HFSS to compute near-field quantities, regardless of which radiation surfaces you instruct HFSS to use when calcu-

lating the near fields. You do not need to re-solve the problem if you modify radiation surfaces in the **Near Field Line Setup** window.

Note For parts of the near-field line lying outside of the model region, near-field approximation is calculated. However, if parts of the line lie inside the model region, the model fields are used to compute interpolated values. A section of the near-field line is considered to overlap the model if it lies in the enlarged model region after accounting for symmetry planes.

Related Topics

Assigning PML Boundaries

Assigning Radiation Boundaries

Radiated Fields Post Processing

Computing Maximum Near-Field Parameters

You must have defined at least one radiation or PML boundary in the design for HFSS to compute maximum field data for the near-field region.

1. Right-click the **Sphere** or **Line** icon in the project tree, and then click **Compute Max Parameters** on the shortcut menu.

   The **Select Solution** dialog box appears.
2. Under the **Solutions** tab, select the solution for which you want HFSS to compute the near-field parameters.
3. Under the **Intrinsic Variables** tab, select the solved frequency point at which you want HFSS to compute the near-field parameters.

   The **Max Field Data** window appears, listing the following information:

   Total

   X

   Y

   Z

   Phi

   Theta

   LHCP

   RHCP

   Ludwig 3/X dominant

   Ludwig 3/Y dominant

Note When calculating the maximum far-field values, the distance *r* is factored out of the E-field. Therefore, the units for the maximum field data values are given in volts.

Related Topics

Radiated Fields Post Processing

Assigning PML Boundaries

Assigning Radiation Boundaries

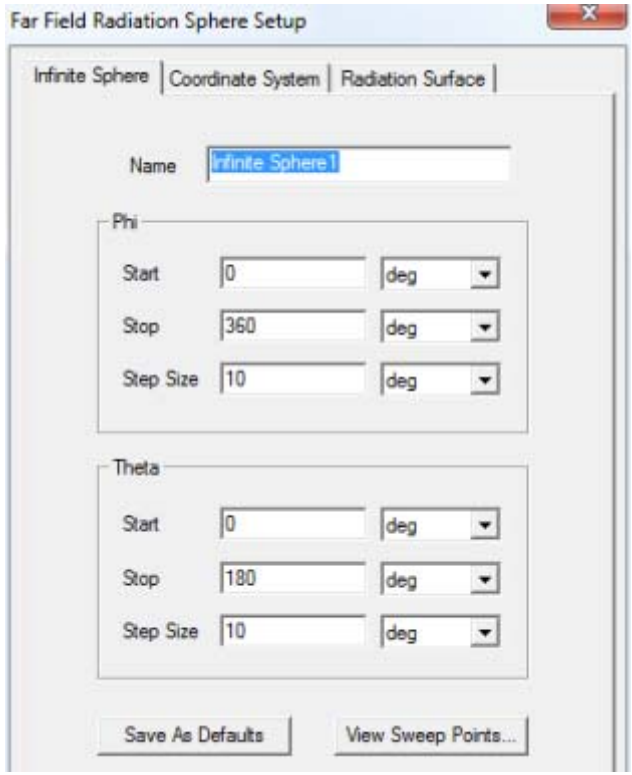
Technical Notes: Maximum Near-Field Data

Setting up a Far-Field Infinite Sphere

To evaluate radiated fields in the far-field region, you must set up an infinite sphere that surrounds the radiating object. To plot far-field values across the sphere, you will select the sphere object from the **Geometry** list in the **Traces** dialog box when you create a report.

1. Click **HFSS** or **HFSS-IE>Radiation>Insert Far Field Setup>Infinite Sphere**.

   The **Far Field Radiation Sphere Setup** window appears.



2. Use the **Sphere** tab define the **Name**, **Radius**, and sampling of **Phi** and **Theta** for the near-field sphere. The radius is measured from the origin of the sphere's coordinate system, which you specify under the **Coordinate System** tab. You can assign a variable to the radius, and a

post-processing variable will often make sense in this context.

Specify the sphere's sampling in terms of **Start**, **Stop**, and **Step Size** angles given in radians or degrees. To verify your settings, use the **View Sweep Points** button to display a list of the theta and phi sweep points. See Spherical Cross-Sections in the **Technical Notes** for guidelines for setting phi and theta.

You can use **Save as Defaults** to set the current values as the default for new near-field sphere setups.

3. Use the **Coordinate Systems** tab to specify the orientation of the sphere.

   Use global coordinate system is selected by default, but in some cases the orientation of the antenna requires the use of a local coordinate system. In this case, select **Use local coordinate system**, and choose a local coordinate system that you created previously in the modeler.

4. Use the **Radiation Surface** tab to select the solved surface from which to calculate radiated fields.

   Use Boundary Radiation Surfaces is selected by default, indicating that the radiated fields will be calculated using the assigned radiation or PML surface. For some models you may find it it more efficient and/or accurate to use an interior surface. In this case, select **Use Custom Radiation Surface**, and choose a face list that you previously created in the modeler.

Note Do not use a sheet-object based face list as the radiation computation surface.

You can use the **Save as Default** to set the current values as a default, and the **Use Defaults** button to use previously saved options.

Note You must have defined at least one radiation or PML boundary in the design for HFSS to compute far-field quantities, regardless of which radiation surfaces you instruct HFSS to use when calculating the far fields. You do not need to re-solve the problem if you modify radiation surfaces in the **Far Field Radiation Sphere Setup** window.

Related Topics

Radiated Fields Post Processing

Global Material Environment

Assigning PML Boundaries

Assigning Radiation Boundaries

Technical Notes: Spherical Cross-Sections

Creating a Face List

Defining Antenna Arrays

Define a regular or custom antenna array when you want HFSS to compute antenna array radiation patterns and antenna parameters for designs that have only a single array element. HFSS models the array radiation pattern by applying an "array factor" to the single element's pattern when far fields are calculated.

The "regular uniform array" geometry defines a finite 2D array of uniformly spaced, equal-amplitude elements. This is a natural specification after phased array applications. The "custom array" geometry defines an arbitrary array of identical elements distributed in 3D space with individual user-specified complex weights.

If you define an antenna array, the antenna setup icon in Project tree changes from a single antenna an array icon.

Related Topics

Radiated Fields Post Processing

Defining a Regular Antenna Array

Defining a Custom Antenna Array

Defining a Regular Antenna Array

A regular antenna array is a finite 2D array geometry of uniformly spaced, equal-amplitude cells with a linear phase shift.

1. Click **HFSS** or **HFSS-IE>Radiation**>**Antenna Array Setup**.

   The **Antenna Array Setup** window appears.
2. Under the **Array Type** tab, select **Regular Array Setup**.
3. Click the **Regular Array** tab.
4. Under **First Cell Position**, enter the xyz-coordinates where the first cell is placed.
5. Under **Directions**, define the lattice vectors:
   a. To the right of **U Vector**, enter the vector coordinates in the **X**, **Y**, and **Z** text boxes along which the cells in the U-direction are placed.
   b. To the right of **V Vector**, enter the vector coordinates in the **X**, **Y**, and **Z** text boxes along which the cells in the V-direction are placed.
6. Under **Distance Between Cells**, enter the distance between cells in the U-direction and the distance between cells in the V-direction in the design units.
7. Under **Number of Cells**, enter the number of unit cells in the U-direction and the number of unit cells in the V-direction.
8. Under **Scan Definition**, specify the scan direction in one of the following ways:
   - Select **Use settings from slave boundary**. If the design includes a master/slave set, you can select this to use that scan angle.
   - Select **Use Custom Scan Angles** to directly enter the scan angles in degrees, in the radiation coordinate system in the **Theta** and **Phi** text boxes.
   - Select **Use Differential Phase Shift** to enter the phase difference in degrees, between adjacent elements, in the **In U direction** and **In V direction** text boxes.
9. Click **OK**.

   The array factor will be applied, using the information you specified, when far fields are calculated.

Related Topics

Radiated Fields Post Processing

Technical Notes: Array Factor Calculation

Technical Notes: Regular Uniform Arrays

Technical Notes: Scan Specification for Regular Uniform Arrays

Defining a Custom Antenna Array

A custom antenna array is a an arbitrary array of identical elements distributed in 3D space with individual user-specified complex weights. The array is defined in a text file that includes the element positions, voltage amplitude weights, and phases. See Custom Arrays in the Technical Notes for examples of custom array geometry text files.

Note For custom antenna arrays, phases should be specified in radians.

1. Click **HFSS** or **HFSS-IE>Radiation>Antenna Array Setup**.

   The **Antenna Array Setup** window appears.
2. Under the **Array Type** tab, select **Custom Array Setup**.
3. Click the **Custom Array** tab.
4. Click **Import Definition**.

   The **Open** dialog box appears.
5. Follow the procedure for opening a file. Select **.txt** as the file type. When you are finished, click **Open**.
6. Optionally, review the definition in the text file by clicking **View Definition** under the **Custom Array Setup** tab.
7. Click **OK**.

   The array factor will be applied, using the information specified in the text file, when far fields are calculated.

Related Topics

Radiated Fields Post Processing

Technical Notes: Custom Arrays

Technical Notes: Array Factor Calculation

Computing Antenna Parameters

You must have defined at least one radiation or PML boundary in the design for HFSS to compute antenna parameters and maximum field data for the far-field region. You can Export the calculated antenna parameters and maximum field data to a file. You can also Export Fields in the .csv format. This can be useful for situations when want to share the fields without sharing the model geometry.

1. To select the radiation setup from the Project tree, right-click the **Infinite Sphere** icon in the project tree under Radiation, and then click **Compute Antenna Parameters** on the shortcut

menu.

Or, to select the radiation setup from a dialog, click **HFSS** or **HFSS-IE>Radiation>Compute Antenna Max/Params**.

The **Select Radiation Setup** dialog appears.

After you have selected the setup by one of these two methods. the **Antenna Parameters** dialog box appears.

2. Under the **Solutions** tab, select the solution for which you want HFSS to compute antenna parameters.
3. Under the **Intrinsic Variables** tab, select the solved frequency point at which you want HFSS to compute antenna parameters.

   The **Antenna Parameters** window appears. If the design includes ports, the following antenna parameters are listed:

   Maximum intensity (Max U)

   Peak directivity

   Peak gain

   Peak realized gain

   Radiated power

   Accepted power

   Incident power

   Radiation efficiency

   Front to Back Ratio

   Decay Factor

Warning The computed values of max U and peak directivity depend on the user-determined set of aspect angles chosen for the computation of the radiated fields. If this set does not encompass the actual peak intensity of the radiated pattern, the displayed results for these three parameters will be inaccurate.

Note Accepted Power is computed from the raw S-parameter data. Post-processing operations are excluded from the calculation, for example, renormalized S-parameters.

Note It is important to understand the impact of passive ports (for Transient Network designs or models containing an Array with passive ports) on antenna parameters. For accepted power calculations, passive ports are not included when computing the total power passing through the union of all port surfaces. This means that the passive ports can be viewed as a loss mechanism for the device and it is not equivalent to viewing the passive ports as active ports with zero excitations.

Gain and realized gain are very close to each other when the antenna is matched. However, for designs with a multi-port antenna, gain and realized gain can be different if incident and accepted power are different. There can be a small reflection at the ports and accepted power may still be small if energy injected in one port exits the model through a different port. A review of the S-matrix can show this to be the case. This does not happen in a single port antenna, which is what most users base their expectations on.

When global material environment is a lossy material, antenna parameters such as gain and radiated power are no longer unique because the radiated power depends on the location of the radiation surface since power is lost as the wave travels in the background material.

If the design does not have ports, the following antenna parameters are listed:

Maximum intensity (Max U)

Peak directivity

Radiated power

4. Click **More** to view the following maximum far-field data:

Total

X

Y

Z

Phi

Theta

LHCP

RHCP

Ludwig 3/X dominant

Ludwig 3/Y dominant

Note When calculating the maximum far-field values, the distance *r* is factored out of the E-field. Therefore, the units for the maximum field data values are given in volts.

Related Topics

Radiated Fields Post Processing

Exporting Antenna Parameters and Maximum Field Data

Technical Notes: Antenna Parameters

Technical Notes: Maximum Far-Field Data

Add Trace Characteristics

Exporting Antenna Parameters and Maximum Field Data

The **Antenna Parameters** dialog displays the calculated antenna parameters and Maximum Field data for a setup. The dialog also includes a buttons to **Export** antenna parameters and to **Export Fields**. The fields can be exported in the.csv format and imported into reporter as a table.

To export the antenna parameters to a text file:

1. Click the **Export** button on the **Antenna Parameters** dialog.

   This displays a file browser.
2. Specify the file name and location (or accept the defaults).
3. Click **Save**.

   This saves the text file and closes the browser.

To export the maximum field data to a comma separated format file:

1. Click the **Export Fields** button on the **Antenna Parameters** dialog

   This displays a file browser
2. Specify the file name and location (or accept the defaults.
3. Click **Save**.

   This saves the comma separated text file and closes the browser.

The first line of the file describes the contents of each succeeding row by column.

Far fields format:

```
Index, Phi(rad), Theta(rad), rEPhi(mag ang), rETheta(mag ang)
```

Near fields format in Cartesian Coordinate System:

```
Index, X, Y, Z, Ex(real, imag), Ey(real, imag), Ez(real, imag),
Hx(real, imag), Hy(real, imag), Hz(real, imag)
```

Near fields format in Spherical Coordinates System:

```
Index, Theta, Phi, Er(real, imag), Etheta(real, imag),
Ephi(real, imag), Hr(real, imag), Htheta(real, imag),
Hphi(real, imag)
```

Related Topics

Radiated Fields Post Processing

Computing Antenna Parameters

Technical Notes: Antenna Parameters

Technical Notes: Maximum Far-Field Data

Plotting the Mesh

Before or after the solution is complete, you can plot the finite element mesh on surfaces or within 3D objects.

1. Select a surface or object to create the mesh plot on or within.

   If it does not exist, create it.

2. Click **HFSS** or **HFSS-IE>Fields>Plot Mesh**.

   The **Create Mesh Plot** dialog appears.

3. Select field type for mesh plot.

   For initial mesh plot, please select "Fields" as field type.

   For FEM or Dielectric IE mesh plot after adaptive passes, please select "Fields" as field type.

   For Metallic IE Surface mesh plot after adaptive passes, please select "IE Surface Fields" as field type.

4. Select the solution to use from the drop down list and click **Done**.

   The mesh appears on the surface or object you selected. An icon for the mesh also appears in the **Project** tree under **Field Overlays --Mesh Plots**.

   If a solution is ongoing, you can select the Mesh Plots icon in the in the Project tree, right-click to display the shortcut menu, and click **Update Plots**. This updates the mesh plot to latest data available. After the last adaptive pass, the Mesh plot is automatically updated.

   If a mesh includes seeding, these effects appear at the start of the adaptive passes. Any mesh adaptation at the start of a sub sequent pass is not plotted until after that pass is completed. This delay ensures that mesh plots and actual solutions remain consistent with each other.

   You can modify an existing plot by selecting the plot and changing the properties.

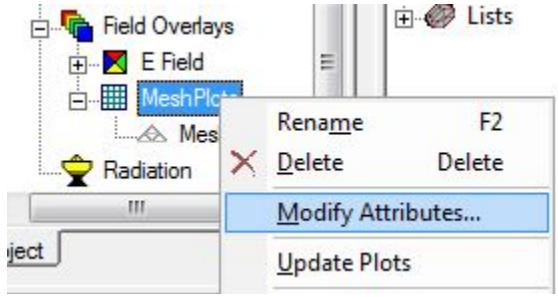
Related Topics

Setting Mesh Plot Attributes

Setting Mesh Plot Attributes

1. Select the Mesh Plots icon under Field Overlays in the Project tree, right-click, and select

Modify Attributes to display the **Mesh Plots** dialog.



You can also click **HFSS** or **HFSS-IE**>**Field Overlays**>**Modify Plot Attributes**  , after which the **Select Folder** window appears. Select the folder containing the mesh plot you want to modify, and then click **OK** to display the **Mesh Plots** dialog.

For Mesh plots, the following attributes can be modified:

Plot	A drop down list of available plots.
Scale Factor	The size at which the tets are displayed. Scaling may let you analyze particular situations better. For example, a scale factor of 80% draws the tetrahedra at 80% of their original size. Use the **Scale factor** slider to increase (move to the right) or decrease (move to the left) the percentage of the tetrahedra size.
Transparency	The degree of transparency for the tets. This is useful for viewing objects or plots behind the current plot. Use the **Transparency** slider to increase (move to the right) or decrease (move to the left) the transparency of the plot.
Mesh type	Whether to display the tets as wire frame or shaded, and whether to Add Grid.
Mesh Color for Line and Fill	The color for the tet edge lines and fill. Clicking the button for each displays a color selection dialog.
Surface Only	Whether to plot the surface only, or all tets inside selected objects.
Real Time	Whether to show changes to a mesh in real time. If this option is cleared, click **Apply** when you want to see the changes

2. By default the **Plot Quality** selection is Normal. You can use the drop down menu to select Coarse, Normal, Fine, or Very Fine. The higher the resolution, the more memory used.

   Higher quality selections permit visualization of curvilinear mesh elements. With the **Course**

setting, facets to not snap to the midpoint edge.

Normal setting displays snapping to the midpoint edge.

Fine setting displays more of the curvilinear tets.

Very fine gives the most accurate display, though using the most memory.

3. Click the **Save as default** button if you want the tab's settings to apply to mesh plots created after this point.
4. Click **Close** to dismiss the dialog box.

Related Topics

Setting a Plot's Visibility

Plotting the Mesh

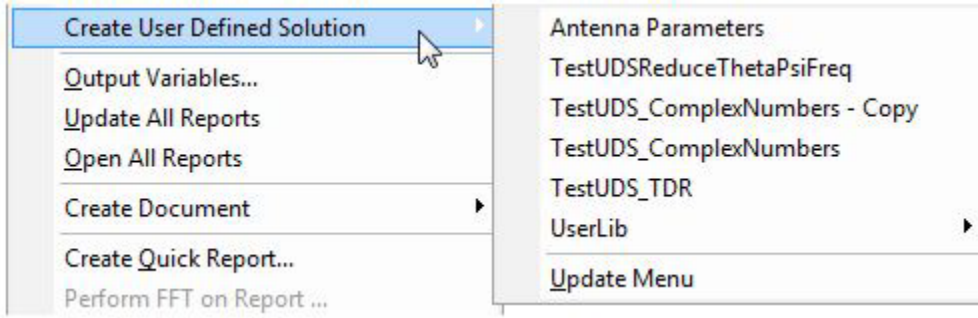
Plotting Field Overlays

User Defined Outputs: Introduction

User defined outputs (UDOs) allow users to define calculations through IronPython scripts or any .NET language (and used by the IronPython script). The UDO scripts need to be in the **UserDefinedOutputs** directory under either **syslib, userlib** or **Personallib** with any directory structure needed for organization (the **Lib** directory name is special and its purpose will be explained later on in the document)

The UDO scripts that are placed in syslib/UserDefinedOutputs, userlib/UserDefinedOutputs or Personallib/UserDefinedOutputs become available to the user to create "User Defined Solutions" through the **Results>Create User Defined Solution** menu.

Use **Results>Create User Defined Solution>Update Menu** to refresh the menu to include the new UDO scripts that might have been copied to syslib, userlib or Personallib, or exclude them if they have been deleted, after the launch of desktop. Once the user-defined-solution is created, the solution and the calculations defined by UDO become available in Reporter as any other quantities in a new "User Defined" report type.



Named Probes and Properties in User Defined Outputs

Computation of Traces Based UDO Calculations

Dimensions Reduction by UDO Calculations

Dynamic Probes

Related Topics

User Defined Outputs: Python Script API

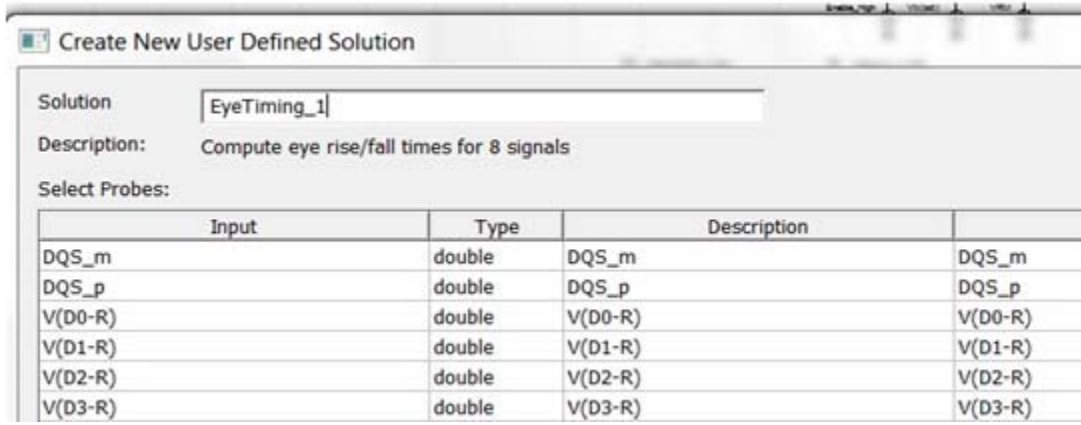
User Defined Outputs: Script Organization

Desktop Scripting with IronPython

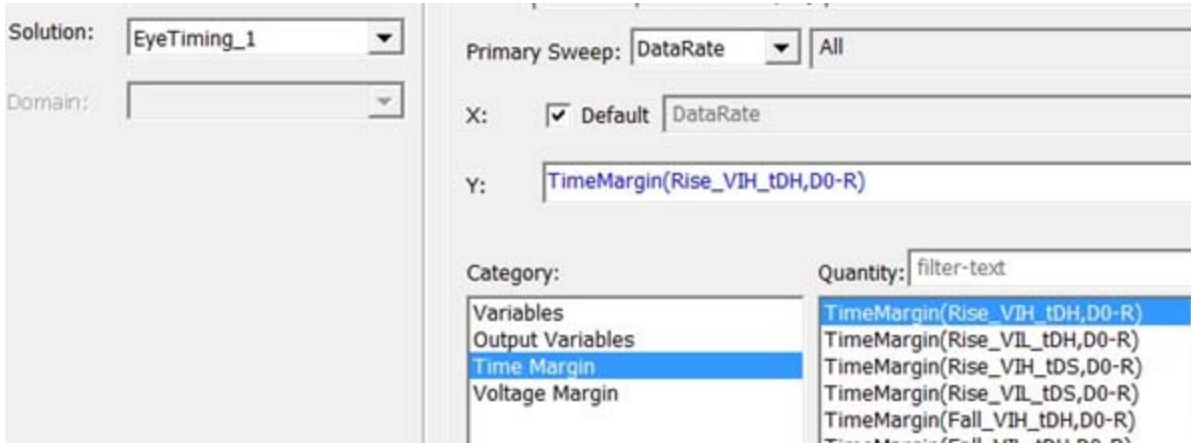
Named Probes and Properties in User Defined Outputs

UDOs allow processing data across traces, solutions and report types. A UDO specifies the named probes and properties for which user selects/enters the values at the time of creation of user defined solution. Probes are very similar to traces except that the user selects the values of only intrinsic variables for probes. The values of design/project variables are selected when a trace is created based upon the user defined solution in reporter.

For example, you could create a user defined solution called EyeTiming_1.



You can then access this solution in the Reporter.



Related Topics

User Defined Outputs: An Introduction

Computation of Traces Based UDO Calculations

Dimensions Reduction by UDO Calculations

Dynamic Probes

Computation of Traces Based UDO Calculations

When traces that are based upon UDO outputs are computed, the data for probes is computed and passed to the UDO script for each design variation. Along with the probe data, the values of properties entered by user are also passed. The information about the UDO calculations that need to be computed is also made available. The UDO then performs the computation and passes the results to reporter. Note that UDOs can compute and pass back more calculations than have been requested at that point of time. This allows UDOs to compute a set of calculations that take almost same amount

of computational resources as any one calculation in that set and cache that with reporter. *When those calculations are subsequently plotted by user, reporter will use the cached results instead of invoking the computation on UDO.*

Related Topics

User Defined Outputs: An Introduction

Named Probes and Properties in User Defined Outputs

Dimensions Reduction by UDO Calculations

Dynamic Probes

Dimensions Reduction by UDO Calculations

The probes in a UDO can have heterogeneous dimensions of data, for example, one probe in a UDO can have data that is a function of n intrinsic variable, while another probe in same UDO can have data that is function of m intrinsic variables, with n and m potentially being different. UDOs allow reducing any number of these intrinsic variables, for example in the above example UDO calculations can be function of any number of intrinsic variables including not being function of any intrinsic variable at all. UDO calculations can also be a function of an intrinsic variable that none of the probes is function of. **The only restriction is that Freq cannot be reduced if any of the probes are on a Fields report type.**

Related Topics

User Defined Outputs: An Introduction

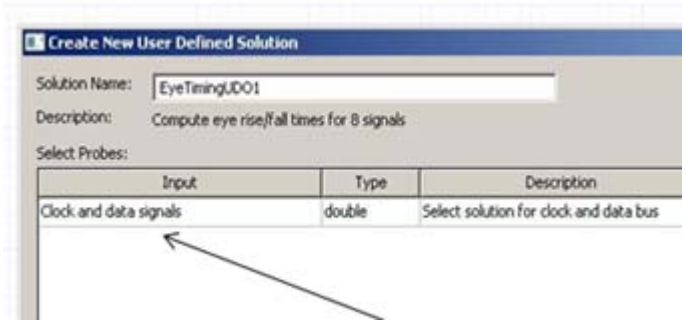
Named Probes and Properties in User Defined Outputs

Computation of Traces Based UDO Calculations

Dynamic Probes

Dynamic Probes

In addition to named probes and properties, UDOs can specify named dynamic probes. The difference between probes and dynamic probes is that while the end user of UDO specifies the complete trace definition for probe, the expression for dynamic probe is specified by UDO code itself and not by end user. This allows UDOs to access the data for probes without requiring the end user to enter each individual probe. For example a UDO can access data for a huge S matrix for 100 port design without having the end user enter the probe information for each of those 10,000 quantities. Each dynamic probe is associated with a named probe that is entered by user, and information about solution, context and intrinsic variables is used from selected probes; however multiple dynamic probes can be associated with the same user selected probe. The dynamic probes are enquired from UDOs at the time of trace computation and not at the time of creation of user defined solution.



This means that you select solution, context, values of intrinsic variables just once, and the same information is used (in this case) for all clock and data signals. The expression for those signals comes from the UDO code.

Related Topics

User Defined Outputs: An Introduction

Named Probes and Properties in User Defined Outputs

Computation of Traces Based UDO Calculations

Dimensions Reduction by UDO Calculations

Dynamic Probes

User Defined Outputs: Python Script API

A User Defined Output (UDO) extension is implemented as an IronPython script that defines a class with a specific name: **UDOExtension** which derives from a specific base class **IUDOPlug-inExtension** and implements its abstract methods.

Related Topics

User Defined Outputs: An Introduction

User Defined Outputs: Script Organization

UDO Extension IMPLEMENTATION

The purpose, argument list and expected return types for each of the **IUDOPluginExtension** abstract methods, which the UDO author is expected to implement are described below.

Related Topics

User Defined Outputs: Python Script API

Import Statements

The base class to be used and the types it uses in turn are contained in .NET assemblies. The use of these requires that the assemblies be imported into the UDO script: the following import statements should be added to the top of the python script:

```
from Ansys.Ansoft.ModulePluginDotNet.Common.API import *
from Ansys.Ansoft.ModulePluginDotNet.Common.API.Interfaces import *
from Ansys.Ansoft.ModulePluginDotNet.UDO.API.Interfaces import *
from Ansys.Ansoft.ModulePluginDotNet.UDO.API.Data import *
```

Related Topics

UDO Extension IMPLEMENTATION

UDOExtension Class

The UDO itself should be implemented as an IronPython class called ***UDOExtension*** which **must** derive from the ***IUDOPluginExtension*** abstract base class **(**from the **Ansys.Ansoft.ModulePlug-inDotNet.UDO.API.Interfaces** namespace**)**.

Note that power users could derive a class hierarchy tuned toward a specific type of UDOs and that they can derive from their own base classes. The only requirement is that directly or indirectly, the UDO class must derive from ***IUDOPluginExtension.***

Example:

```
def BaseClassUDO ((IUDOPluginExtension):
#base class implementation
        …
 def UDOExtension ((BaseClassUDO):
  #UDO class implementation
        …
```

Related Topics

UDO Extension IMPLEMENTATION

IUDOPluginExtension Abstract Class

The implementation of the IUDOPluginExtension class will be described in this section using a simple UDO example that expects a single probe and reduces its dimension returning as its outputs, the max, min and average of its input probe data. The script in its entirety will also be listed later on.

Required functions:

The IUDOPluginExtension abstract class declares the following abstract methods that must be implemented in the UDOExtension class or one of its base classes. Not implementing any of these methods will result in a run-time error and a non functioning UDO. The UDS refers to user Defined Solution parameters.

GetUDSName()

GetUDSDescription()

GetUDSSweepNames()

GetCategoryNames()

GetQuantityNames(string categoryName)

GetQuantityInfo(string quantityName)

GetInputUDSParams(List<UDSProbeParams> udsParams,

GetDynamicProbes(List<UDSDynamicProbes> dynamicProbes);

Compute(IUDSInputData inData,

Related Topics

UDO Extension IMPLEMENTATION

GetUDSName()

- **Purpose**: Return a string that is used as a prefix for all solution instances created using this UDO.
- **Returns**: string.

Example:

```
def GetUDSName(self):
    return "MinMaxAvg
```

Related Topics

IUDOPluginExtension Abstract Class

GetUDSDescription()

- **Purpose**: Returns a description for the UDO, its purpose etc. This is used in multiple UDO related dialogs in the application to describe the UDO.
- **Returns**: string.

Example:

```
def GetUDSDescription(self):
    return "Sample UDO for dimension reducing quantities
```

Related Topics

IUDOPluginExtension Abstract Class

GetUDSSweepNames()

- **Purpose**: Returns a list of sweep names to be used for the solution generated by the UDO. These will appear in the sweeps list displayed in the standard reporter dialog when used to create reports from the solution generated by the UDO.
- **Returns**: list of strings. If the UDO outputs have no sweeps, return the empty list [].

Example:

```
# Returns list of sweeps names
# We have no sweeps as we reduce them.
 def GetUDSSweepNames(self):
      return []
```

Related Topics

IUDOPluginExtension Abstract Class

GetCategoryNames()

- **Purpose**: The outputs that the UDO solution provides/generates can be classified into multiple categories (like how the application does as displayed in the report creation dialog). These will be listed in the categories box in the dialog when creating reports from the UDO generated solution data.
- **Returns**: list of strings.

Example:

```
def GetCategoryNames(self):
    return ["UDOOutputs"]
```

Related Topics

IUDOPluginExtension Abstract Class

GetQuantityNames(string categoryName)

- **Purpose**: For each of the category names returned from the *GetCategoryNames* method, this function is called to return a list of quantities to be organized under that category name. **Note that the quantity names must be unique across the categories: that is, no two categories can have quantities with the same name.**
- **Parameters**:
  - categoryName (input python string) - category name.
- **Returns**: python list of strings.

Example:

```
# returns a list of quantity names for the supplied category name
def GetQuantityNames(self, catName):
    if catName == "UDOOutputs":
        return ["min_val", "max_val", "avg_val"]
    else:
        return []
```

Related Topics

IUDOPluginExtension Abstract Class

GetQuantityInfo(string quantityName)

- **Purpose**: For each quantity that the UDO creates, it must also describe the quantity (unit and other details). This method is called for each quantity name (across all categories) as returned from an earlier call of the *GetQuantityNames* method.
- **Parameters**:
  - quantityName (input string) - quantity name.
- **Returns**: Object of type QuantityInfo.

Example:

```
# Returns an instance of QuantityInfo for the qtyName supplied or None if such a
#   quantity could not be found
 def GetQuantityInfo(self, qtyName):
    # All the quantities we have are simple doubles
    # we can leave them unitless
    return QuantityInfo(Constants.kDoubleParamStr)
```

Related Topics

IUDOPluginExtension Abstract Class

GetInputUDSParams(List<UDSProbeParams> udsParams,

IPropertyList propList,
List<UDSProbeParams> userSelectionForDynamicProbes)

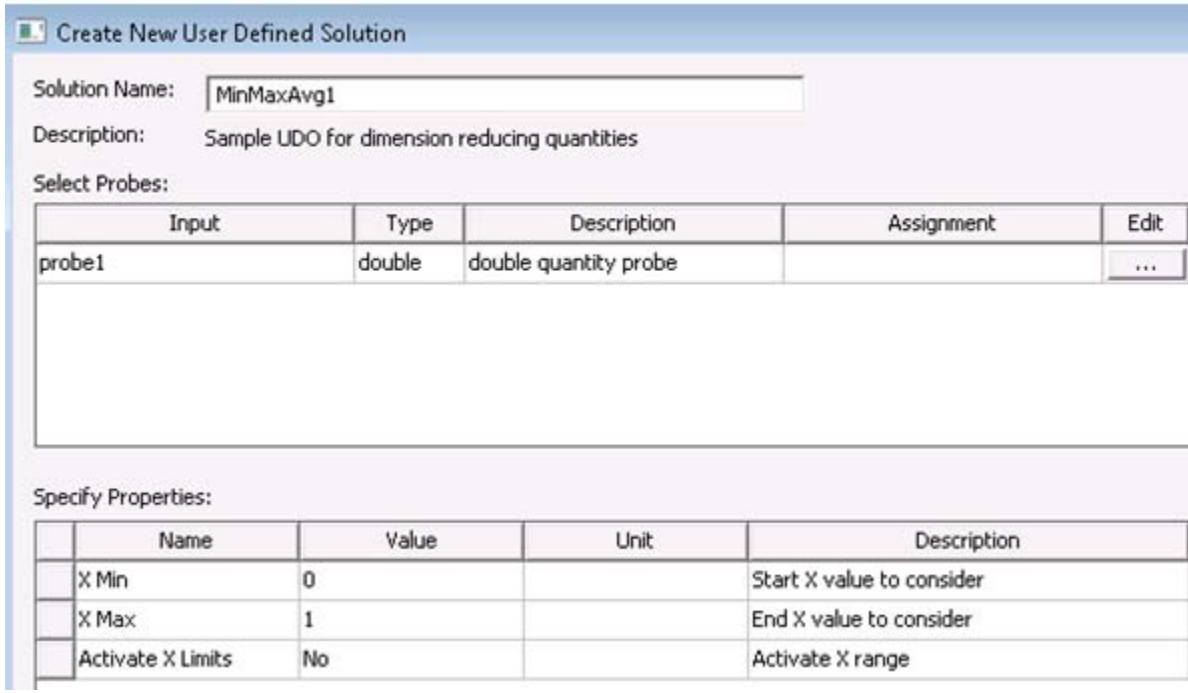
- **Purpose**: This is the main definition part of the UDO. The supplied arguments are used to populate details of the parameters to which the UDO user will specify value, specify the probe names and their types as well as the dynamic probe selections.
- **Parameters**:
  - udsParams - .NET list of UDSProbeParams objects: The UDO script is expected to add one instance of UDSProbeParams for each probe definition it wants displayed. The UDO user will, when creating the UDO solution assign a matching quantity to each such probe.
  - propList - IPropertyList object: The propList object is used to add properties that should be displayed to the user for data collection. These properties with the user supplied values will be returned to the UDO script in the Compute methods.
  - userSelectionForDynamicProbes - .NET list of UDSProbeParam objects.
- **Returns**: boolean: True on success, False on failure.

Example:

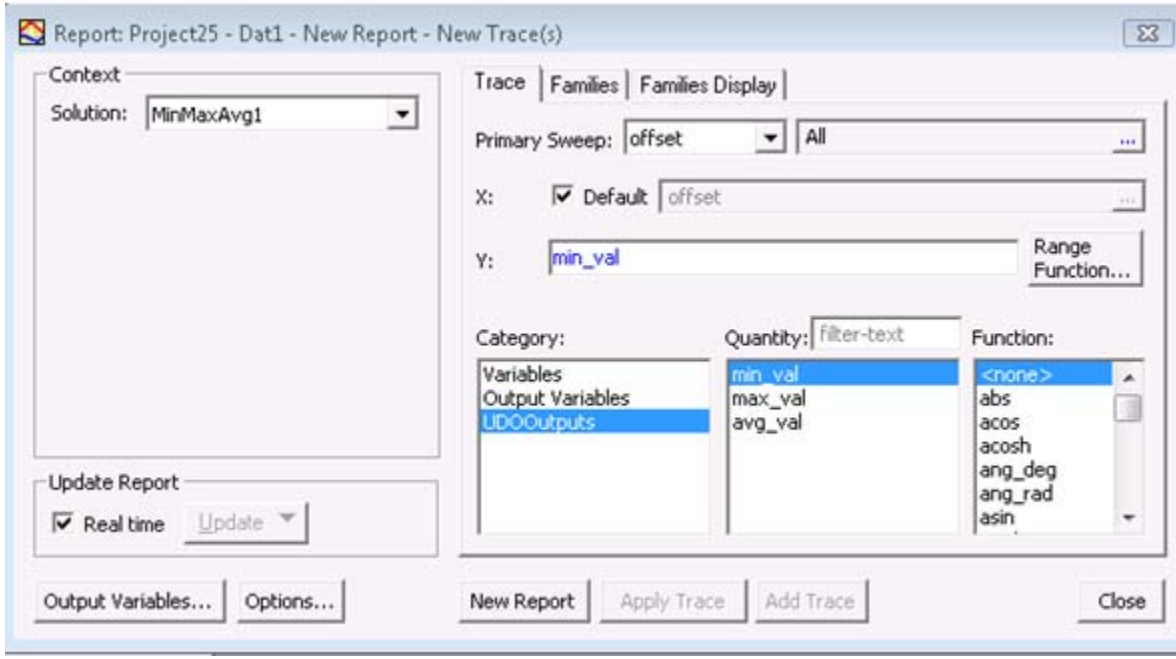
```
# Returns list of UDSParams and list of dynamic properties
# Adds setup time properties to the propList
def GetInputUDSParams(self, udsParams, propList, userSelectedDynamicProbes):


    # Add the probes. We need only one double quantity
    param1 = UDSProbeParams("probe1",
                    "double quantity probe",
                    Constants.kDoubleParamStr,
                    "", "")
    udsParams.Add(param1)
```

```
# Add the properties we want the user to supply
# In this case, we will ask for a start/end range for
# X parameters. Since we cannot reasonably provide defaults
# as we have no idea what the sweep limits will be, we will
# also ask if the limits are to be activated.
prop = propList.AddNumberProperty("X Min", "0")
prop.Description = "Start X value to consider"

prop = propList.AddNumberProperty("X Max", "1")
prop.Description = "End X value to consider"

# For menus, the first option is the default.
prop = propList.AddMenuProperty("Activate X Limits", ["No", "Yes"])
prop.Description = "Activate X range"

return True
```

The above function results in the following dialog when you click **Reports>Create User Defined Solution**. The mapping from the UDSParams and the properties to the GUI elements should be unambiguous. The name and description of the UDS are also displayed in this dialog.



When a report is created from the UDO dialog, the category/quantity names specified by the UDO are used (as shown below).



Related Topics

IUDOPluginExtension Abstract Class

GetDynamicProbes(List<UDSDynamicProbes> dynamicProbes);

- **Parameters**: dynamicProbes - .Net list of UDSDynamicProbes objects. Output parameter.
- **Returns**: True on success, False on failure.

Example:

```
# Returns list of UDSParams and list of dynamic properties
 # output UDSDynamicProbeCollection probes
 def GetDynamicProbes(self, probes):
     pass
```

Related Topics

IUDOPluginExtension Abstract Class

Compute(IUDSInputData inData,

IUDSOutputData outData,

IPropertyList propList,

IProgressMonitor progressMonitor)

- **Purpose**: This is the main computation method which generates the data for the quantities that make up the UDO solution.
- **Parameters**:
  - ***inData*** - UDSInputData object: Used to get the input probe data.
  - ***outData*** - UDSOutputData object: Used to set the UDO solution quantity and sweep data.
  - ***propList*** - IPropertyList object: Used to get the user entered values for each of the properties defined during the *GetInputUDSParams* call.
  - ***progressMonitor*** - IProgressMonitor object. This can be used to set progress for long running calculations, check for user initiated abort etc.
- **Returns**: True on success, False on failure **.**

The data is received from UI using ***IUDSInputData*** API**.** It is processed and the result data is sent to UI using ***IUDSOutputData*** API.

Example:

```
# IUserDefinedSolutionHandle API implementation.
# Calculates output values and sets them using IUDSInputData/IUDSOutputData API.
def Compute(self, inData, outData, propList, progMon):

    # Get the sweeps associated with the probe and validate
    # use the probe name that we had defined earlier
    sweeps = inData.GetSweepNamesForProbe("probe1")
    if( sweeps == None or sweeps.Count > 1):
        AddErrorMessage(self.GetName() + "Unexpected sweep count 0 or > 1 in Compute")
        return False



    # Get the data associated with our probe
    probeData = inData.GetDoubleProbeData("probe1")
    sweepData = inData.GetSweepsDataForProbe("probe1", sweeps[0])

    # Get the user specified properties.
    # Note that ideally, these "X Min" etc names should be written as
    # constant membets and referred to in both the GetInputUDSParams
    # and in Compute to reduce the change of typos.
    useXRangeProp = propList.GetMenuProperty("Activate X Limits").SelectedMenuChoice
    xRangeStart   = propList.GetNumberProperty("X Min").ValueSI
```

```
xRangeEnd     = propList.GetNumberProperty("X Max").ValueSI

# At this stage, one can look at the RequestedQuantities and create
# a dictionary to later check against. However, I am simply computing
# all the quantities.
minVal = 0
maxVal = 0
avgVal = 0

# Check if we need to perform range computation
if useXRangeProp == "Yes":
        seenAny = False
        avgSum = 0
        count = 0

    # zip is used since we also need to pull in sweep data
    # an index and the array notation could also have been used
    for probeVal, sweepVal in zip(probeData, sweepData):
            if sweepVal < xRangeStart or sweepVal > xRangeEnd:
                pass

            # Note that in a better written script, this code would be
            # refactored into it's own function to avoid code
            # duplication
            if not seenAny:
                minVal  = probeVal
                maxVal  = probeVal
                avgSum  = probeVal
                seenAny = True
                count   = 1
            else:
                if probeVal < minVal:
                    minVal = probeVal

                if probeVal > maxVal:
```

```
                    maxVal = probeVal

                avgSum += probeVal
                count  += 1

        if seenAny:
            avgVal = avgSum/count

    else:
        seenAny = False
        avgSum = 0
        for probeVal in probeData:
            if not seenAny:
                minVal = probeVal
                maxVal = probeVal
                avgSum = probeVal
                seenAny = True
            else:
                if probeVal < minVal:
                    minVal = probeVal

                if probeVal > maxVal:
                    maxVal = probeVal

                avgSum += probeVal

        if seenAny:
            avgVal = avgSum/probeData.Count


    # Finally set the output values. Note that these are always set as
    # lists even if we have just one item.
    outData.SetDoubleQuantityData("min_val", [minVal])
    outData.SetDoubleQuantityData("max_val", [maxVal])
    outData.SetDoubleQuantityData("avg_val", [avgVal])
```

```
# And we are done.
return True
```

Related Topics

IUDOPluginExtension Abstract Class

Optional Functions in IDO Extension Abstract Class

The following functions, while a part of the IUDOExtension abstract class, have meaningful default implementations and are therefore optional. However, they can be overridden to take advantage of advanced functionality.

Validate(List<string> errorStringList,

Related Topics

User Defined Outputs: Python Script API

Validate(List<string> errorStringList,

List<UDSProbeParams> udsProbParams,

IPropertyList propList,

List<UDSProbeParams> userSelectionForDynamicProbes)

- **Purpose**: This method is used to validate the user choices. The values of the properties entered, the probes etc. can be checked for suitability.
- **Parameters**:
  - udsProbParams - C# list of UDSProbeParams objects.
  - ***propList*** - IPropertyList object.
  - userSelectionForDynamicProbes - C# list of UDSProbeParams objects.
  - errorStringList - C# list of python strings. Output parameter. Should be set only if validation failed; ignored if validation is successful. One error string should be set per each validation error.
- **Returns**: True on validation success, False on failure.
- **Default implementation**: always returns true.

Example:

```
def Validate(self, errorStringList,probeList,propList, dynamicProbes):
    if probeList == None or probeList.Count == 0:
        errorStringList.Add("Empty probe list")
        return False
    return True
```

Related Topics

Optional Functions in IDO Extension Abstract Class

Data Types Used in Python Script

There are several types that need to be used while authoring the python script. Some of them are used to pass data from UI to python script and to provide interface for working with this data. Some are used to pass data from python script to UI.

To pass data from python script to UI the objects of the C# class must be created in python script using their C# constructors. Then they can be set as functions return values or set to the output parameters using their API.

Their API is described in this section.

Types Used During UDO Computation

There are 2 important types used in the UDO Compute function:

- IUDSInputData class - to get data from UI to python script
- IUDSOutputData class - to set data to UI after doing the necessary computations.

Understanding these two classes and their methods is critical to a correct implementation of the UDO Compute method.

IUDSInputData

GetDoubleProbeData(probeName)

GetSweepsDataForProbe(probeName, sweepName)

GetComplexProbeData(probeName)

GetSweepNamesForProbe(probeName)

GetRequiredQuantities()

GetVariableValues()

GetInterpolationOrdersData(probeName);

IUDSOutputData

SetSweepsData(sweepName, sweepData)

SetDoubleQuantityData(qtyName,qtyData)

SetComplexQuantityData(qtyName, qtyData)

Related Topics

User Defined Outputs: Python Script API

IUDSInputData

The purpose of this class is to get data (probe and sweep) from Desktop.

Examples in this section are just to show proper syntax of the function calls. For actual usage of the class see Compute function example.

GetDoubleProbeData(probeName)

GetSweepsDataForProbe(probeName, sweepName)

GetComplexProbeData(probeName)

GetSweepNamesForProbe(probeName)

GetRequiredQuantities()

GetVariableValues()

GetInterpolationOrdersData(probeName);

Related Topics

Data Types Used in Python Script

GetDoubleProbeData(probeName)

- **Purpose**: This is the primary mechanism by which the UDO script obtains the probe data (as double precision values) for its compute process.
- **Parameters**:
  - probeName: string representing the probe name for which data is requested. This has to be one of the many probes supplied during a call to the UDO's **GetInputUDSParams** method.
- **Returns**: .NET double Array of data for the specified probe if the probe exists or null if the probe is unknown.

Example:

```
# doubleData is a list of floats
doubleData = inData.GetDoubleProbeData("probe1")
```

Related Topics

Data Types Used in Python Script

GetSweepsDataForProbe(probeName, sweepName)

- **Purpose**: All probe data that is supplied is associated with one ore more sweep (an intrinsic quantity like Time, Frequency, Theta, Phi etc that is swept) quantities.
- **Parameters**:
  - probeName - probe name for which which want the sweep data
  - sweepName - sweep name
- **Returns**: .NET double Array of data for the specified probe and sweep.

Example:

```
# sweepData is C# Array of doubles (floats in python)
sweepData = inData.GetSweepsDataForProbe("FarFieldsProbe","Freq"])
```

Related Topics

Data Types Used in Python Script

GetComplexProbeData(probeName)

- **Purpose**: The primary mechanism by which the UDO retrieves data for its input probes (if it expects complex data for the probe).
- **Parameters**:
  - probeName - probe name for which complex data is requested
- **Returns**: .NET double Array (float in python) of data for the specified probe. Each pair of floats represent one complex number: first value is for real part, second value for imaginary part. For instance, array [10.0, 0, 5.1, 2.1] represents 2 complex numbers: (10.0, 0) and (5.1, 2.1).

Example:

```
# complexDataAsDouble is C# Array of doubles (floats in python)
# each pair of floats represents one  complex number
complexDataAsDouble = inData.GetComplexProbeData("FarFieldsProbe")
# creating a list of complex numbers from complexDataAsDouble array
complexData = []
if complexDataAsDouble != None:
    for i in xrange(0,complexDataAsDouble.Count , 2):
        complexData.append(complex(complexDataAsDouble[i],complexDataAsDouble[i+1]))
```

Related Topics

Data Types Used in Python Script

GetSweepNamesForProbe(probeName)

- **Purpose**: To obtain the list of sweep quantity names associated with a given probe. This also indicates the dimensionality of the data. One name implies that the probe-data is 2D (probe-quantity vs Sweep Quantity) and two names implies 3D data (probe-quantity vs Sweep 1 X Sweep 2).
- **Parameters**:
  - probeName - probe name.
- **Returns**: .NET IList<string> - list of sweep names for the current probe name.

Example:

```
# sweepNames  is C# Array of strings
sweepNames = inData.GetSweepNamesForProbe("FarFieldsProbe")
```

Related Topics

Data Types Used in Python Script

GetRequiredQuantities()

- **Purpose**: A given UDO can specify that it provides one of more computed quantities. The user

might choose to create a report from only a few among the various available UDO outputs. This function, returns that list of the UDO output quantities that the user has requested. Only these need be computed in the UDO's compute method.

- **Returns**: .NET IList<string> - list of required quantities names.

Example:

```
# quantities is C# Array of strings
quantities= inData.GetRequiredQuantities()
```

Related Topics

Data Types Used in Python Script

GetVariableValues()

- **Purpose**: This allows the UDO to obtain the names and values of all the design variables for which the UDO quantities are being requested.
- **Returns**: .NET IDictionary<string,string> of key-value pairs for variables. Both key and value are strings.

Example:

```
# theDict is C# Dictionary<string, string>
theDict = inData.GetVariableValues()
if theDict != None:
    #varPair is of .Net KeyValuePair type
    for varPair in theDict:
        varName = varPair.Key       #string
        varValue = varPair.Value    #string
```

Related Topics

Data Types Used in Python Script

GetInterpolationOrdersData(probeName);

- **Purpose**: Returns the interpolation orders that are associated with the probe-data. The probe data is specified at each value of the various sweeps. Any value in between the sweep data points, can use the interpolation data to get a possibly more accurate (compared to linear interpolation) inter-sweep value.
- **Parameters**:
  - probeName (input python string) - probe name.
- **Returns**: NET byte Array of interpolation order for the specified probe. These are to be treated as 8bit signed integers, that is, their values range from 0-127.

Example:

```
# interData is C# Array of bytes (integers in python)
interData = inData.GetInterpolationOrdersData(kProbeNames[0])
```

```
for interValue in theDict:
    order = interValue          # interValue and order are integers
```

Related Topics

Data Types Used in Python Script

IUDSOutputData

This type is a twin of the IUDSInputData in that it is used to store the values computed by the UDO's compute method.

Examples in this section are just to show proper syntaxis function calls. For actual usage of the class see the Compute function example.

SetSweepsData(sweepName, sweepData)

SetDoubleQuantityData(qtyName,qtyData)

SetComplexQuantityData(qtyName, qtyData)

Related Topics

Data Types Used in Python Script

SetSweepsData(sweepName, sweepData)

- **Purpose**: Each quantity that is computed by the UDO can be associated with a sweep. If it is, the values that make up the sweep's data points must be specified using this call.
- **Parameters**:
  - sweepName (string) - sweep name.
  - sweepData (python list of floats) - sweep data for the specified sweep.
- **Returns**: True on success, False on failure.

Example:

```
sweepList = [12.3, 14.5, 16.7]
outData.SetSweepsData("Freq", sweepList)
```

Related Topics

Data Types Used in Python Script

SetDoubleQuantityData(qtyName,qtyData)

- **Purpose**: This method is used to record the computed quantity data for each output that is computed. Please note that unless all the sweeps are reduced, this should be used in conjunction with SetSweepsData
- **Parameters**:
  - qtyName (string) - quantity name.
  - qtyData (python list of floats) - quantity data for the specified quantity.
- **Returns**: True on success, False on failure.

Example:

```
doubleList = [12.3, 14.5, 16.7]
outData.SetDoubleQuantityData("V1PlusV2", doubleList)
```

Related Topics

Data Types Used in Python Script

SetComplexQuantityData(qtyName, qtyData)

- **Purpose**: If the quantity computed is a complex quantity, use this method to set the quantity values. Any sweep values must be set separately via the SetSweepsData method.
- **Parameters**:
  - qtyName (string) - quantity name.
  - qtyData (python list of floats) - quantity data for the specified quantity. Complex numbers are passed as pairs of floats
- **Returns**: True on success, False on failure.

Example:

```
doubleFromComplexList=[]
complexList = [(1+1j), (2+4j), (9.1+3.2j)]
for aComplex in complexList:
    doubleFromComplexList.append(aComplex.imag)
    doubleFromComplexList.append(aComplex.real)
outData. SetComplexQuantityData ("V1PlusV2", doubleFromComplexList)
```

Related Topics

Data Types Used in Python Script

Working With Properties for UDO

A property is the unit for collecting and using input from the user that is used to influence the UDO's Compute. These are initially set up when the UDOs **GetInputUDSParams** method is called and are retrieved in the UDO's Compute method.

There are 3 supported property types that could be used in the UDO script:

- INumberProperty to specify number properties (with unit support).
- IMenuProperty to allow the user to select from a list of options.
- ITextProperty to allow the user to enter text.

The IPropertyList type implements a collection for these properties.

IPropertyList Abstract class

IProperty Abstract class

INumberProperty Abstract class

ITextProperty Abstract class

IMenuProperty Abstract class

Related Topics

User Defined Outputs: Python Script API

IPropertyList Abstract class

Attributes:

- AllProperties (IEnumerable<IProperty> - see *IProperty*)
- NumProperties (int)

Functions:

- GetProperty(string propName): Returns a named property as an IProperty.
- GetMenuProperty (string propName): Returns the named property as an IMenuProperty.
- GetTextProperty (string propName): Returns the named property as an ITextProperty
- GetNumberProperty (string propName): Returns the named property as an INumberProperty
- DeleteProperty (string propName): Deletes an already added named property
- AddNumberProperty(string name, string numberWithUnits): Adds a new number property. If a property with the same name already exists, it is overwritten.
- AddTextProperty(string name, string textValue): Adds a new named text property with the supplied value. Any existing property with the same name is overwritten.
- AddMenuProperty(string name, IList<string> menuChoices): Creates a new named menu property with the supplied list of choices. The default selection is set to item 0 (the first item). Any property with the same name is overwritten.

IProperty Abstract class

Attributes:

- Name (string)
- Description (string)
- PropType (read-only EPropType - see Constants)

Constructor:

- IProperty(string name, EPropType type)

The class is used as base class for INumberProperty, IMenuProperty, and ITextProperty.

INumberProperty Abstract class

Base class:

- abstract class IProperty

Attributes:

- ValueSI (read-only double)

- ValueInUnits (read-only double)
- Units (read-only string)
- HasUnits (read-only bool)

Constructor:

- INumberProperty(string name)

Functions:

- Set(string numberWithUnits)
- SetDouble(double number, string unitString)

ITextProperty Abstract class

Base class:

- abstract class IProperty

Attributes:

- Text (string)

Constructor:

- ITextProperty(string name)

IMenuProperty Abstract class

Base class:

- abstract class IProperty

Attributes:

- MenuSelection (int): *This represents the index into the MenuChoices list.*
- SelectedMenuChoice (string): *This is the item in the MenuChoices list corresponding to the MenuSelection index*
- MenuChoices (IList<string>)

Constructor:

- IMenuProperty (string name)

Example:

```
# adding data to IPropertyList propList; used in Compute function
prop = propList.AddNumberProperty('Offset 1', '0')
prop.Description = 'Trace 1 Offset'
prop = propList.AddNumberProperty("TRATE", "800 MHz")
prop.Description = "Frequency"
prop = propList.AddTextProperty("Text", "The Text")
prop.Description = "Text Property"
prop = propList.AddMenuProperty('Operation', ['Add', 'Subtract', 'Max' ,'Min','Mean'])
```

```
prop.Description = 'Operation menu'

# reading data from IPropertyList propList; used in Validate function
numOfNumberProperties = 0
    if propList != None and propList.AllProperties != None:
        for prop in propList.AllProperties:
            if prop.PropType == Constants.EPropType.PT_NUMBER:
                numOfNumberProperties ++
```

Other Application Specific Classes Used in Python Scripts

This section describes other classes used in Python scripts:

Constants Class

UDSProbeParams Class

UDSDynamicProbes Class

QuantityInfo Class

IProgressMonitor Abstract Class

Related Topics

User Defined Outputs: Python Script API

Constants Class

The constants used in python script are defined in the Constants class.

Attributes:

- kDoubleParamStr : string constant used to specify *double* as the type of a quantity
- kComplexParamStr: string constant used to specify *complex* as the type of a quantity

Enum EPropType: (used to set property type)

```
EPropType.PT_NUMBER
EPropType.PT_TEXT
EPropType.PT_MENU
```

Example:

```
paramType = Constants.kDoubleParamStr
propType = Constants.EPropType.PT_NUMBER
```

Related Topics

Other Application Specific Classes Used in Python Scripts

UDSProbeParams Class

The objects of this class must be created in python script in **GetInputUDSParams** function**.** They are supplied to the Validate function if implemented.

Attributes:

- ProbeName (read-only string)
- ProbeDescription (read-only string)
- ParamType (read-only string)
- ReportTypeName (read-only string)
- ComponentExpression (read-only string)

Constructor: UDSProbeParams(string probeName, string probeDescription, string paramType, string reportTypeName, string componentExpression);

- probeName - required.
- probeDescription - optional (can be empty string).
- paramType - required; can be one of the Constants
  - kDoubleParamStr
  - kComplexParamStr
- reportTypeName - optional (can be empty string)
- ComponentExpression - optional (can be empty string)

Example:

udsProbParam = UDSProbeParams("probe1","", Constants.kDoubleParamStr, "", "",)

Related Topics

Other Application Specific Classes Used in Python Scripts

UDSDynamicProbes Class

Attributes:

- UDSParam (read-only UDSProbeParams)
- UserSelectedProbeName (read-only string)

Constructor: UDSDynamicProbes (UDSProbeParams udsParam, string userSelectedProbeName.

- udsParam - required
- userSelectedProbeName - required

Example:

udsProbParam = UDSProbeParams("probe1","", Constants.kDoubleParamStr, "", "",)

selectedName = "probe1"

udsDynamicProbParam = UDSDynamicProbes(udsProbParam , selectedName)

Related Topics

Other Application Specific Classes Used in Python Scripts

QuantityInfo Class

Attributes:

- ParamType (read-only string)
- FullUnitType (read-only string)

Constructors:

- QuantityInfo(string paramType)
- QuantityInfo(string paramType, string fullUnitType)
- **Parameters**:
  - paramType can be one of the Constants
    kDoubleParamStr
    kComplexParamStr
- fullUnitType is a case insensitive string representing full unit type. It is not defined in Constants. Instead you can use any of the units in string representation - for example, "mm" or"ghz".

Example:

quantityInfo1 = QuantityInfo(Constants.kDoubleParamStr)

quantityInfo2 = QuantityInfo(Constants.kDoubleParamStr,"ghz")

Related Topics

Other Application Specific Classes Used in Python Scripts

IProgressMonitor Abstract Class

The object of this class is a progress monitor. It is used to display calculations progress in UI and check is the user has requested an abort of the computation.

When displayed in the application, each progress message has four items:

- A task name
- A sub-task name
- The progress amount
- A button to abort the task in progress.

All of this functionality and abort interaction is achieved using the following functions.

- SetTaskName (string taskName):
- SetSubTaskName (string subTaskName)
- BeginTask (string name)
- SetTaskProgressPercentage(int progressPercent)
- CheckForAbort(): If the quantities being generated are computationally expensive, the UDO author can periodically call this method and then call EndTask with Fail and return False.
- EndTask (bool passFail)

Example:

```
progMon.BeginTask("Process DQS")
progMon.SetSubTaskName("Compute UI segments")
progMon.SetTaskProgressPercentage(33)
progMon.SetSubTaskName("Compute the rest")
progMon.SetTaskProgressPercentage(100)
progMon.EndTask(True)
```

Related Topics

Other Application Specific Classes Used in Python Scripts

Using .NET Collection Classes and Interfaces in Python Scripts

Some of the API functions specified above use .Net collection classes and interfaces, that is, Array class, IList interface, IEnumerable interface, and IDictionary interface. The following section describes how to work with the .Net collection objects in Python scripts.

.NET Array, IEnumerable, and IList objects can be indexed and iterated over as if they were Python lists. You can also check for membership using 'in'. To get .Net Array and IList sizes you can use python's 'len' or .Net 'Count'.

Example:

Getting size:

```
arraySize = doubleDataArray.Count
arraySize = len(doubleDataArray)

listSize = sweepsNamesList.Count
listSize = len(sweepsNamesList)
```

Iterating:

```
for sweep in sweepsNamesList:
    print sweep

for in in xrange(listSize)
    print sweepsNamesList[i]
```

Checking for membership:

```
if 'Time' in sweepsNamesList:
    doThis()
else:
    doThat()
```

For .NET IDictionary, the same as for Array and IList, you can get size with 'len' or 'Count' and check for membership of the keys using 'in'. Getting values for the keys also works the same way as in python 'dict'.

Example

Getting size:

```
varValuesSize = varValues.Count
varValuesSize = len(varValues)
```

Checking for membership:

```
if 'offset' in varValues:
    print varValues['offset']
```

Getting value:

```
if 'offset' in varValues:
    offsetValue = varValues['offset']
```

As for iteration .NET Dictionary is different from python dict. While iterating, python dict will return keys, .Net Dictionary will return .Net KeyValuePair.

Example:

Iterating:

for .Net IDictionary:

```
for varPair in varValues:   #varPair is of .Net KeyValuePair type
    varName = varPair.Key
    varValue = varPair.Value
```

for python dict:

```
for varName in varValues:
    varValue = varValues[varName]
```

You can use python types instead of .Net types if you prefer. For this you need to cast .Net Array and .Net iList to python list type and .Net Dictionary to python dict type.

Casting should not be used for data arrays - it can be extremely costly for the memory usage as well as time consuming.

Example:

```
aPythonList = list(dotNetArray)
aPythonList = list(dotNetList)
aPythonDict = dict(dotNetDictionary)
```

Related Topics

User Defined Outputs: Python Script API

User Defined Outputs: Messaging Methods

Messaging methods are provided to convey additional information to the user from any of the UDOs methods. The Compute function is the one typically location where such use is anticipated. Any message sent via these functions are displayed in the application's message window using the appropriate icon.

These functions can also be used for debugging purposes.

- **AddErrorMessage(string)**: Call this method to convey an error condition to the user.
- **AddWarningMessage(string)**: Call this method to convey a warning message: typically used for conditions that are not ideal but can be tolerated by the script.
- **AddInfoMessage(string)**: Call this method to convey an informational message to the user. This is the call to use when outputting messages for debugging purposes.

```
########################################################################
#                       Imports
########################################################################
from Ansys.Ansoft.ModulePluginDotNet.Common.API import *
from Ansys.Ansoft.ModulePluginDotNet.Common.API.Interfaces import *
from Ansys.Ansoft.ModulePluginDotNet.UDO.API.Interfaces import *
from Ansys.Ansoft.ModulePluginDotNet.UDO.API.Data import *



class UDOExtension(IUDOPluginExtension):

    def __init__(self):
        pass

    #--- IDA IUDOPluginExtension ------------------------
    def GetUDSName(self):
        return "MinMaxAvg"

    #--- ISA IUDOPluginExtension ------------------------
    def GetUDSDescription(self):
        return "Sample UDO for dimension reducing quantities"



     #--- ISA IUDOPluginExtension ------------------------
      # Returns list of category names
```

```
  def GetCategoryNames(self):
        return ["UDOOutputs"]

 #--- ISA IUDOPluginExtension ------------------------
# returns a list of quantity names for the supplied category name
  def GetQuantityNames(self, catName):
      if catName == "UDOOutputs":
          return ["min_val", "max_val", "avg_val"]
      else:
          return []

 #--- ISA IUDOPluginExtension ------------------------
  # Returns an instance of QuantityInfo for the qtyName supplied or None if such a
#   quantity could not be found
  def GetQuantityInfo(self, qtyName):
      # All the quantities we have are simple doubles
      # we can leave them unitless
      return QuantityInfo(Constants.kDoubleParamStr)

 #--- ISA IUDOPluginExtension ------------------------
  # Returns list of UDSParams and list of dynamic properties
# Adds setup time properties to the propList
  def GetInputUDSParams(self, udsParams, propList, userSelectedDynamicProbes):

      # Add the probes. We need only one double quantity
      param1 = UDSProbeParams("probe1",
          "double quantity probe",
          Constants.kDoubleParamStr,
          "", "")
      udsParams.Add(param1)

      # Add the properties we want the user to supply
      # In this case, we will ask for a start/end range for
      # X parameters. Since we cannot reasonably provide defaults
      # as we have no idea what the sweep limits will be, we will
```

```
        # also ask if the limits are to be activated.
        prop = propList.AddNumberProperty("X Min", "0")
        prop.Description = "Start X value to consider"

        prop = propList.AddNumberProperty("X Max", "1")
        prop.Description = "End X value to consider"

        # For menus, the first option is the default.
        prop = propList.AddMenuProperty("Activate X Limits", ["No", "Yes"])
        prop.Description = "Activate X range"

            return True

    #--- ISA IUDOPluginExtension ------------------------
    # Returns list of UDSParams and list of dynamic properties
    # output UDSDynamicProbeCollection probes
    def GetDynamicProbes(self, probes):
        pass

    #--- ISA IUDOPluginExtension ------------------------
    # Returns list of sweeps names
    # We have no sweeps as we reduce them.
    def GetUDSSweepNames(self):
            return []


    #-------------------------------------------------------------------------------
    # IUserDefinedSolutionHandle API implementation.
    # Calculates output values and sets them using IUDSInputData/IUDSOutputData API.
    def Compute(self, inData, outData, propList, progMon):

        # Get the sweeps associated with the probe and validate
        # use the probe name that we had defined earlier
        sweeps = inData.GetSweepNamesForProbe("probe1")
        if( sweeps == None or sweeps.Count > 1):
```

```
        AddErrorMessage(self.GetName() + "Unexpected sweep count 0 or > 1 in Compute")
        return False


# Get the data associated with our probe
probeData = inData.GetDoubleProbeData("probe1")
sweepData = inData.GetSweepsDataForProbe("probe1", sweeps[0])

            # Get the user specified properties.
            # Note that ideally, these "X Min" etc names should be written as
            # constant members and referred to in both the GetInputUDSParams
            # and in Compute to reduce the change of typos.
            useXRangeProp = propList.GetMenuProperty("Activate X Limits").SelectedMe-
nuChoice
            xRangeStart   = propList.GetNumberProperty("X Min").ValueSI
            xRangeEnd     = propList.GetNumberProperty("X Max").ValueSI

            # At this stage, one can look at the RequestedQuantities and create
            # a dictionary to later check against. However, I am simply computing
            # all the quantities.
            minVal = 0
            maxVal = 0
            avgVal = 0

                # Check if we need to perform range computation
                if useXRangeProp == "Yes":
                    seenAny = False
                    avgSum = 0
                    count = 0

                    # zip is used since we also need to pull in sweep data
                    # an index and the array notation could also have been used
                    for probeVal, sweepVal in zip(probeData, sweepData):
                        if sweepVal < xRangeStart or sweepVal > xRangeEnd:
                            pass
```

```
                # Note that in a better written script, this code would be
                # refactored into its own function to avoid code
                # duplication
                if not seenAny:
                    minVal  = probeVal
                    maxVal  = probeVal
                    avgSum  = probeVal
                    seenAny = True
                    count   = 1
                else:
                    if probeVal < minVal:
                        minVal = probeVal

                    if probeVal > maxVal:
                        maxVal = probeVal

                    avgSum += probeVal
                    count  += 1

            if seenAny:
                avgVal = avgSum/count

        else:
            seenAny = False
            avgSum = 0
            for probeVal in probeData:
                if not seenAny:
                    minVal = probeVal
                    maxVal = probeVal
                    avgSum = probeVal
                    seenAny = True
                else:
                    if probeVal < minVal:
                        minVal = probeVal
```

```
                    if probeVal > maxVal:
                        maxVal = probeVal

                    avgSum += probeVal

                if seenAny:
                    avgVal = avgSum/probeData.Count

            # Finally set the output values. Note that these are always set as
            # lists even if we have just one item.
            outData.SetDoubleQuantityData("min_val", [minVal])
            outData.SetDoubleQuantityData("max_val", [maxVal])
            outData.SetDoubleQuantityData("avg_val", [avgVal])

            # And we are done.
            return True
```

Related Topics

User Defined Outputs: Python Script API

User Defined Outputs: Script Organization

As described in the Introduction section, the UDO scripts should all reside under the **UserDefined-Outputs** folder under either of the three library locations (system, user or personal).

Using Script Libraries

Additional .NET Assemblies

Related Topics

User Defined Outputs: An Introduction

User Defined Outputs: Python Script API

Using Script Libraries

If you decide that you need base classes, additional data files, and etc., to organize your UDOs better, you can do so. This type of library organization allows code reuse between similar UDOs and can be very helpful. There is special support provided for this type of script-library organization:

- **All script-library and other support files need to be in a *Lib* sub-directory under the UserDefinedOutputs directory**. Any *.py* files found in such **Lib** directories are ignored and not displayed in the GUI as a valid UDO choice.
- For a UDO script at any given directory depth, all Lib directories in its parent directories will be automatically added to the system include path (and so, any support script files from any Lib directory till the top level UserDefinedOutputs directory can be imported)

Related Topics

User Defined Outputs: Script Organization

Additional .NET Assemblies

Using additional .NET assemblies

Because the UDO functionality uses IronPython, we have access to the full .NET eco system. If needed, any subset of the UDO functionality can be implemented in any .NET language and used by the UDO script. There are simple rules to follow to achieve this.

1. Build your .NET assembly for .NET 2.0 runtime.
2. Drop the built assembly in any **Lib** directory upstream of the UDO script location: that is, if you have your UDO script in *C:\Users\x\PersonalLib\UserDefinedOutputs\a\b\c\myudo.py* and have a .NET assembly called *com.Acme.UDOLib* You can keep the .NET assembly under
   - UserDefinedOutputs\Lib,
   - UserDefinedOutputs \a\Lib,
   - UserDefinedOutputs \a\b\Lib
   - UserDefinedOutputs\a\b\c\Lib
3. Add the following line to your python script
   - Import clr
   - clr.AddReference("com.Acme.UDOLib")

- import com.Acme.UDOLib -or-- from com.Acme.UDOLib import * *etc*

If for some reason you cannot place the .NET assemblies into a Lib directory under UserDefined-Outputs, you need to do a couple more steps before step 3 listed above.

```
Import sys
sys.path.append("full path to your .NET assembly location")
```

Related Topics

User Defined Outputs: Script Organization

Using Script Libraries

Toolkit

The Toolkit command in the HFSS menu provides access to a design type specific IronPython script, such as the Hearing Aid Compliance Test, MIMO calculation or a module specific task, such as for Boundary, Excitation, or Solve Setup. The Hearing Aid Compliance Test toolkit example involves multiple modules. The MIMO example includes a Toolkit script and a User Defined Solution Script. The Cable modeling toolkits serve automotive cable applications and oil and gas applications.

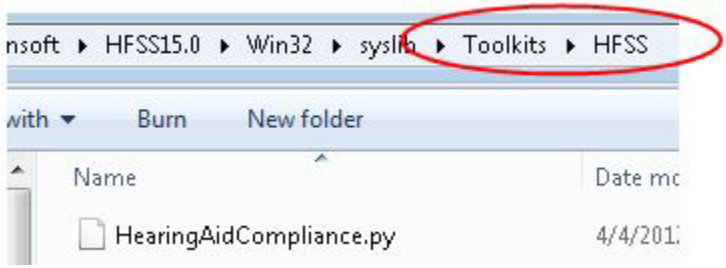
The python script provides default but customizable UI and canned automations. The UDD format can be adjusted/extended as needed. The default UI is a modal dialog.

This allows us to support changing standards, or multiple standards, without necessarily being tied to the product release cycle. Similarly, you can create new UDD to customize the test report.

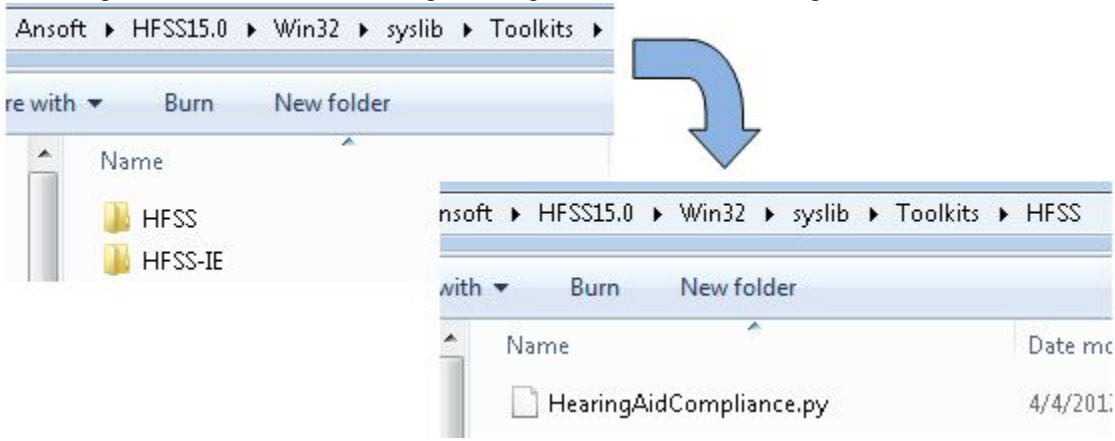
UDD are saved in project result directory. This type of reports can be created prior to simulation, but will only be populated when there is solution.

Undo/Redo/Scripting is supported. If a script includes multiple commands, it takes multiple undo(s) to revert all the changes made by the python script. Also, the script playback might not work if the baseline of the design has changed.

A directory called "Toolkits" appears in the syslib, userlib and personallib of HFSS installation. Menu items for files found in <installation>/syslib will be inserted at the first level of the menu.



There are product specific directories in "Toolkits" and each directory contains toolkits (scripts) that are specific for that product. The following snapshot illustrates the file structure on disk that provides access to the "HearingAidCompliance" toolkit at the design level menu.



Related Topics

Hearing Aid Compliance Test

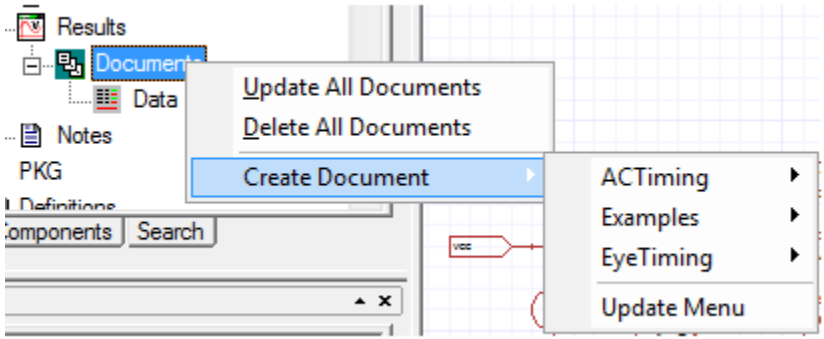
User Defined Solution for MIMO Calculations

Cable Modeling - Automotive Toolkit

Cable Modeling - Oil and Gas Toolkit

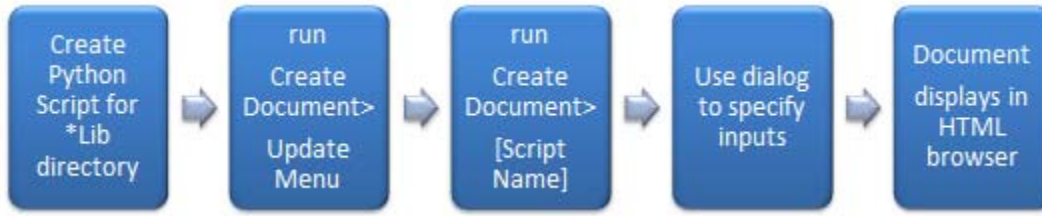
User Defined Documents (UDDs)

User defined documents (UDDs) are custom reports that you define through IronPython scripts. Once placed in a Lib directory, you can access the scripts via the **Create Document** command. The scripts describe a **Create User Defined Document** dialog that lets you specify trace and solution inputs. After you confirm your input selections, an xml, html and pdf document is generated. A web browser window opens to display the generated html file. The created document appears in the Project tree, under Results in the Documents folder.



Results>Create Document submenu lists UDD script names when such scripts are placed in the appropriate *lib directories

The general UDD process flow is as follows.



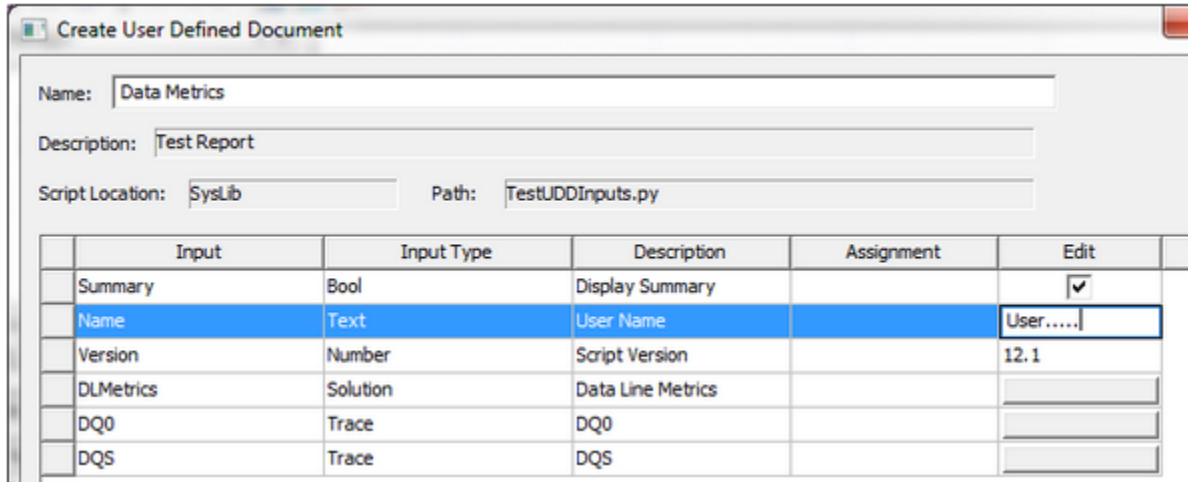
The UDD python scripts must be placed in the **UserDefinedDocuments** directory under either of **syslib**, **userlib** or **Personallib** with any subdirectory structure needed. The Lib directory can contain python scripts that have common code that other scripts can use.

Use **Results>Create Document>Update Menu** to refresh the menu to include the new UDD scripts that have been copied to syslib, userlib or Personallib, or to exclude them if they have been deleted, after the launch of desktop.

The UDD scripts that are in syslib/UserDefinedDocuments, userlib/UserDefinedDocuments or Personallib/UserDefinedDocuments become available through the **Results >Create Document** menu.

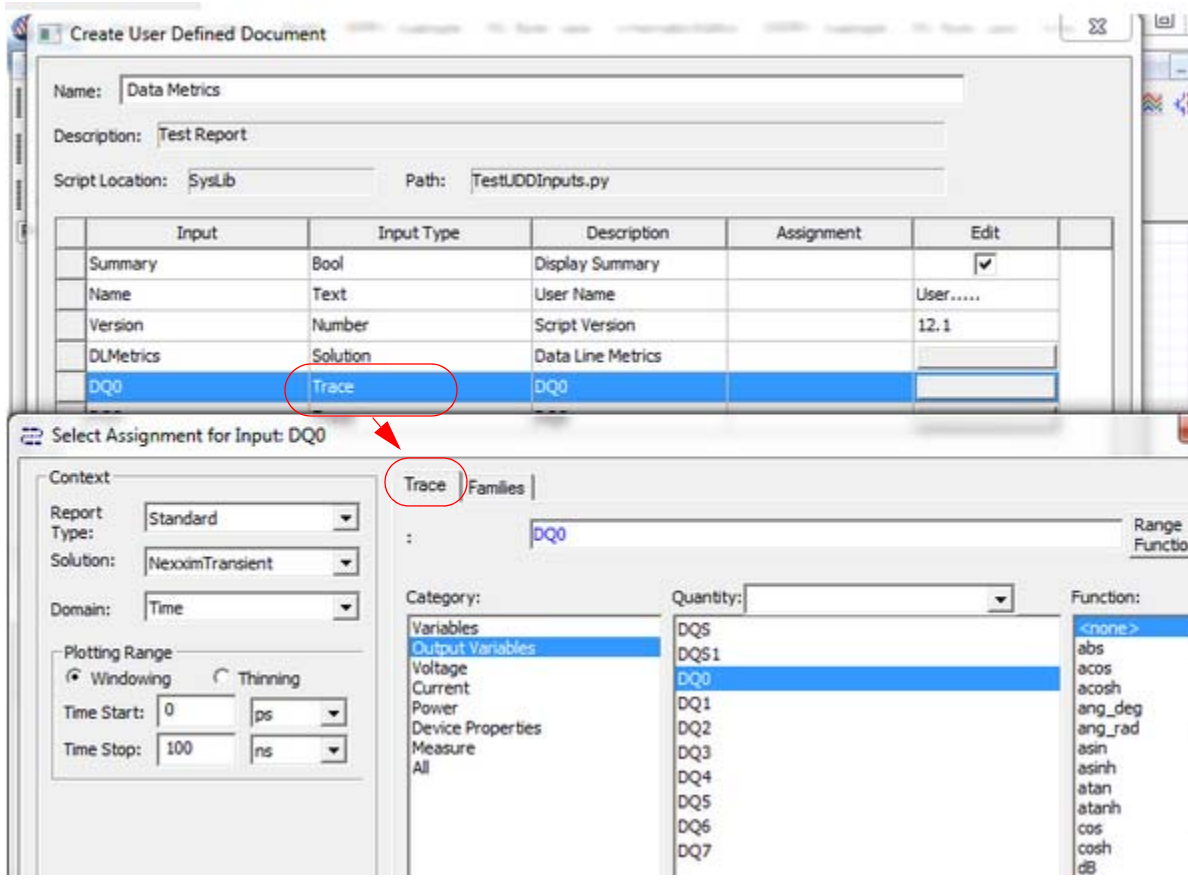
Create User Defined Document Dialog Inputs

User defined documents allow data from traces, solutions and report types as inputs. A UDD can specify the named inputs for which you select or enter the values in the **Create User Defined Document** dialog that displays when you run **Results>Create Document>**<*scriptName*>.



Input Types can be of Boolean, number, text, trace or solution type. The boolean, number and text type can be given a default value that you can interactively override when the document is created or modified. For example, you can select a trace when you create or modify a UDD document. The trace data is available to the user and can be accessed from the python script.

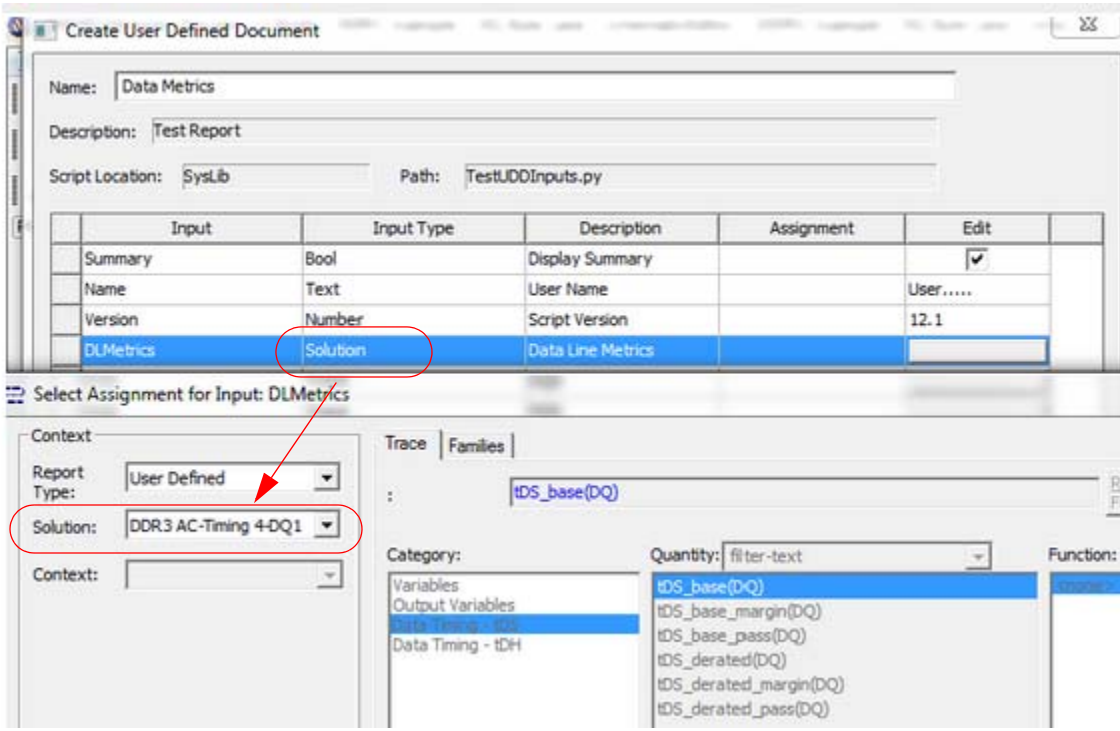
At the time of selection you can choose from the **Reporter** dialog, the report type (Standard, Eye Diagram, User Defined), solution name, context and the quantity for which you want the trace data.



Input Type can also be Solution. You can select an entire solution when the document is created or modified. The solution data in its entirety, is now available to the user and can be accessed from the python script.

At the time of selection you can choose from the reporter dialog, the report type (Standard, Eye Diagram, User Defined), solution name and context. A specific quantity cannot be selected since data for all quantities in the solution are available.

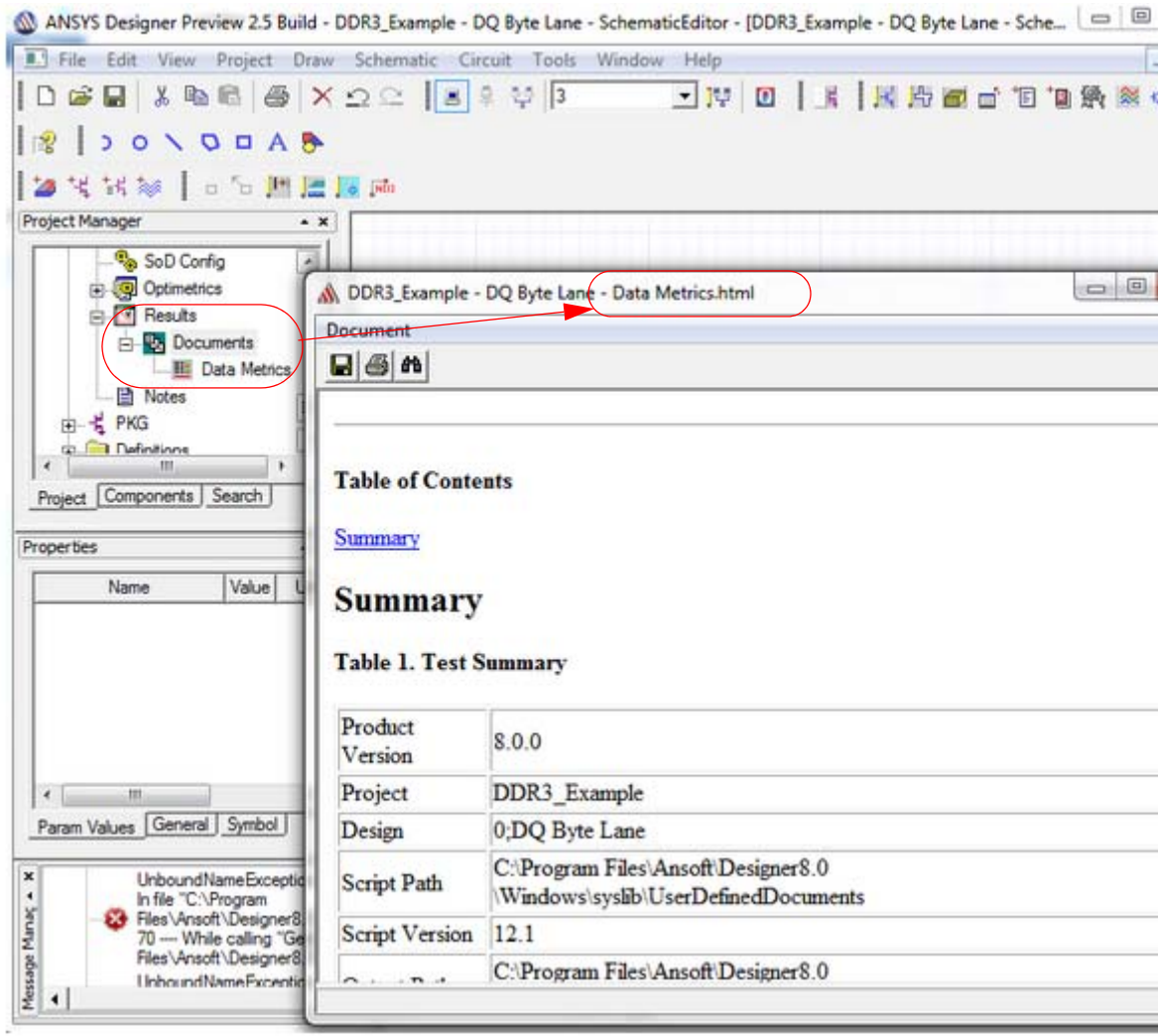
(Note: The category/Quantity/Function portion of the dialog is disabled for user input)



UDD Document Creation and Display

After all the input selections for a UDD are confirmed, based on the script, an xml, html and pdf document is generated based on the inputs provided by the user. (The xml, html and pdf generation is based on specific calls in the python script, which are explain in a following section). A web browser window also opens to display the generated html file.

The created document will be placed under a new folder named "Documents" under the "Results" folder. All documents that are created by the user for the design will be placed under this folder.



Related Topics

Managing Documents Listed in the Project Window Under Results

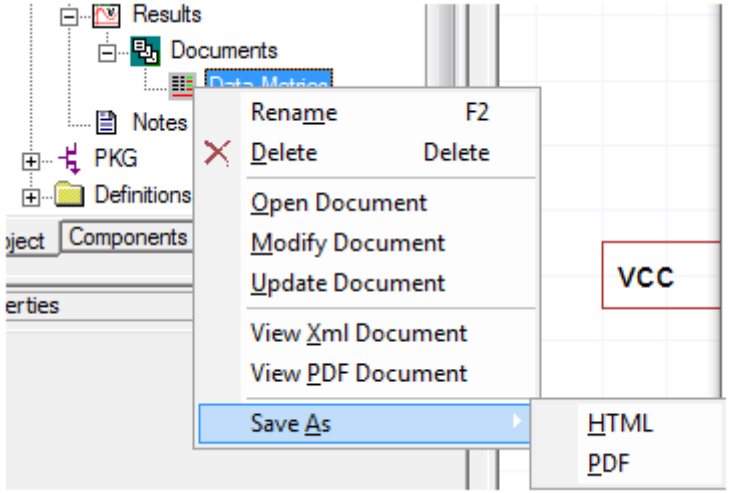
Viewing UDDs with an Html Web Browser

UDD Script Libraries

User Defined Documents: Python Script API

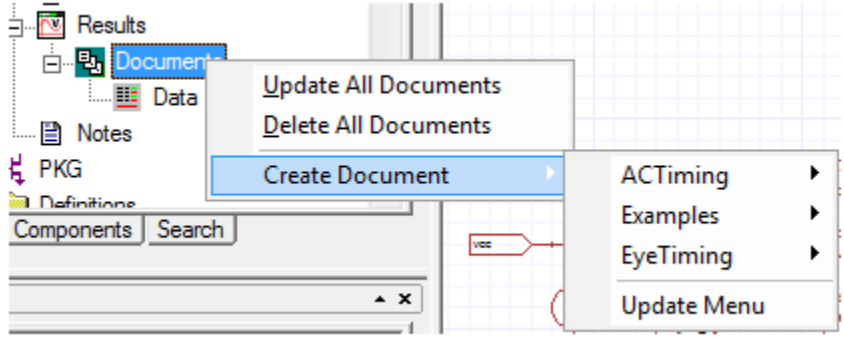
Managing Documents Listed in the Project Window Under Results

Right click on a user defined document displayed in the Project Manager tree to bring up a menu where you can rename, delete the document. **Open document** opens the web browser with the html document. **Modify document** opens the setup dialog where you can change the selections for the input. To view the xml and the pdf document simply choose the appropriate menu items. There is also a menu item to save the document in a different location.



Documents folder right click menu

Right click on the documents folder has the menu options to Update All Documents or Delete All Documents. It also provides the option of creating a document from here.

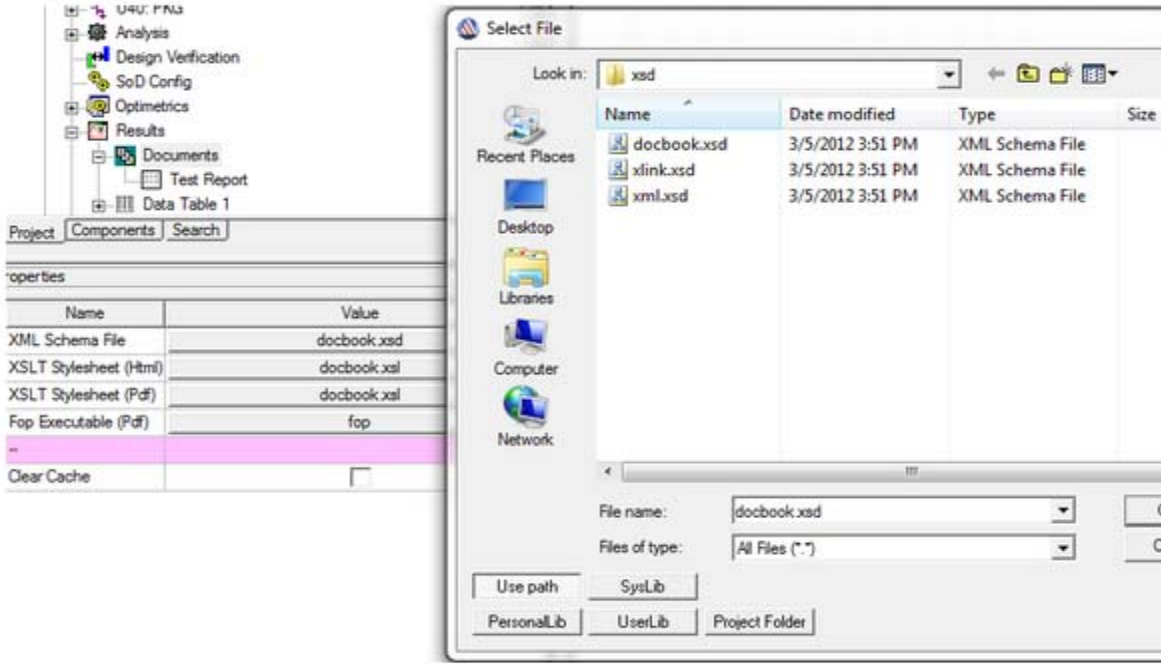


Document folder Property window

When the documents folder is selected, the Property window shows the following properties

- XML Schema File - File path to the XML schema file.

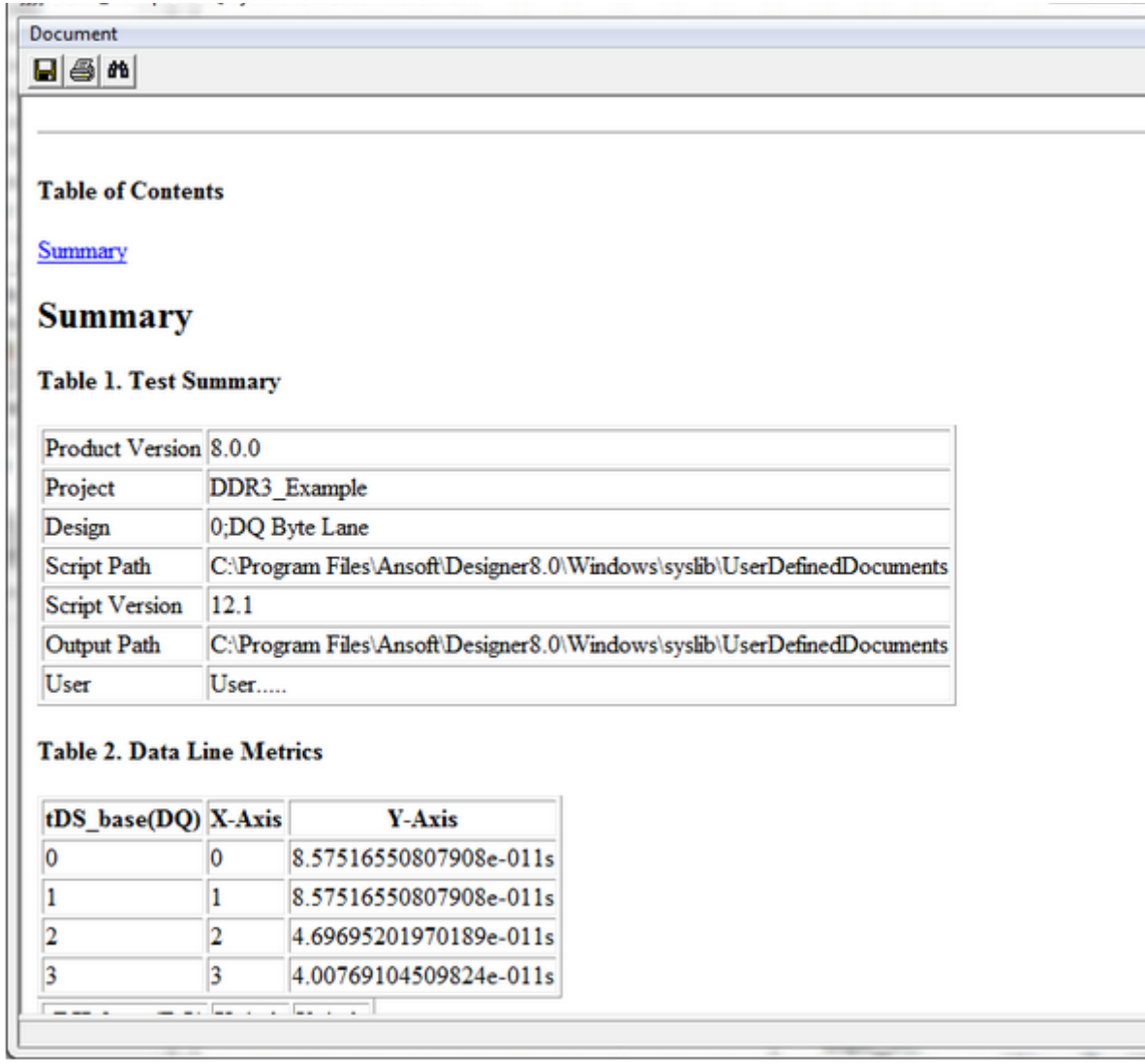
- XSLT StyleSheet (Html) - File path to the XSLT stylesheet file used for Html generation.
- XSLT StyleSheet (Fo) - File path to the XSLT stylesheet file used for Pdf generation.
- Fop Executable (Pdf) - File path the Fop executable used for Pdf generation.
- Clear Cache - Clears the cached XSL transform object and forces creation of a new one. (The caching is done to save time during document generation, so subsequent generation or update of the document can use the cached transform object. But sometimes you may want to force a recompile of the document if you change the stylesheet).



The XML, HTML and PDF generation require the XML schema file and XSLT stylesheets to generate proper output. In addition, the PDF generation requires a FOP executable. You can use the defaults provided in the installation or provide the file paths of your own preferred stylesheets and fop executable installed in his machine.

Viewing UDDs with an Html Web Browser

The XML and HTML documents can be viewed in a web browser with some basic functionality like printing the document, searching the document for a phrase or sentence and saving the document.



Related Topics

UDD Script Libraries

Base classes and data files shared between similar UDDs can be organized to reuse the code in a better way. All script-library and other support files need to be in a Lib sub-directory under the UserDefinedDefinitions directory. Any .py files found in such Lib directories are ignored and not displayed in the GUI as a valid UDD choice. For a UDD script at any given directory depth, all Lib directories in its parent directories will be automatically added to the system include path (and so, any support script files from any Lib directory till the top level UserDefinedDefintions directory can be imported)

The UDD functionality uses IronPython so we have access to all the .NET assembles. If needed, any subset of the UDD functionality can be implemented in any .NET language and used by the UDD script. There are simple rules to follow to achieve this.

1. Build your .NET assembly for .NET 2.0 runtime.
2. Drop the built assembly in any Lib directory upstream of the UDD script location: that is, if you have your UDD script in C:\Users\x\PersonalLib\UserDefinedDefintions\a\b\c\myudd.py and have a .NET assembly called com.Acme.UDDLib You can keep the .NET assembly under
   - UserDefinedDefintions\Lib,
   - UserDefinedDefintions\a\Lib,
   - UserDefinedDefintions\a\b\Lib
   - UserDefinedDefintions\a\b\c\Lib
3. Add the following line to your python script
   - Import clr
   - clr.AddReference("com.Acme.UDDLib")
   - import com.Acme.UDDLib -or-- from com.Acme.UDDLib import * etc

   If for some reason you cannot place the .NET assemblies into a Lib directory under UserDefinedDefintions, you need to do a couple more steps before step 3 listed above.

   ```
   Import sys
   sys.path.append("full path to your .NET assembly location")
   ```

Related Topics

User Defined Documents: Python Script API

User Defined Documents: Python Script API

A User Defined Documents (UDD) extension is implemented as an IronPython script that defines a class with a specific name: **UDDExtension** which derives from a specific base class **IUDDPluginExtension** and implements its abstract methods.

Import Statements

The base class to be used and the types it uses in turn are contained in .NET assemblies. The use of these requires that the assemblies be imported into the UDD script: the following import statements should be added to the top of the python script:

```
from
Ansys.Ansoft.DocGeneratorPluginDotNet.DocGenerator.API.Data
import *
from
Ansys.Ansoft.DocGeneratorPluginDotNet.DocGenerator.API.Interfac
es import *
```

UDDExtension Class

The UDD itself should be implemented as an IronPython class called **UDDExtension** which must derive from the **IUDDPluginExtension** abstract base class (from the **Ansys.Ansoft.DocGeneratorPluginDotNet.DocGenerator.API.Interfaces** namespace).

Note that power users could derive a class hierarchy tuned toward a specific type of UDDs and that they can derive from their own base classes. The only requirement is that directly or indirectly, the UDD class must derive from **IUDDPluginExtension**.

Example:

```
def BaseClassUDD ((IUDDPluginExtension):
#base class implementation
…
def UDDExtension ((BaseClassUDD):
#UDD class implementation
…
```

Note All of the above text has been copied from the help section for the UDOs and modified for the UDDs. Since the UDDs are modelled after the UDOs, the usage is also similar.

IUDDPluginExtension Abstract Class

Required functions:

The **IUDDPluginExtension** abstract class declares the following abstract methods that must be implemented in the UDDExtension class or one of its base classes. Not implementing any of these methods will result in a run-time error and a non functioning UDD.

GetUDDName() : Return a string that is used as a prefix for all solution instances created using this UDD.

Example:

```
def GetUDDName(self):
return "MinMaxAvg
```

GetUDDDescription() : Returns a description for the UDD, its purpose etc.

Example:

```
def GetUDDDescription(self):
return "Sample UDD"
```

ShowDefaultSetupDialog() : Returns True if the default dialog is to be shown. Return False if the user does not want the default dialog. In this case the user might want to implement/show a customized setup dialog.

Example:

```
def ShowDefaultSetupDialog(self):
return True
```

GetUDDInputParams(List<UDDInputParams> uddInputs) : Returns the list of inputs parameters for the User Defined Document. Returns boolean: True on success, False on failure.

The supplied input parameters are used to populate details of the parameters to which the UDD user will specify value, specify the input names and their types.

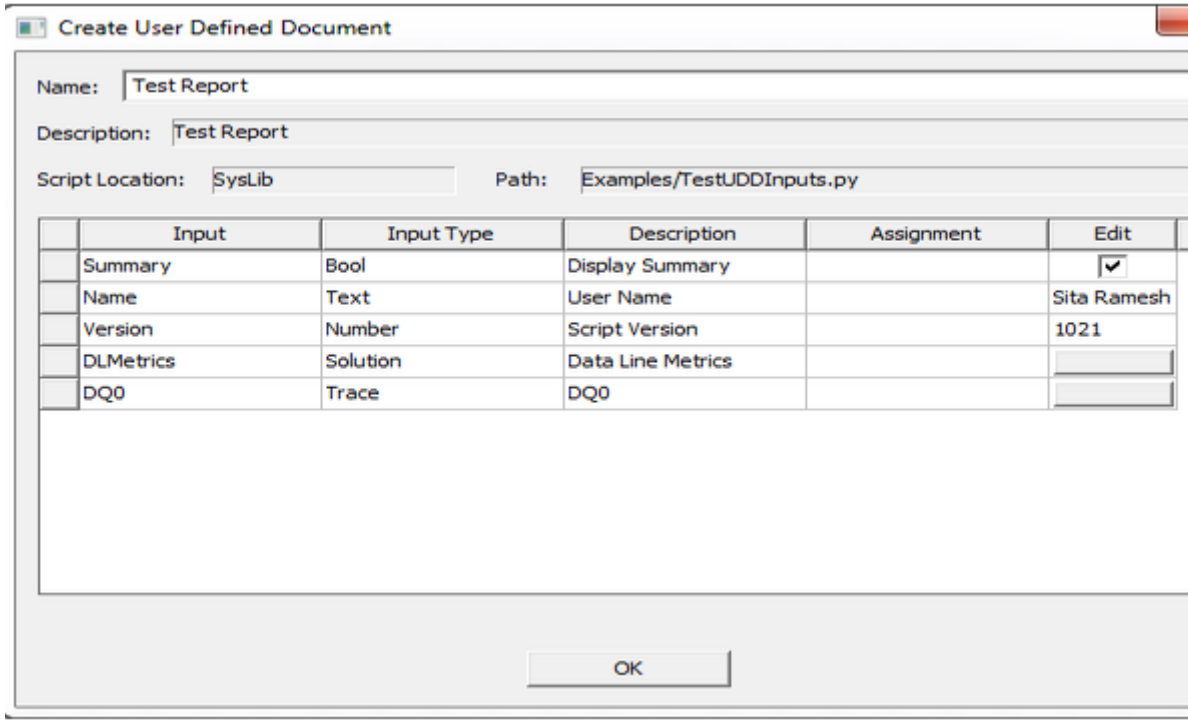
uddInputs: .NET list of UDDInputParams objects. The UDD script is expected to add one instance of UDDInputParams for each input definition it wants displayed. The UDD user will, when creating the UDD, assign a matching value to each such input.

Example:

```
def GetUDDInputParams(self, uddInputs)
# Boolean input
param1 = UDDInputParams("Summary","Display Summary",
Constants.kBoolTypeStr, True)
uddInputs.Add(param1)
# Text input
param2 = UDDInputParams("Name","User Name",
Constants.kTextTypeStr , "Sita Ramesh")
```

```
uddInputs.Add(param2)
# Number input
param3 = UDDInputParams("Version","Script Version",
Constants.kNumberTypeStr, 1021)
uddInputs.Add(param3)
# Solution input
param5 = UDDInputParams("DLMetrics","Data Line
Metrics",Constants.kSolutionTypeStr)
uddInputs.Add(param4)
# Trace input
param5 = UDDInputParams("DQ0","DQ0",Constants.kTraceTypeStr)
uddInputs.Add(param5)

return True
```

Based on the input params the following dialog is displayed when you click **Reports>Create Document**. The name and description of the UDD are also displayed in this dialog.



Generate(List<UDDInputData> uddInputs, IUDDGenerator generator, IProgressMonitor progressMonitor) : This is the main method which accesses the data from the uddInputs and generates the document.

uddInputs: The list of inputs that the user setup in the dialog. They are now available to query for data.

generator: This is the document generator object which we use to create different elements of the document like titles, sections, tables, images and write the data too. This interface is explained in the Document Generator Interface document.

progressMonitor : IProgressMonitor object. This can be used to set progress for long running calculations, check for user initiated abort etc.

Example:

```
def Generate(self, input, docgen, progMon):

# Gather data from inputs
boolinput = input[0].Data()
textinput = input[1].Data()
dblinput = input[2].Data()

# Get document root
docroot = docgen.GetDocumentRoot()

# Add Section
section1 = docroot.AddSection("Summary", "Overall Results ")

# Add a table
table1 = section1.AddTable("Test Summary")

#Add a table group with 2 columns
tgroup1 = table1.AddTableGroup(2)

# get desktop application
oApp = self.GetUDDAppContext()
if oApp != None:
   oDesktop = oApp.GetAppDesktop()
   if  oDesktop != None:
   # version number
```

```
version = oDesktop.GetVersion()
text1 = tgroup1.AddContent()
text1 .Add(0, "Product Version")
text1 .Add(1, version)

oProject = oDesktop.GetActiveProject()
if  oProject != None:
      projectname= oProject.GetName()
      text1 = tgroup1.AddContent()
      text1 .Add(0, "Project")
      text1 1.Add(1, projectname)

   oDesign = self.GetUDDDesignContext()
   if oDesign != None:
      designname = oDesign.GetName()
      text1 = tgroup1.AddContent()
      text1 .Add(0, "Design")
      text1 .Add(1, designname)

   # Provides a script path
   scriptpath = docgen.GetScriptPath()
   text1 = tgroup1.AddContent()
   text1 .Add(0, "Script Path")
   text1 .Add(1, scriptpath )

   #Provides the script version
   text1 = tgroup1.AddContent()
   text1 .Add(0, "Script Version")
      text1 .Add(1, str(dblinput ))

   #Provides the output xml path
   outputpath = docgen.GetOutputFilePath()
   text1 = tgroup1.AddContent()
   text1 .Add(0, "Output Path")
   text1 .Add(1, outputpath )
```

```
        #Provides the user information
        text1 = tgroup1.AddContent()
        text1 .Add(0, "User")
        text1 .Add(1, textinput)

# Generate Xml output
docgen.Write(False)

# Generate Html output
docgen.WriteHTML()

# Generate PDF output
docgen.WritePDF()

return True
```

Optional functions:

SetupUDDInputParams(List<UDDInputParams> uddInputs) : Displays a customized dialog and returns the user choices for the input params.

uddInputs- .NET list of UDDInputParams objects with values for each of them. These can be the user choice for each input obtained through a custom dialog or some other non graphical assignment.

We cannot process trace and solution types of input with a custom dialog because there is no way of assigning solution data to the input without the invocation of the reporter dialog.

Example:

```
def SetupUDDInputParams(self, uddInputs)
udddialog = BaseExampleUDDDialog()
if udddialog.ShowDialog() == Forms.DialogResult.OK:
   # Boolean input
   param1 = udddialog.GetInput("Summary")
   uddInputs.Add(param1)

   # Text input
```

```
param2 = udddialog.GetInput("Name")
uddInputs.Add(param2)

# Number input
param3 = udddialog.GetInput("Version")
uddInputs.Add(param3)
```

HandleUDDEvents(List<string> eventTags) : The tags associated with the event is received by plugin using this abstract class.

This method is the event handler for all link events set by the SetEventLink() method on a IUDDText. Refer to the definition of the IUDDText object in the Document Generator Interface document.

Example:

```
def HandleUDDEvents(self, uddLinks):
if uddLinks[0] == "Open" Report":
   # Get Design Name
   oDesign = self.GetUDDDesignContext()
    if oDesign != None:
      oDesign.OpenReport(uddLinks[1])
return True
```

GetUDDSchema() : Returns the file path of the schema to validate the xml. This will override the default schema used. Return string containing the full file path of the schema.

```
def GetUDDSchema(self):
return "C:\\Program
Files\\Ansoft\\Designer8.0\\Windows\\common\\docbook\\schema\\x
sd\\docbook.xsd"
```

GetUDDStyleSheetForHtml() : Returns the file path of the style sheet used to generate the html document. This will override the default stylesheet for html. Returns string containing the full file path of the style sheet.

```
def GetUDDStyleSheetForHtml(self):
return "C:\\Program
Files\\Ansoft\\Designer8.0\\Windows\\common\\docbook\\"
```

GetUDDStyleSheetForPdf() : Returns the file path of the style sheet used to generate the pdf document. This will override the default stylesheet for pdf. Returns string containing the full file path of the style sheet.

Example:

```
def GetUDDStyleSheetForPdf(self):
return "C:\\Program
Files\\Ansoft\\Designer8.0\\Windows\\common\\docbook\\xsl\\fo\\
docbook.xsl"
```

GetFopExecutable() : Returns the file path of the fop executable used to generate the pdf document. This will override the default stylesheet for pdf.Returns string containing the full file path of the fop executable.

Example:

```
def GetFopExecutable(self):
return "C:\\Program
Files\\Ansoft\\Designer8.0\\Windows\\common\\ApacheFOP\\fop-
1.0\\fop"
```

GetUDDAppContext() : Returns the UDD Owner (if set). This is a Dispatch wrapper that is essentially a COM IDispatch implementation and corresponds to the IDispatch pointing to the desktop app.

GetUDDDesignContext() : Returns the UDD Owner (if set). This is a Dispatch wrapper that is essentially a COM IDispatch implementation and corresponds to the IDispatch pointing to the Design.

Data Types Used in Python Script

There are several types that you must use while authoring the python script. Some of them are used to pass data from UI to python script and to provide interface for working with this data. Some are used to pass data from python script to UI.

To pass data from python script to UI the objects of the C# class must be created in python script using their C# constructors. Then they can be set as functions return values or set to the output parameters using their API.

Constants class

kTraceTypeStr : string constant used to specify an input of trace type

kSolutionTypeStr : string constant used to specify an input of solution type

kNumberTypeStr : string constant used to specify an input of number type

kTextTypeStr : string constant used to specify an input of text type

kBoolTypeStr : string constant used to specify an input of boolean type

kStandardReportStr : string constant to specify a standard report

kEyeDiagramReportStr : string constant to specify an eye diagram report

kUserDefinedReportStr : string constant to specify a user defined report

kSweepDomainStr : string constant to specify the sweep domain

kTimeDomainStr : string constant to specify the time domain

UDDInputParams class

The objects of this class must be created in python script in the **GetUDDInputParams()** function and the **SetUDDInputParams()** function.

Attributes :

Input Name (string)

Input Description (string)

Input Type (Can be Boolean, Number, Text, Trace or Solution) (string)

BoolData (boolean)

DoubleData (double)

TextData (string)

ReportType (string)

SolutionName (string)

DomainName (string)

Constructors:

UDDInputParams(string name, string description, string type)

UDDInputParams(string name, string description, string type, bool data)

UDDInputParams(string name, string description, string type, double data)

UDDInputParams(string name, string description, string type, string data)

UDDInputParams(string name, string description, string type, string reportType, string solutionName, string domainName)

Property Accessors :

Name : Get/Set the name of an input

Description : Get/Set the desccription of an input

Type : Get/Set the type of an input

BoolData : Get/Set the data of a boolean input

DoubleData : Get/Set the data of a number input

TextData : Get/Set the data of a text input

ReportType : Get/Set the report type

SolutionName : Get/Set the name of the solution

DomainName : Get/Set the name of the domain

IProgressMonitor Abstract Class

The object of this class is a progress monitor. It is used to display calculations progress in UI and check is the user has requested an abort of the computation.

When displayed in the application, each progress message has four items:

A task name

A sub-task name

The progress amount

A button to abort the task in progress.

All of this functionality and abort interaction is achieved using the following functions.

SetTaskName (string taskName):

SetSubTaskName (string subTaskName)

BeginTask (string name)

SetTaskProgressPercentage(int progressPercent)

CheckForAbort(): If the quantities being generated are computationally expensive, the UDO author can periodically call this method and then call EndTask with Fail and return False.

EndTask (bool passFail)

Example:

```
progMon.BeginTask("Process DQS")
progMon.SetSubTaskName("Compute UI segments")
progMon.SetTaskProgressPercentage(33)
progMon.SetSubTaskName("Compute the rest")
progMon.SetTaskProgressPercentage(100)
progMon.EndTask(True)
```

UDD Input interfaces

The Generate function takes in a list of inputs. These input interfaces allow the user to access data from the design.

IUDDInputBool : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

Data() : Gets the boolean data, set by the user in the setup dialog.

IUDDInputDouble : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

Data() : Gets the double data, set by the user in the setup dialog.

IUDDInputText : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

Data() : Gets the text data, set by the user in the setup dialog.

IUDDInputTrace : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

DoubleData() : Method used to return x and y double data as a IDictionary<double, double>

DoubleData(IDictionary<string, string> variation) : Method used to return x and y double data as a IDictionary<double, double>, given a variation.

ComplexData() : Method used to return x data and y complex data as a IDictionary<double, double[]>

ComplexData(IDictionary<string, string> variation) : Method used to return x data and y complex data as a IDictionary<double, double[]>, given a variation.

TextData() : Method used to return x data and y data as a IDictionary<string, string>

TextData(IDictionary<string, string> variation) : Method used to return x data and y data as a IDictionary<string, string>, given a variation.

VariableValues() : Method used to get a list of variations as a IList<Dictionary<string, string>>

IUDDInputSolution : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

DoubleData(string name) : Method used to return x and y double data as a IDictionary<double, double>, given a quantity name.

DoubleData(string name, IDictionary<string, string> variation) : Method used to return x and y double data as a IDictionary<double, double>, given a quantity name and a variation.

ComplexData(string name) : Method used to return x data and y complex data as a IDictionary<double, double[]>, given a quantity name.

ComplexData(string name, IDictionary<string, string> variation) : Method used to return x

data and y complex data as a IDictionary<double, double[]>, given a quantity name and a variation.

TextData(string name) : Method used to return x data and y data as a IDictionary<string, string> given a quantity name.

TextData(string name, IDictionary<string, string> variation) : Method used to return x data and y data as a IDictionary<string, string>, given a quantity name and a variation.

CategoryNames() : Method to return a list of category names in the solution as an IList<string>

QuantityNames(string category) : Method to return a list of quantity names in the solution as an IList<string>, given a category.

VariableValues() : Method used to get a list of variations as a IList<Dictionary<string, string>>

Examples:

```
def Generate(self, input, docgen, progMon):
# Getting the boolean data set by the user
boolinput = input[0].Data()
# Getting the double data set by the user
dblinput = input[1].Data()
# Getting the text data set by the user
textinput = input[2].Data()
# Getting the category names in a solution
categories = input[3].CategoryNames()
# Getting the quantity names based on a category
quantities = input[3].QuantityNames(categories[0])
# Getting the XY data from the trace
xydata = input[4].DoubleData()
```

User Defined Document Scripting Interface

To access the UserDefineddocuments scripting object, use:

```
Set oModule = oDesign.GetModule("UserDefinedDocuments")
```

Once you have the scripting object, you can use the following methods:-

1. `AddDocument([in] VARIANT data, [in] VARIANT traces, [out, retval] BSTR* uniqueName)`
   a. Takes a VARIANT data which defines the document.
   b. Takes a VARIANT trace data for the inputs in the document.
   c. Returns a unique name
2. `EditDocument( [in] BSTR originalName, [in] VARIANT modifiedData, [in] VARIANT modifiedTraces, [out, retval] BSTR* uniqueName)`
   a. Takes the name of the original document.
   b. Takes a VARIANT data which defines the edited document.
   c. Takes a VARIANT trace data for the inputs in the document.
   d. Returns a unique name
3. `RenameDocument( [in] BSTR oldName, [in] BSTR newName)`
   a. Takes the name of the original document.
   b. Takes the new name of the document.
4. `DeleteDocument( [in] BSTR name)`
   a. Takes the name of the document to be deleted.
5. `UpdateDocument( [in] BSTR name)`
   a. Takes the name of the document to be updated.
6. `ViewHtmlDocument( [in] BSTR name)`
   a. Takes the name of the document to be viewed in HTML.
7. `ViewPdfDocument( [in] BSTR name)`
   b. Takes the name of the document to be viewed as a PDF.
8. `SaveHtmlDocumentAs( [in] BSTR name, [in] BSTR saveTo)`
   a. Takes the name of the document to be saved.
   b. Takes the file path to save the socument as.
9. `SavePdfDocumentAs( [in] BSTR name, [in] BSTR saveTo)`
   a. Takes the name of the document to be saved.
   b. Takes the file path to save the socument as.
10. `GetDocumentDefinitionNames( [in] BSTR separator, [out, retval] VARIANT* names)`

a. 'separator' is used to convey the directory "level"

b. Returns the (file) names of doc definitions according to the files in various installation directories.

11.DeleteAllDocuments()

12.UpdateAllDocuments()

For 6, 7, 8, and 9, the document must have an existing, generated HTML or PDF.

The UserDefinedDocument Data format in the script:

To define a document in VB script:

```
Array("NAME:Test Report",  (Name of the document)
"Test Report",             (Description of the document)
"SysLib",                  (Location of the python
script(Syslib, Userlib, PeronalLib etc)
"TestUDDReport",           (Relative path of the script in the
UserDefinedDocuments folder)

// Start of input definition //
Array("NAME:Inputs",       (Document Inputs keyword)
// Solution input //
Array("NAME:DLMetrics",    (Input name)
"Solution",                (Solution Input Type)
"Data Line Metrics",       (Input Description)
-1,                        (Solution ID)
-1),                       (Report ID)
// Trace input //
Array("NAME:DQ0",          (Input name)
"Trace",                   (Trace Input Type)
"DQ0",                     (Input Description)
-1,                        (Solution ID)
-1),                       (Report ID)
// Text input //
Array("NAME:Name",         (Input name)
"Text",                    (Text Input Type)
"User Name",               (Input Description)
Array("Sita Ramesh")),     (Default Value)
```

```
// Bool input //
Array("NAME:Summary",       (Input name)
"Bool",                     (Boolean Input Type)
"Display Summary",          (Input Description)
Array(true)),               (Default Value)
// Number input //
Array("NAME:Version",       (Input name)
"Number",                   (Number Input Type)
"Script Version",           (Input Description)
Array(1021)))),             (Default Value)
// Trace selection for the solution and trace inputs //
Array("NAME:DocTraces",     (Document traces keyword)
// For input "DLMetrics" //
Array("NAME:DLMetrics",     (Input name)
// Trace defintion similar to the UDO. This trace definition is
a User defined solution //
Array("User Defined",
"", "DDR3 AC-Timing 4-DQ1", Array("Context:=", ""),
Array("Index:=", Array("All"), "Trise:=", Array("Nominal"),
"Tfall:=", Array("Nominal"), "Pulse_Width:=", Array("Nominal"),
"Data_Rate:=", Array("Nominal"), "Length:=", Array("Nominal")),
Array("Probe Component:=", Array("")), Array())),
// For input "DQ0" //
Array("NAME:DQ0",
// Trace defintion similar to the UDO. This trace definiton is a
Standard solution //
Array("Standard", "DQ0", "NexximTransient",
Array("NAME:Context", "SimValueContext:=", Array(1, 0, 2, 0,
false, false, -1, 1, 0, 1, 1, "", 0, 0, "DE", false, "0", "DP",
_
false, "20000000", "DT", false, "0.001", "WE", false, "100ns",
"WM", false,  _
"100ns", "WN", false, "0ps", "WS", false, "0ps")),
Array("Time:=", Array("All"), "Trise:=", Array( _
  "Nominal"), "Tfall:=", Array("Nominal"), "Pulse_Width:=",
Array("Nominal"), "Data_Rate:=", Array("Nominal"), "Length:=",
Array("Nominal")), Array("Probe Component:=", Array( _
  "DQ0")), Array())))
```

Python Script to Define Document

To define a document in Python script:

```
 [
    "NAME:Test Report",  "Test Report", "SysLib", "Examples/
TestUDDInputs",
     [
        "NAME:Inputs",
     [
        "NAME:DLMetrics","Solution", "Data Line Metrics", -1, -1
   ],
    [
      "NAME:DQ0", "Trace", "DQ0", -1, -1
    ],
    [
       "NAME:DQS", "Trace", "DQS", -1,  -1
    ],
    [
       "NAME:Name", "Text", "User Name", ["Sita Ramesh"]
    ],
    [
       "NAME:Summary", "Bool", "Display Summary", [True]
    ],
    [
       "NAME:Version", "Number", "Script Version" [1021]
    ]
  ]
 ],
 [
    "NAME:DocTraces",
    [
       "NAME:DLMetrics",
       [
```

```
            "User Defined",  "", "DDR3 AC-Timing 4-DQ1",
             [
                 "Context:="  , ""
             ],
             [
                "Index:="  , ["All"], "Trise:="  , ["Nominal"],
"Tfall:="  , ["Nominal"], "Pulse_Width:="  ,
                 ["Nominal"], "Data_Rate:="  , ["Nominal"],
"Length:="  , ["Nominal"]
            ],
            [
                 "Probe Component:=" , [""]
            ],
         []
     ]
  ],
  [
     "NAME:DQ0",
     [
        "Standard",  "DQ0",  "NexximTransient",
      [
         "NAME:Context",  "SimValueContext:=" ,
[1,0,2,0,False,False,-
1,1,0,1,1,"",0,0,"DE",False,"0","DP",False,"20000000","DT",Fals
e,"0.001","WE",False,"100ns","WM",False,"100ns","WN",False,"0ps
","WS",False,"0ps"]
     ],
     [
        "Time:="  , ["All"],"Trise:="  , ["Nominal"],"Tfall:="
, ["Nominal"],"Pulse_Width:="  ,["Nominal"],
         "Data_Rate:="  , ["Nominal"], "Length:="  , ["Nominal"]
    ],
    [
     "Probe Component:=" , ["DQ0"]
    ],
   []
```

```
   ]
  ],
 ]
```

Sample Script:- This one adds, edits, renames and deletes a document

```
Set oModule = oDesign.GetModule("UserDefinedDocuments")
' Add a UDD
oModule.AddDocument Array("NAME:Test Report1", "Test Report",
"SysLib",  _
"Examples/TestUDDInputs", Array("NAME:Inputs",
Array("NAME:DLMetrics", "Solution",  _
"Data Line Metrics", -1, -1), Array("NAME:DQ0", "Trace", "DQ0",
-1, -1), Array("NAME:DQS",
"Trace", "DQS", -1, -1), Array("NAME:Name", "Text", "User
Name", Array("Sita Ramesh")), Array("NAME:Summary",  "Bool",
"Display Summary", Array(true)), Array("NAME:Version",
"Number",  "Script Version"))), Array("NAME:DocTraces",
Array("NAME:DLMetrics", Array("User Defined",  "", "DDR3 AC-
Timing 4-DQ1", Array("Context:=", ""), Array("Index:=",
Array("All"), "Trise:=", Array( "Nominal"), "Tfall:=",
Array("Nominal"), "Pulse_Width:=", Array("Nominal"),
"Data_Rate:=", Array(  "Nominal"), "Length:=",
Array("Nominal")), Array("Probe Component:=", Array("")),
Array())), Array("NAME:DQ0", Array( _
"Standard", "DQ0", "NexximTransient", Array("NAME:Context",
"SimValueContext:=", Array( _
1, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "", 0, 0, "DE", false,
"0", "DP",   false, "20000000", "DT", false, "0.001", "WE",
false, "100ns", "WM", false,  "100ns", "WN", false, "0ps",
"WS", false, "0ps")), Array("Time:=", Array("All"), "Trise:=",
Array("Nominal"), "Tfall:=", Array("Nominal"), "Pulse_Width:=",
Array("Nominal"), "Data_Rate:=", Array("Nominal"), "Length:=",
Array("Nominal")), Array("Probe Component:=", Array("DQ0")),
Array())))

' Edit Document
oModule.EditDocument "Test Report1", Array("NAME:Test Report",
"Test Report",  _
```

```
  "SysLib", "Examples/TestUDDInputs", Array("NAME:Inputs",
Array("NAME:DLMetrics",  _
  "Solution", "Data Line Metrics", 1000001, 0),
Array("NAME:DQ0", "Trace", "DQ0", 32,  _
  2), Array("NAME:DQS", "Trace", "DQS", 32, 4),
Array("NAME:Name", "Text", "User Name", Array(  "Sita
Ramesh")), Array("NAME:Summary", "Bool", "Display Summary",
Array(true)), Array("NAME:Version",  "Number", "Script
Version"))), Array("NAME:DocTraces", Array("NAME:DLMetrics",
Array( "User Defined", "Solution", "DDR3 AC-Timing 4-DQ1",
Array("Context:=", ""), Array("Index:=", Array( "All"),
"Trise:=", Array("Nominal"), "Tfall:=", Array("Nominal"),
"Pulse_Width:=", Array( "Nominal"), "Data_Rate:=",
Array("Nominal"), "Length:=", Array("Nominal")), Array("Probe
Component:=", Array("")), Array())), Array("NAME:DQ0",
Array("Standard", "DQ1", "NexximTransient",
Array("NAME:Context", "SimValueContext:=", Array(1, 0, 2, 0,
false, false, -1, 1, 0, 1, 1, "", 0, 0, "DE", false, "0", "DP",
_
  false, "20000000", "DT", false, "0.001", "WE", false,
"100ns", "WM", false, "100ns", "WN", false, "0ps", "WS", false,
"0ps")), Array("Time:=", Array("All"), "Trise:=",
Array("Nominal"), "Tfall:=", Array("Nominal"), "Pulse_Width:=",
Array("Nominal"), "Data_Rate:=", Array("Nominal"), "Length:=",
Array("Nominal")), Array("Probe Component:=", Array("DQ1")),
Array())))

' Rename a UDD
oModule.RenameDocument "Test Report", "Test UDD Report"

' Update UDD
oModule.UpdateDocument "Test UDD Report"

' View Html
oModule.ViewHtmlDocument "Test UDD Report"

' View Pdf
oModule.ViewPdfDocument "Test UDD Report"

' Save Html
```

```
oModule.SaveHtmlDocumentAs "Test UDD Report", "c:/
AnsysProjects/Test.html"

' Save pdf
oModule.SavePdfDocumentAs "Test UDD Report", "c:/AnsysProjects/
Test.pdf"

' Delete UDD
oModule.DeleteDocument "Test UDD Report"
```

Note The product has to implement the GetModule call to create the UserDefinedDocument scripting object. For e.g. Check AltraSimDesign.cpp (function GetMgrIDispatch()).

Document Generator Interfaces

This document briefly describes the API interfaces available in the document generator plugin. (Ansys.Ansoft.DocGeneratorPluginDotNet.dll)

Scripting objects available in the script for the Generate function

- oApp = self.GetUDDAppContext(
  Gets the application context
  Usage:- Gets the active project ad the version of the product

```
oDesktop = oApp.GetAppDesktop()
             if  oDesktop != None:
                vr = oDesktop.GetVersion()
                oProject = oDesktop.GetActiveProject()
```

- oDesign = self.GetUDDDesignContext()
  Gets the design context
  Usage:- Gets the design name.

```
oDesign = self.GetUDDDesignContext()
          if oDesign != None:
              nm = oDesign.GetName()
```

- IUDDGenerator interface
  This interface available in the Generate method of the UDDPluginExtension.
  This interface can be used to
  1. Set the document output file path.

```
docgen.SetOutput("C:\\Examples\\DocumentOutput.xml")
```

  2. Get the document root.

```
docroot = docgen.GetDocumentRoot()
```

  3. Write out to the output file.

```
docgen.Write()
```

  4. Write Html document

```
 void WriteHTML();
```

5. Write PDF document
```
void WritePDF();
```
6. Load the Html transform object
```
void LoadHTMLTransform();
```

7. Load the cached PDF transform object
```
void LoadPDFTransform();
```

8. Get script path
```
string GetScriptPath();
```

9. Get output file path
```
string GetOutputFilePath();
```

- IUDDRoot interface

  Calling GetDocumentRoot() on the IUDDGenerator interface provides you with the this interface. This interface can be used to

  1. Add a new section to the document. Provide a section title.
```
section1 = docroot.AddSection("Section title")
```

  2. Add a new section to the document. Provide a section title and subtitle
```
section1 = docroot.AddSection("Section title", "Section
subtitle")
```

  3. Add a new title
```
section1 = docroot.AddTitle("Title")
```

  4. Add a new subtitle
```
section1 = docroot.AddSubtitle("Subtitle")
```

- IUDDSection interface

  Calling AddSection() on the IUDDRoot interface provides you with the this interface. This interface can be used to

  1. Set an ID for the section for internal links.
```
section1.SetID("id")
```

2. Add a new table to the document. Provide a table title.
   ```
   table1 = section1.AddTable("Table title")
   ```

3. Add a new image to the document. Provide an image title and a file path to the image file.
   ```
   image1 = section1.AddImage("Image title")
   ```

4. Add text to the document.
   ```
   text1 = section1.AddText("Random text........")
   ```

- IUDDImage interface

  Calling AddImage() on the IUDDSection interface provides you with the this interface. On this interface you can call the following methods

  1. Set an ID for the image for internal links.
     ```
     image1.SetID("id")
     ```

  2. Set alignment information . can be "center", "left" and "right".
     ```
     image1.SetAligment("center")
     ```

  3. Set the file path of the image file. Not necessary if image file path is set through the AddImage() method
     ```
     image1.SetFileRef("Image path")
     ```

  4. Set the format of the image file. Can be "BMP", "PNG", "JPEG", "JPG", "DVI" etc.. Not sure about the necessity of this one.
     ```
     image1.SetFormat("format")
     ```

- IUDDText interface

  Calling AddText() on the IUDDSection interface provides you with the this interface. On this interface you can call the following methods

  1. Set an ID for the text for internal links.
     ```
     text1.SetID("id")
     ```

  2. Set the emphasis attribute on the text
     ```
     text1.SetEmphasis()
     ```

  3. Set the quotes attribute on the text

```
Text1.SetQuotes()
```

4. Set the block quotes attribute on the text
   ```
   text1.SetBlockquotes()
   ```

5. Set quotes on the text
   ```
   Text1.SetQuotes()
   ```

6. Set the wordsize attribute on the text
   ```
   text1.SetSize(size as an integer)
   ```

7. Set a link to an ID of any element to provide internal links
   ```
   text1.SetLink("linkname")
   ```

8. Set an event link to handle an event. The HandleUDDEvents method should be implemented in the script to handle the event.
   ```
   text1.SetEventLink("linkname")
   ```

- IUDDTable interface

  Calling AddTable() on the IUDDSection interface provides you with the this interface. On this interface you can call the following methods

  1. Set an ID for the table for internal links.
     ```
     table1.SetID("id")
     ```

  2. Set alignment information . can be "center", "left" and "right".
     ```
     table1.SetAlignment("center")
     ```

  3. Set the background color of the table
     ```
     table1.SetBgColor(string bgcolor)
     ```

  4. Set the frame type. Can be "all", "bottom", "top,", "sides", "topbot"
     ```
     table1.SetFrame(string frame)
     ```

  5. Add a table group and specify the number of columns. A table can have multiple table groups.
     ```
     IUDDTableGroup table1.SetTableGroup(int columns)
     ```

- IUDDTableGroup interface

  Calling AddTableGroup() on the IUDDTable interface provides you with the this interface. On this interface you can call the following methods

  1. Set an ID for the table group for internal links.

     ```
     tgroup1.SetID("id")
     ```

  2. Set alignment information . can be "center", "left" and "right".

     ```
     tgroup1.SetAlignment("center")
     ```

  3. Set the column width of a column given the index of the column and the required width. Width can be set in 2 ways.
     - Width can be set relative to 1. E.g Setting it to "2*" makes the column width double the width of the others.
     - If the entire table width is considered to be 99.99 units. Width can be a number relative to this.

     ```
     tgroup1.SetColumnWidth(int index,  string width)
     ```

  4. Add a header to the table group

     ```
     IUDDTableRow tgroup1.AddHeader()
     ```

  5. Add a header with multiple rows to the table group. Takes number of sub rows.

     ```
     IUDDTableRow tgroup1.AddHeader(int rows)
     ```

  6. Add a row of content to the table group

     ```
     IUDDTableRow tgroup1.AddContent()
     ```

  7. Add content with multiple rows to the table group. Takes number of sub rows.

     ```
     IUDDTableRow tgroup1.AddContent(int rows)
     ```

- IUDDTableRow interface

  Calling AddHeader() & AddContent() on the IUDDTableGroup interface provides you with the this interface. On this interface you can call the following methods

  1. Set an ID for the table row for internal links.

     ```
     trow1.SetID("id")
     ```

  2. Set alignment information . can be "center", "left" and "right".

```
trow1.SetAlignment("center")
```

3. Set cell text. Can be cell content or header text. Takes a column index and a text string. It is added to the first row.

   ```
   IUDDTextElement trow1.Add(int  column,  string text)
   ```

4. Set cell text. Can be cell content or header text. Takes a column index, row index and a text string. Takes in a row number because a table row can have multiple sub rows.

   ```
   IUDDTextElement trow1.Add(int  column,  int subrow, string
   text)
   ```

5. Set cell content. Takes a column index and an int value. It is added to the first row.

   ```
   IUDDTextElement trow1.Add(int  column,  int value)
   ```

6. Set cell content. Takes a column index, row index and a int value.

   ```
   IUDDTextElement trow1.Add(int  column,  int subrow, string
   text)
   ```

7. Set cell text. Takes a column index and a double value. It is added to the first row.

   ```
   IUDDTextElement trow1.Add(int  column,  double value)
   ```

8. Set cell text. Takes a column index, row index and a double value.

   ```
   IUDDTextElement trow1.Add(int  column,  int subrow, double
   value)
   ```

9. Set cell text spanning 2 columns. Can be cell content or header text. Takes a sub row index , starting column index., ending column index and a text string.

   ```
   IUDDTextElement trow1.AddSpanningcolumnst(int subrow, int
   columnstart,  int columnend,  string text)
   ```

10. Set cell text. Can be cell content or header text. Takes a column index, starting sub row index, ending sub row index and a text string.

    ```
    IUDDTextElement trow1.AddpanningRows(int  column, int
    subrowstart,  int subrowend,  string text)
    ```

- IUDDTableRow interface

  Calling Add() on the IUDDTableGroup interface provides you with the this interface. On this

interface you can call the following methods

1. Set an ID for the table row for internal links.

   ```
   trow1.SetID("id")
   ```

2. Set alignment information . can be "center", "left" and "right".

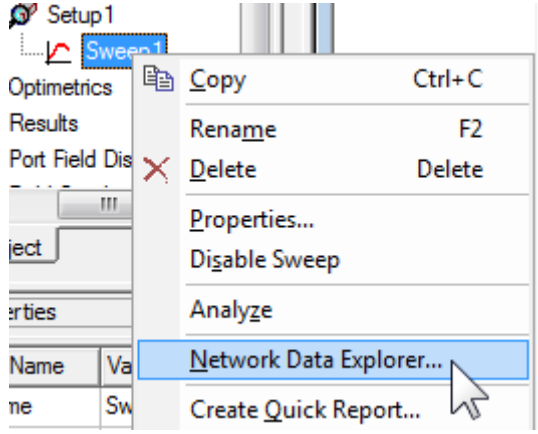
   ```
   trow1.SetAlignment("center")
   ```

Includes all the methods exposed by the IUDDText interface.

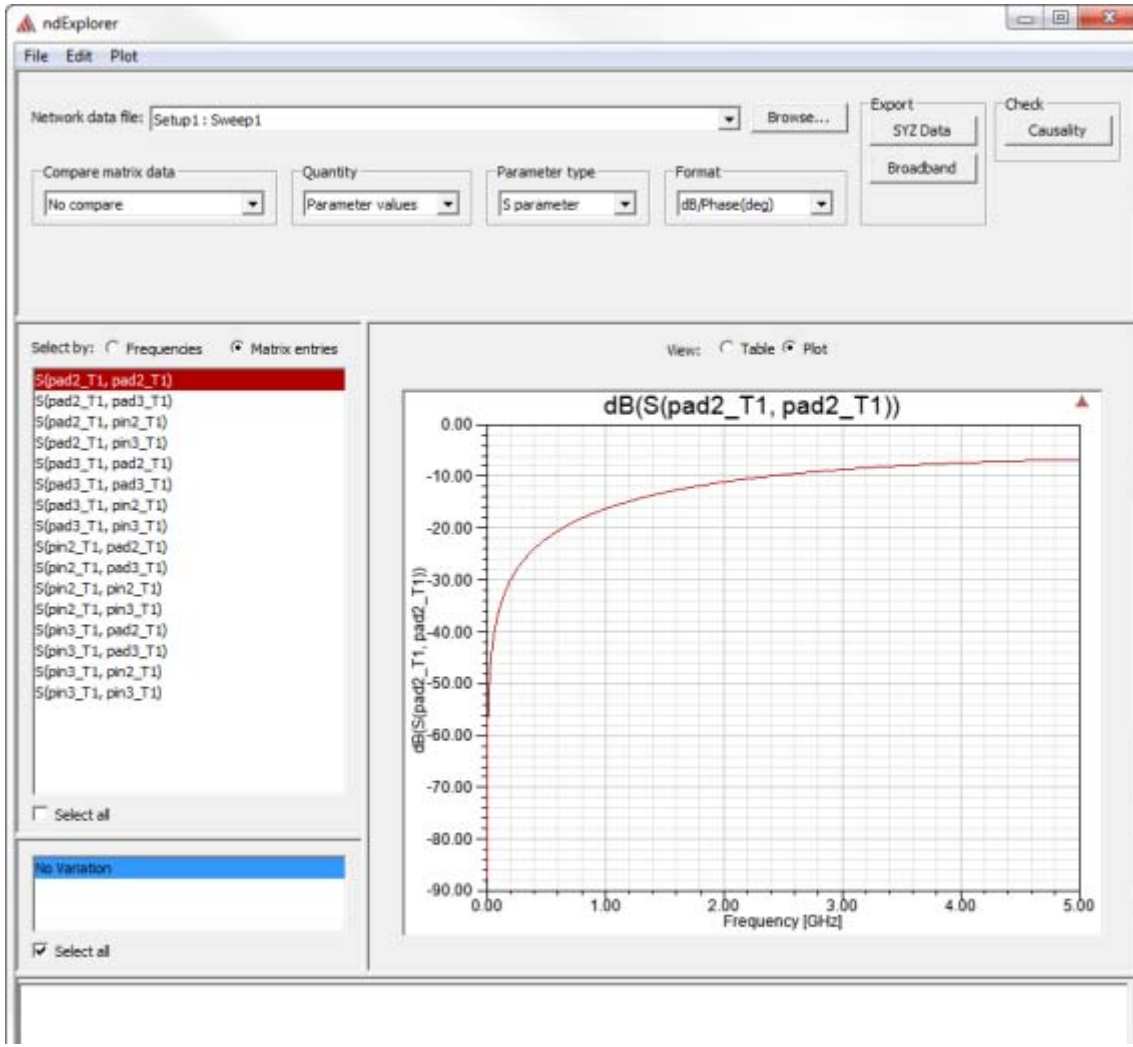
18

Network Data Explorer

The network data explorer, NdExplorer, provides visualization, analysis, and manipulation tools for network data. It is accessible on the shortcut menu for Sweeps in the project tree and also through **Tools > Network Data Explorer**.



If you have performed a simulation for a sweep, selecting **Network Data Explorer** from the Sweeps shortcut menu automatically loads the data into NdExplorer.



The topics for NdExplorer include:

NdExplorer Overview

Loading Data Into NdExplorer

Exporting Data from NdExplorer

Edit Menu Commands

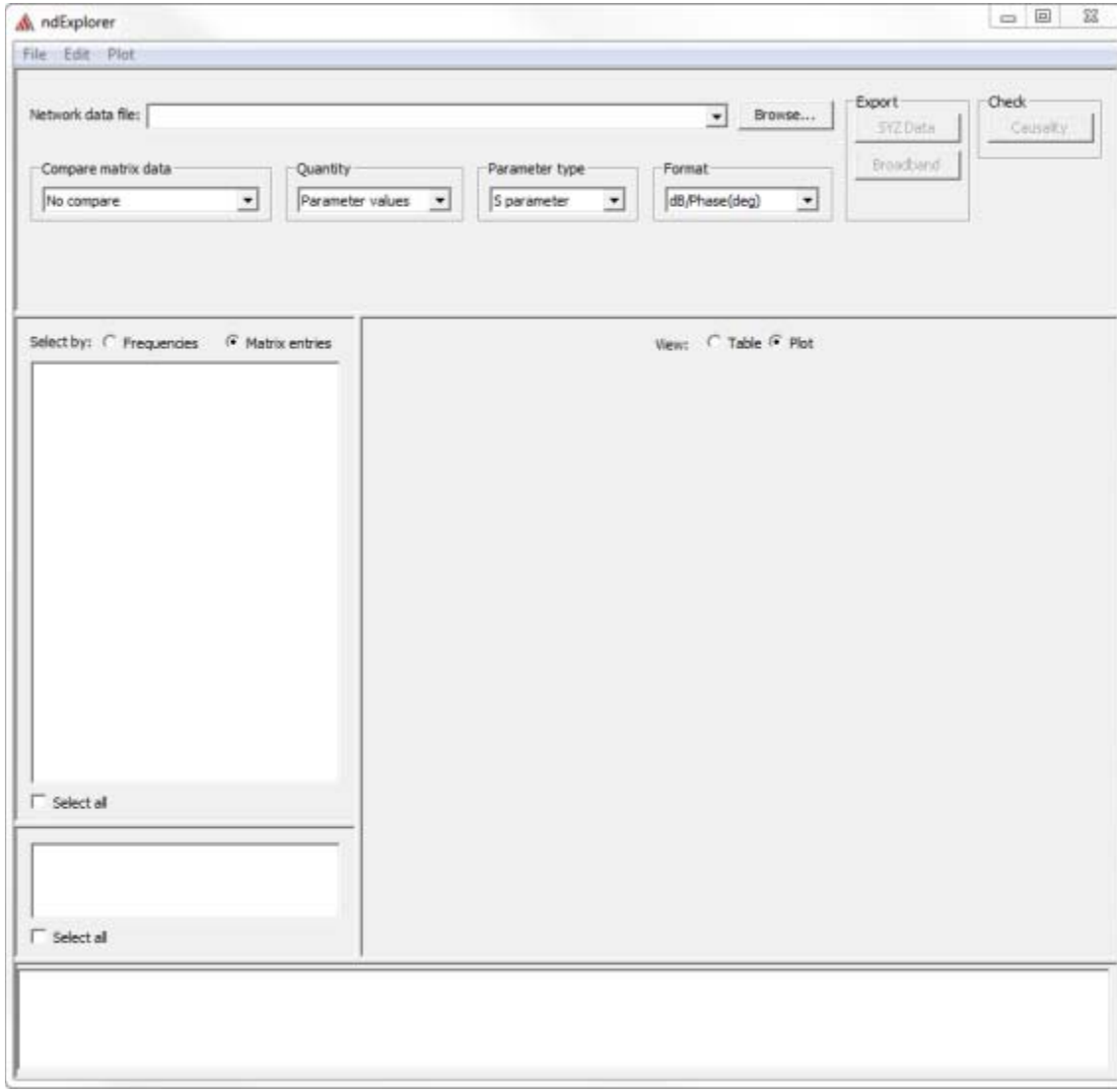
Data Display Pane Context Menus

Exploring Network Data and Modifying the Display

NdExplorer Overview

To open the NdExplorer, click **Tools > Network Data Explorer**. The NdExplorer window is divided into the following panes:

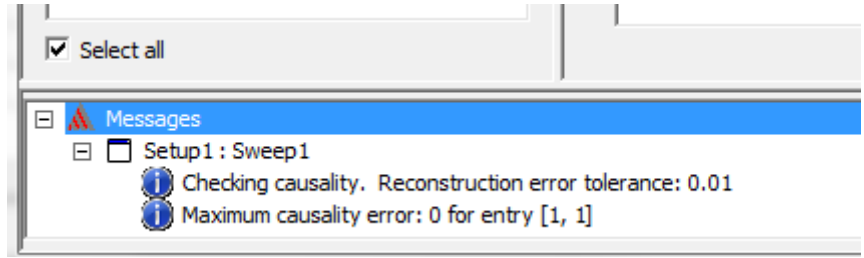
- **Control Pane** across the top
- **Data View Pane** to the right
- **Data Selection Panes** to the left — divided into **Frequencies** or **Matrix Entries** selection and **Variation Selection**



Control Pane

The control pane is used to select the network data to be viewed as well as the manner in which it is to be displayed.

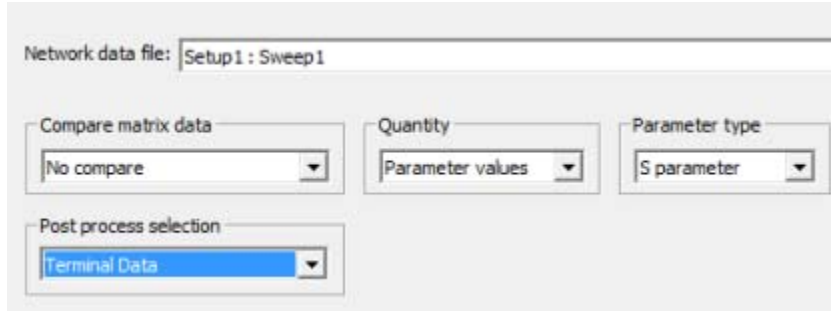
- **The File** menu provides commands for **Open, Close** and **Exit**, as well as equivalents for **Browse, Check Causality**, and **Compare**.
- The **Edit** menu provides a range of commands for editing properties and display features.
- **Network data file**: Name of the currently active data set.
- **Browse**: Clear the current data sets and browse for a new file to load. Equivalent to **File>Open**.
- **Export:** Quick buttons for export; same options as those accessed via the **File** menu.
- **Check Causality**: button to check causality for the active data set. Equivalent to **File>Check Causality**. After a check, you can look in the Message pane in the lower-left to see the result.



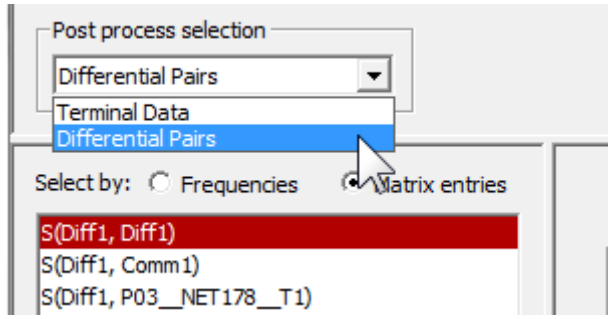
- The **Plot>Log Frequency** menu toggles the log representation of matrix data.
- **Compare matrix data**: Lets you choose a comparison data set (or none at all). Equivalent to **File>Compare**. See Comparing Network Data for an example.)
- **Quantity**: The quantity to display (parameter or statistical).
- **Parameter type**: S, Y, or Z parameters, Port Impedance, or Gamma.
- **Format**: The display function to apply to the data: magnitude/phase, real/imaginary, dB/phase, magnitude, phase, dB, real, or imaginary.

Post Process Selection Menu

If your design includes Differential Pairs, the Control pane includes a Post Process selection menu.

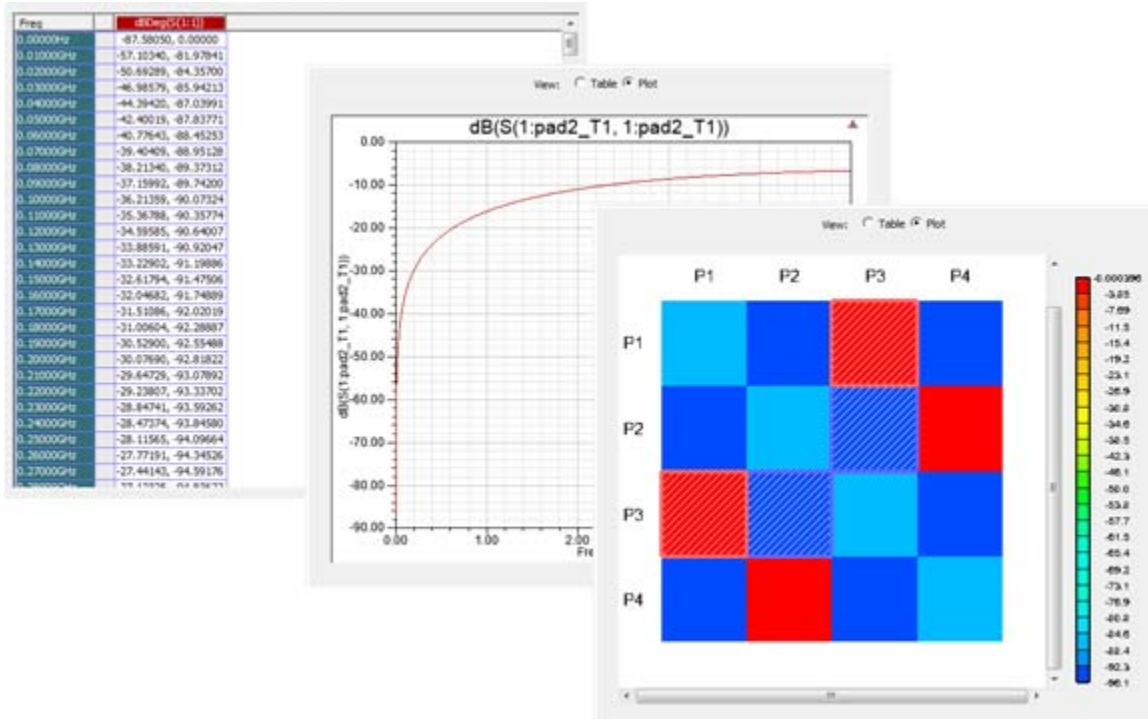


The Post process selection menu lets you select Terminal Data or Differential Pairs.



Data View Pane

The data view pane supports tabular views of the matrix data and graphical representations.



Different context menus appear in the data view pane, depending on your view selections.

Data Selection Pane

The data selection pane allows you to choose to display data by frequency or by matrix cell, and to choose which frequencies or cells to display.

- The **variation list box** allows you to control which variations are currently active; this affects all displays (table, plot, and statistics). When displayed by frequency, the entire matrix is presented in the data view pane for each selected frequency; when by matrix cell, the data for the individually chosen cells is shown across all frequencies.
- **Frequencies unavailable for a variation** are dimmed and are not highlighted when selected.
- Checking **Select all** causes all items within the list box to be selected; the corresponding data is displayed in the data view pane. **Warning**: given the volume of data in many network data sets, this display may take a considerable time to generate, especially when the graph plot view is enabled.

Loading Data Into NdExplorer

When launching **NdExplorer** from within HFSS by using the right click menu on Sweeps, the current solution data is automatically loaded and ready for viewing. Alternatively, you can also open a file from within **NdExplorer**.

- The file browser may be accessed via the **File > Open** command or by clicking the **Browse** button.

The file browser allows you to open multiple files at a time. However, the displayed data always corresponds to the data set indicated in the **Network data file** field on the control pane; you can use its drop-down list to switch between primary datasets.

- Use the **Files of type** pulldown menu to select from the following file formats: Touchstone Format, Touchstone 2 Format, Citifile, or Neutral Format.
- All data sets are available for data comparison, and all data sets which are open for comparison are also available as primary data sets.
- Opening one or more new files clears the current data sets. Additional data sets may be opened through the **Compare matrix data** control field, but this option does not clear the current data sets. For more information see Comparing Network Data.

Exporting Data from NdExplorer

NdExplorer allows you to export data to a file in a variety of different formats. You can also create an Nport Model, which exports the active data set back to the project.

The topics for this section include:

Scripting

Export SYZ Data

Export Macro Model

Scripting for NdExplorer

Scripting is available for each NdExplorer export method, which means a script can be recorded to duplicate the export process. NdExplorer can be invoked in the following contexts, and scripting is available from the Project Context and the Design Instance Context (simulation setup):

- No design — When there is no design available, the export functionality is not available, and so scripting is not available.
- Project Context — In the context of a project, you can open a touchstone file and then export. Scripting is available.
- Design Instance Context — In the context of a solution (RCM from the simulation setup in a design), you can export the corresponding network data solution. Scripting is available.

For more information, see NdExplorer Script Commands in the HFSS Scripting Guide

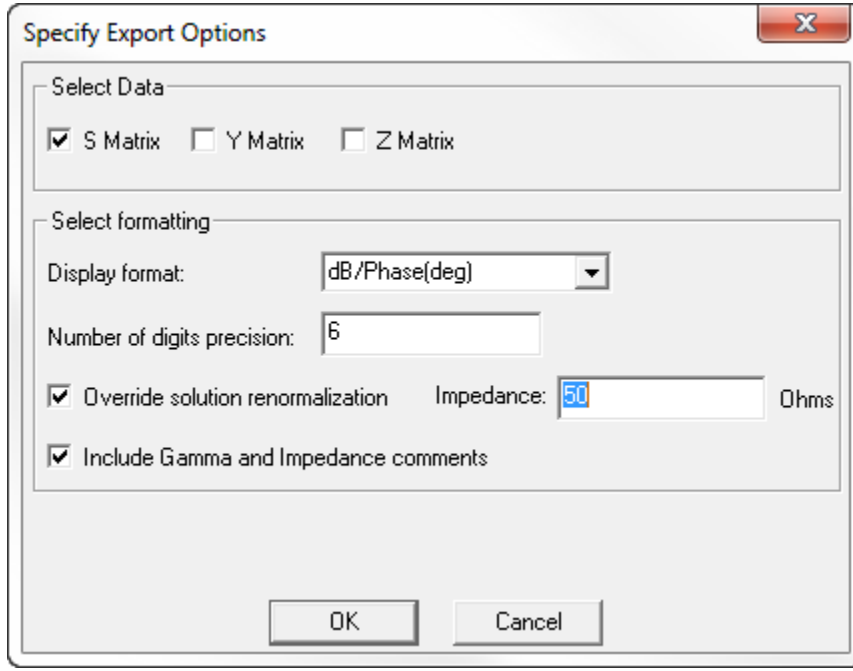
Export SYZ Data

The active data set may be saved as a touchstone file (under a name different to the original) with the **File > Export SYZ Data** command. At the time of export, you can select from the following options:

- Precision
- Type of data
- NPort model
- Renormalize data

Touchstone Export

After clicking **File > Export SYZ Data** and specifying a file name and Touchstone format (*.sNp) in the **Save File As** dialog, the **Specify Export Options** dialog opens:

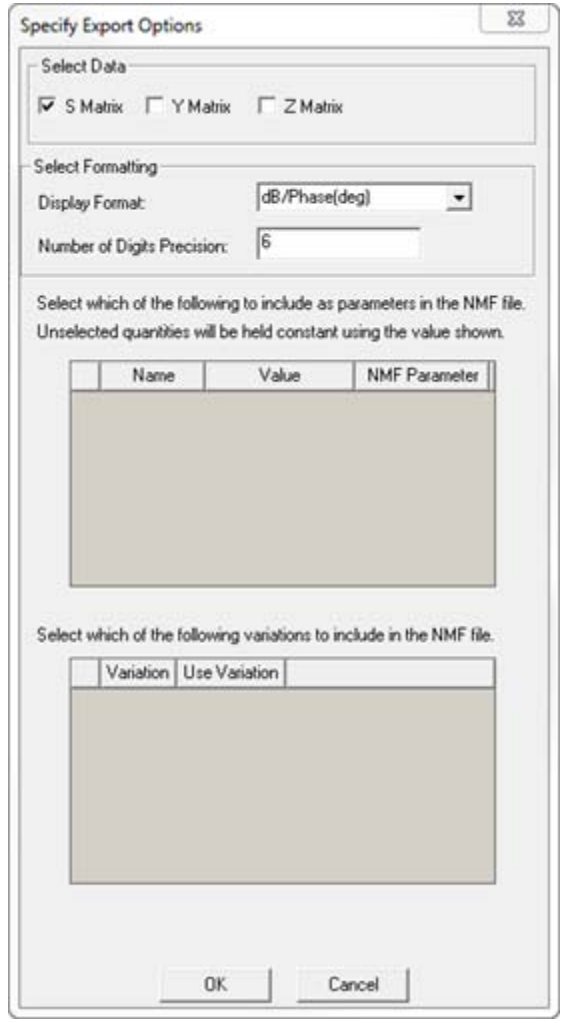


You are then able to:

- Select the data
- Select the display format
- Enter the precision
- Reference Impedance to renormalize
- Include impedance comment
- Create NPort model

NMF Export

After clicking **File > Export SYZ Data** and specifying a file name and the NMF format (*.nmf) in the **Save File As** dialog, the **Specify Export Options** dialog opens:

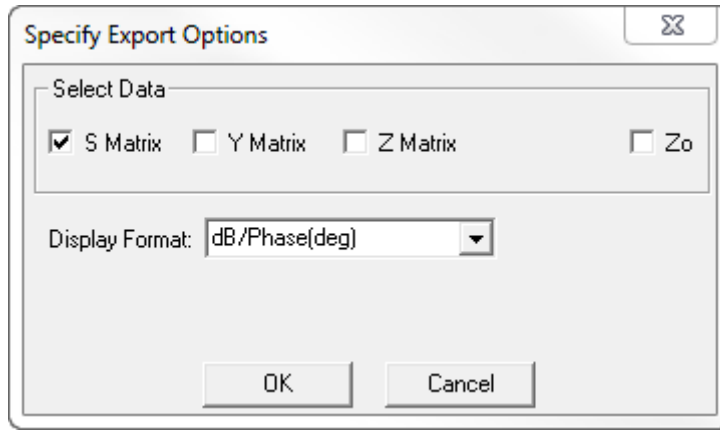


You are then able to:

- Select the data
- Select the display format
- Enter the precision
- Select/unselect the listed parameters
- Select the variations

Spreadsheet, Matlab, and Citifile Export

After clicking **File > Export SYZ Data** and specifying a file name and the format — Spreadsheet (*.tab), Matlab(*.m), or Citifile(*.cit) — in the **Save File As** dialog, the **Specify Export Options** dialog opens:

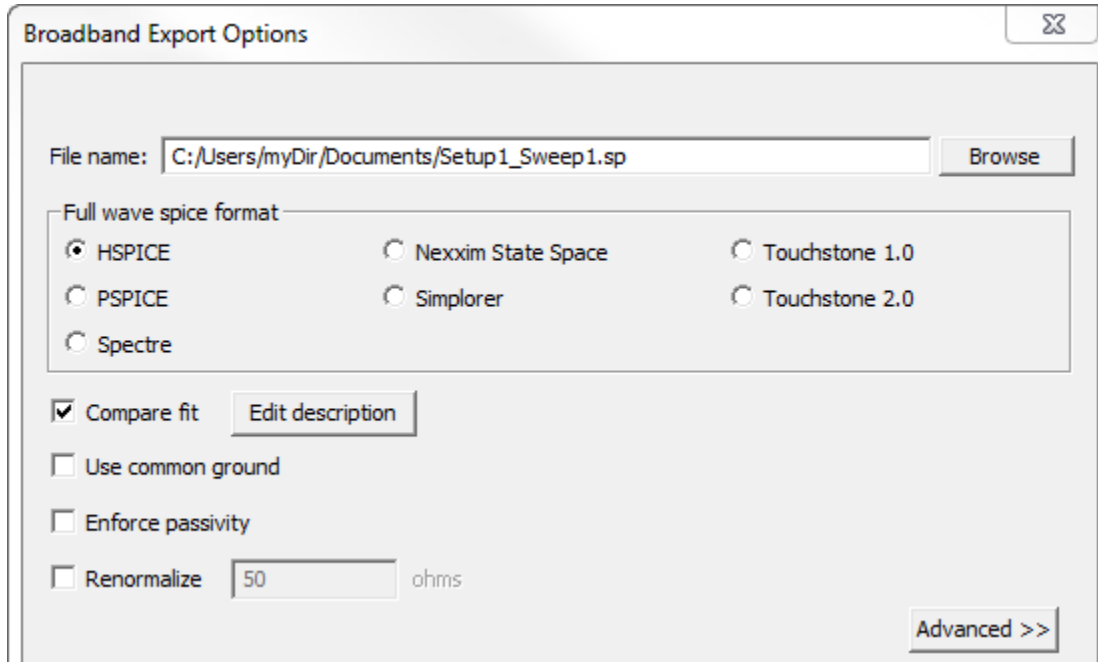


You are then able to:

- Select the data
- Select the display format

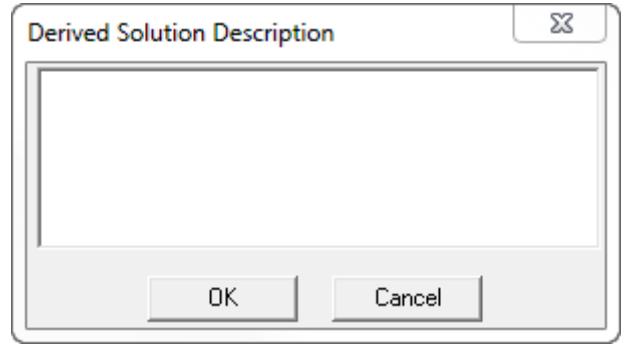
Export Macro Model

NdExplorer lets you export macro model data in a variety of different formats. In the NdExplorer window, click **File > Export MacroModel** and select either **Broadband** or **Lumped** (presently unimplemented) to open an export dialog.



The following controls are available:

- **File Name** — Use the default file name or browse to a different file/location.
- **Full Wave Spice Format** — Select the export format.
- **Compare Fit**. The **Edit description** button opens an box for adding text**.**



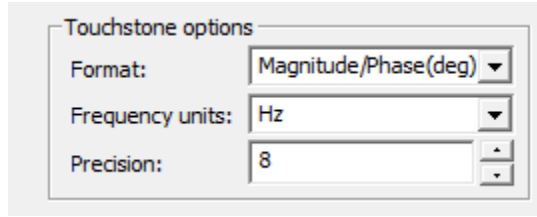
- **Use Common Ground** — Select this box to use common ground.
- **Enforce Passivity** — Select this box to enforce passivity.

- **Renormalize**

Clicking the **OK** button begins the export, which is implemented by a standalone process -- genequiv.exe -- that is available from the designer installation. Additionally, for Nexxim State Space and Touchstone formats, there is an option to automatically create an Nport model using the newly created file.

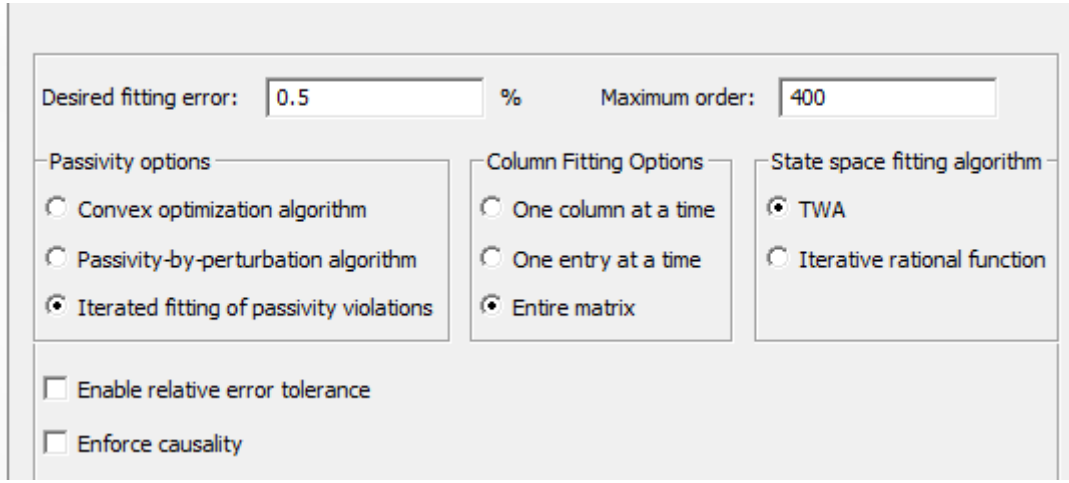
Touchstone Options

When Touchstone 1.0 or Touchstone 2.0 is selected as the format, the following **Touchstone Options** are displayed:



Advanced Options

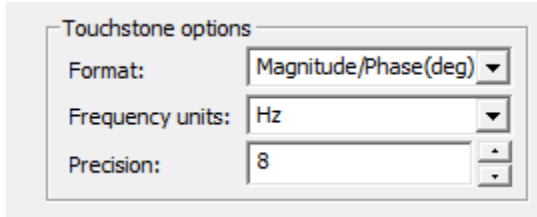
Click the **Advanced** button to display the advanced options.



The following advanced options are available:

- Desired fitting error
- Maximum order
- Passivity Options
- Column Fitting Options
- State Space Fitting Algorithm
- Enable relative error tolerance (works best with TWA State-Space Fitting algorithm, not recommended with Iterative Rational Function fitting.)

- Enforce causality
- Renormalize
- If you select Touchstone format, additional fields appear:



State Space Fitting — Tsuk-White Algorithm

The "Tsuk-White Algorithm" (TWA) is an ANSYS-proprietary method for fitting a state space model to extracted s-parameter data (Adding Frequency Sweeps). It uses techniques based on Singular Value Decomposition (SVD) to quickly determine required number of poles for fitting a model. It is also capable of simultaneously finding sets of poles that match multiple matrix entries. This can result in a much smaller state space model at the end of the process which improves performance when the model is a component in a transient spice simulation. Due to its many benefits, TWA is the preferred method for state space fitting.

The iterated rational fitting approach takes a matrix of S-parameter data and, for each matrix entry, tries a succession of different pole-zero approximations (increasing the number of poles used at each iteration) until it can find an acceptable fit to the data. For broad frequency sweeps and large numbers of excitations this process can be time consuming because of all the iterations and is not guaranteed to produce a good fit to the data. It is retained as a fallback if the TWA algorithm fails

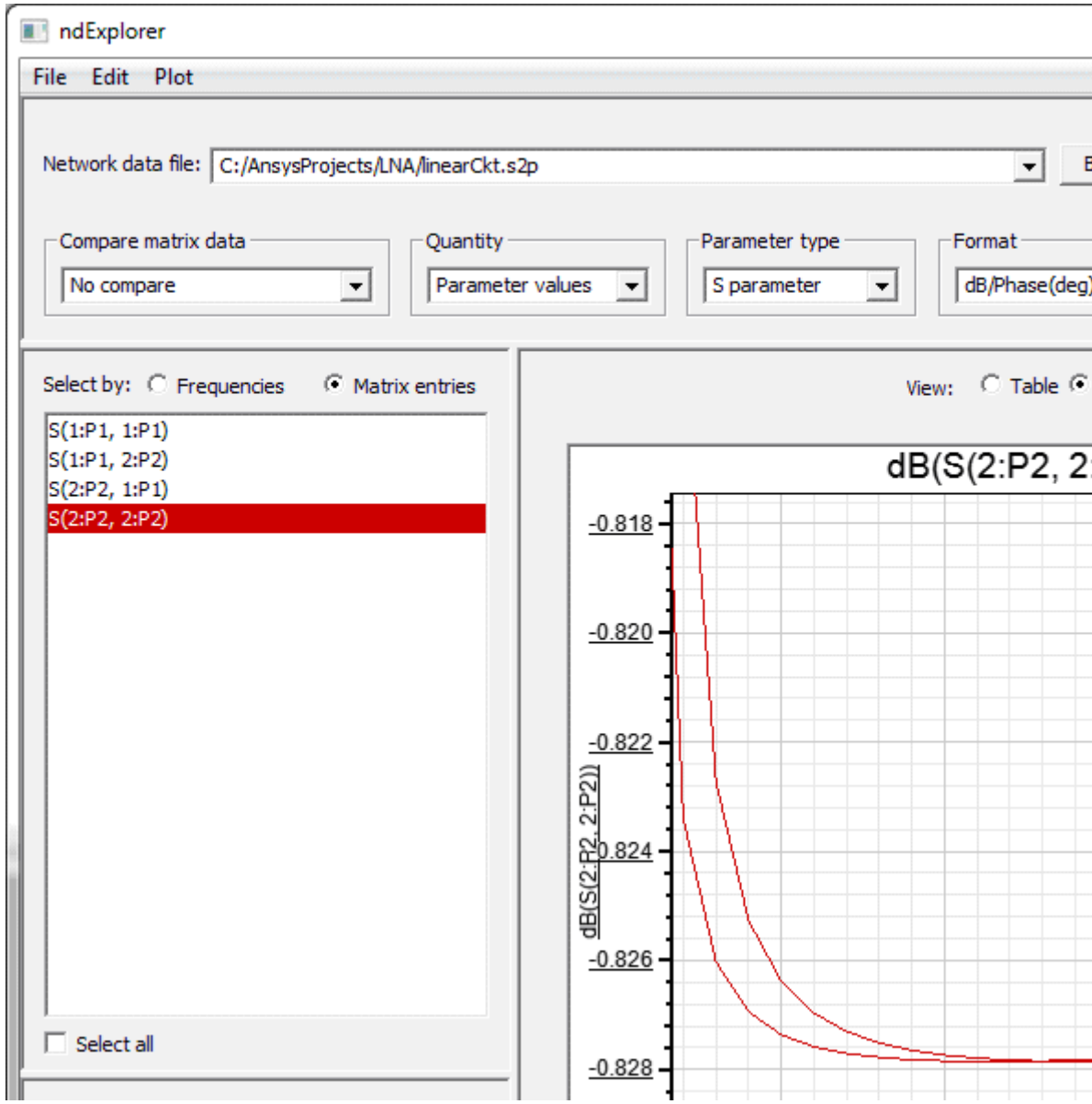
Note that the Enable Relative Error Tolerance option works best with the TWA fitting algorithm, and is not recommended for use with iterated rational fitting.

Comparison of Original S-Parameters with Exported S-Parameters

The Broadband export dialog displays a checkbox option **Add as derived solution**. When this box is checked, the exported file is read in as a s-parameter solution and loaded into NdE for comparison. This option is visible for all export formats and is turned on by default. For HSPICE, Nexxim State Space and the Touchstone export, if the solution was renormalized during the export, then the original solution impedances are applied on importing back.

The **Edit** button opens a rich-text edit box where you can give the solution a description which will be displayed in NdE to better identify the imported solution.The solution is added to NdExplorer as a derived solution so that you can compare this solution with the original solution. All the features of comparison that are available in NdExplorer can be used to compare the original and derived solutions.

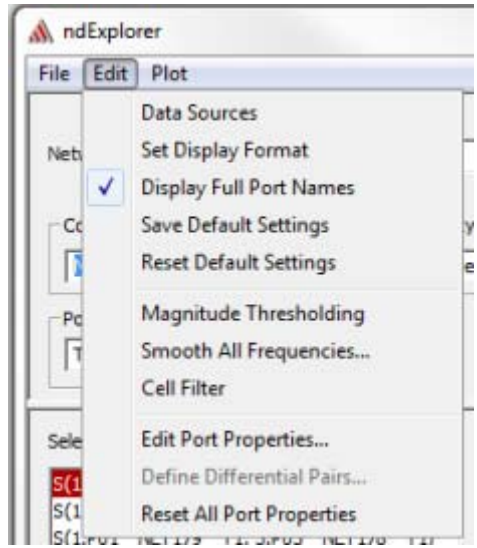
~DerivedSolution='1' is the exported solution. When you hover over the derived solution line in the list box below, the tool tips displays the description entered in the export dialog.



The import of the exported solution is now done by reading the exported Touchstone file.

NdExplorer Edit Menu Commands

The **Edit** menu on the **Control** pane includes several commands..



The topics for this section include:

Data Sources

Set Display Format

Display Full Port Names

Save or Reset Default Settings

Magnitude Thresholding

Smooth All Frequencies

Cell Filtering

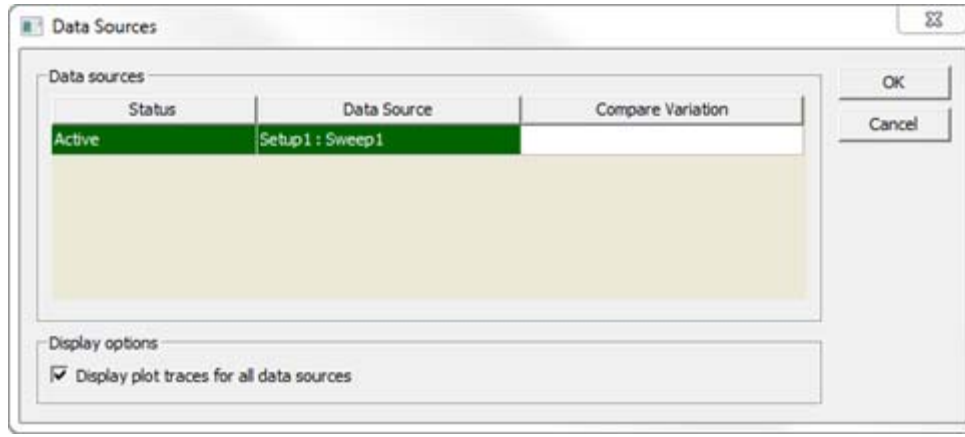
Edit Port Properties

Reset All Port Properties

The Define Differential Pairs command is disabled in HFSS and HFSS-IE because they handle differential pairs data differently than NdExplorer. If you have defined differential pairs in HFSS or HFSS-IE, you can see them by using the Post Process Selection menu.

NdExplorer Edit Data Sources

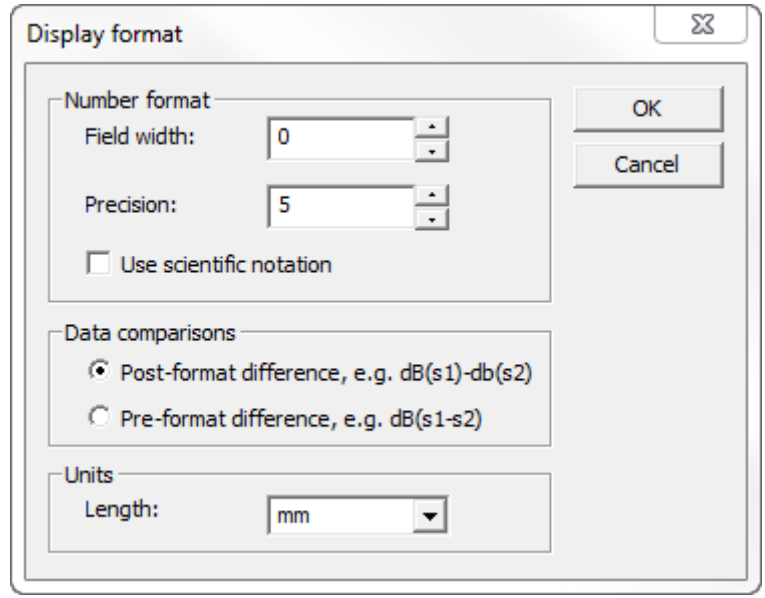
Click **Edit>Data Sources** to open the **Data Sources** dialog.



This shows the current status source, and lets you specify whether to Display plot traces for all data sources.

NdExplorer Display Format

Select **Edit>Set Display Format** to open the **Display format** dialog. The **Display format** dialog allows the user additional control over the display of values throughout NdExplorer.

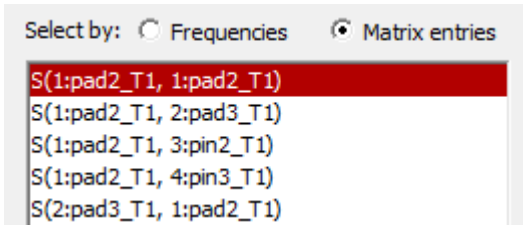


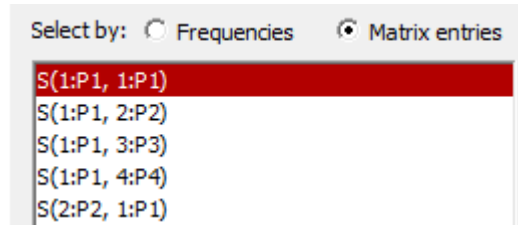
- **Field width**: The minimum number of characters used to display a number; values are padded with blanks as necessary; use to increase the blank space displayed in the data tables to force an increase in column width.
- **Precision**: The number of decimals to display.
- **Use scientific notation**: When checked, values are displayed using scientific notation, e.g. 3.88124e-001.
- **Post-format difference**: When comparing data sets, subtract values after applying the formatting function (e.g. dB, magnitude, etc.); the values displayed will be the difference between the magnitude, dB, and so on.
- **Pre-format difference**: When comparing data sets, subtract values before applying the formatting function (e.g. dB, magnitude, etc.); the values displayed will be the magnitude, dB, etc., of the complex difference.
- **Length**: The default units used to display and interpret length values (e.g. de-embedding length).

NdExplorer Display Full Port Names

By default, port names are displayed in an abbreviated form: P1, P2, etc. This applies to the matrix display and to the cell list display. Use the **Edit>Display Full Port Names** command to toggle the display of port names. Note that tool-tips always display the full port name.

.



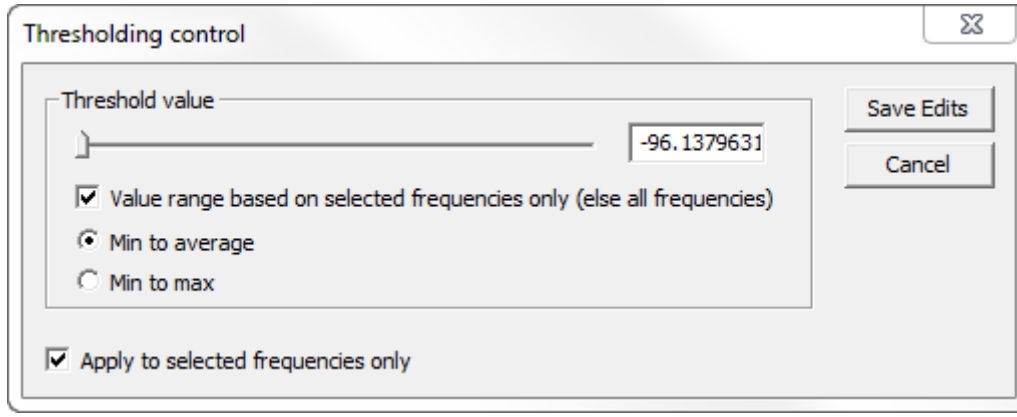


NdExplorer Save or Reset Default Settings

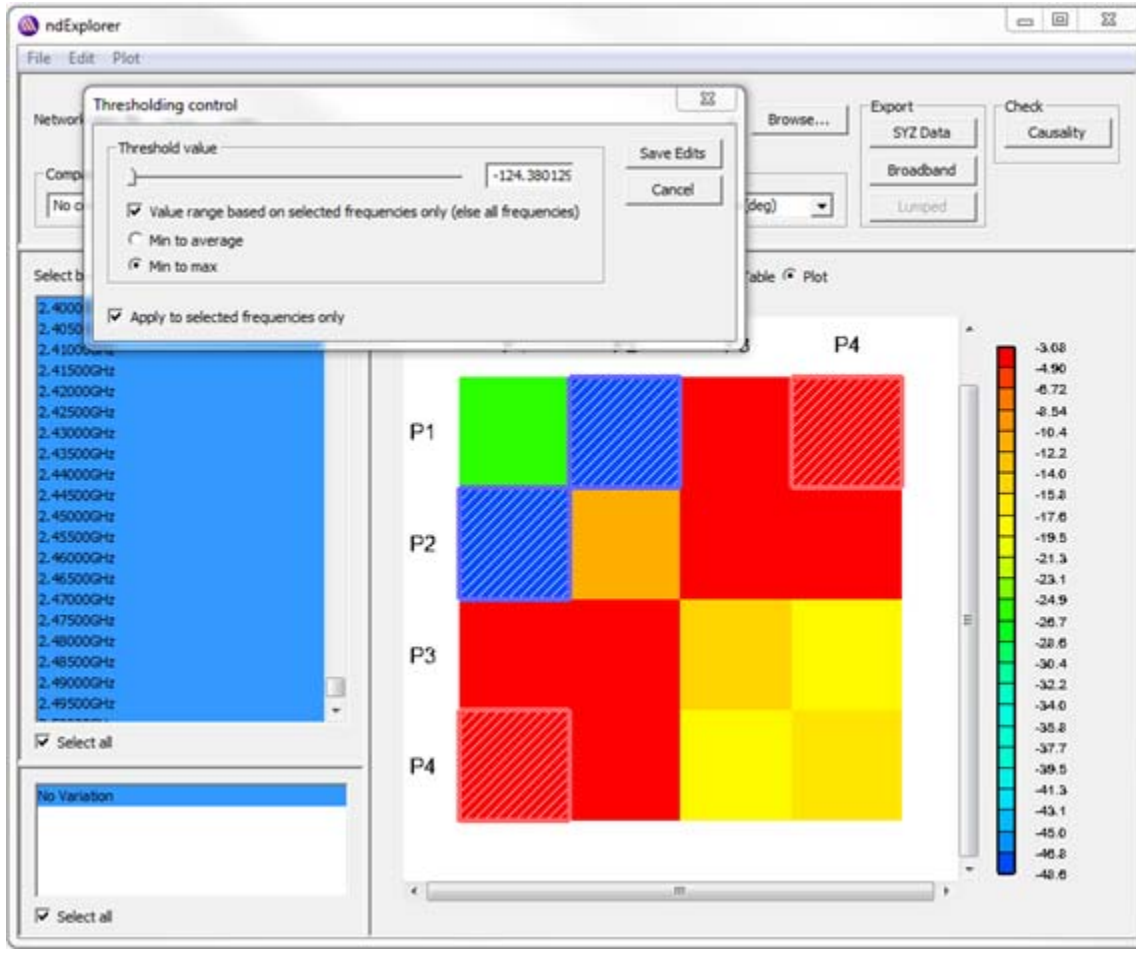
Use **Edit>Save Default Settings** to create your own defaults, or **Edit>Reset Default Settings** to return to the original settings.

NdExplorer Magnitude Thresholding Control

Use **Edit>Magnitude Thresholding Control** to open the **Thresholding Control** dialog.



The **Thresholding control** is a modeless dialog accessed through the **Magnitude Thresholding** option on the **Edit** menu. Thresholding is a way of reducing matrix entries to zero based on a value threshold. Thresholding applies to the displayed data, e.g. Magnitude(S), but it is the original matrix value that is reset to zero. Hence, thresholding is affected by the **Parameter type** and **Format** settings.



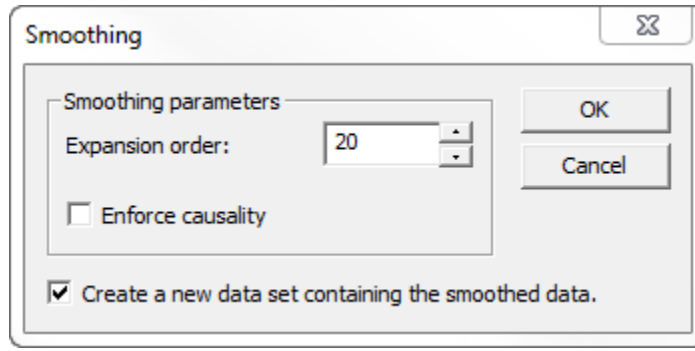
Set the threshold value by sliding the control or by explicitly entering a value. The data range for the slider is determined by the following dialog options:

- **Value range based on selected frequencies only**: When selected, the slider value range is determined from the selected frequencies only, otherwise, the entire data set is used to determine the range.
- **Min to average**: When selected, the slider data ranges extends from the data minimum to the mean.
- **Min to max**: When selected, the slider data range extends over the entire data range, i.e. from the minimum to the maximum data values.
- **Apply to selected frequencies only**: When selected, the **Save edits** button only modifies

the currently selected network data matrices, otherwise, the matrix data for all frequencies and variations are adjusted.

Smoothing

The data smoothing dialog is accessed via the **Smooth All Frequencies** option on the **Edit** menu.

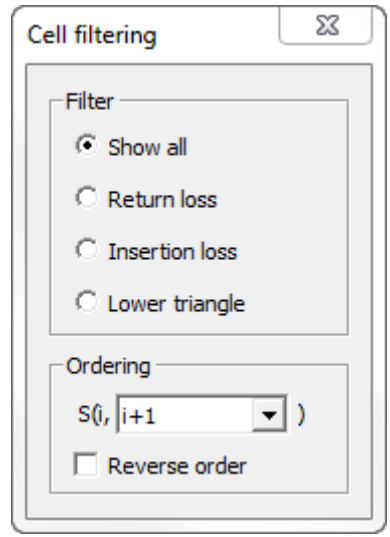


A least-squares polynomial fit of the specified order is used to interpolate new data points for the magnitude and phase components of the S-parameters.

- **Left window size** and **Right window size** control the number of data values collected on either side of the point to be interpolated — please note that these are point counts and not distances.
- **Spline order** can not exceed the total number of points sampled (i.e. the sum of the two window sizes).

Cell Filtering

The cells available in the data selection pane may be restricted using the **Cell Filtering** dialog by selecting **Edit > Cell Filter**.



The **Cell filtering** dialog is modeless and the filters are immediately applied to the cell list. Filtering remains in effect even after the dialog has been closed.

For an n-port model with a total of 2n pins in the standard arrangement, the choices are:

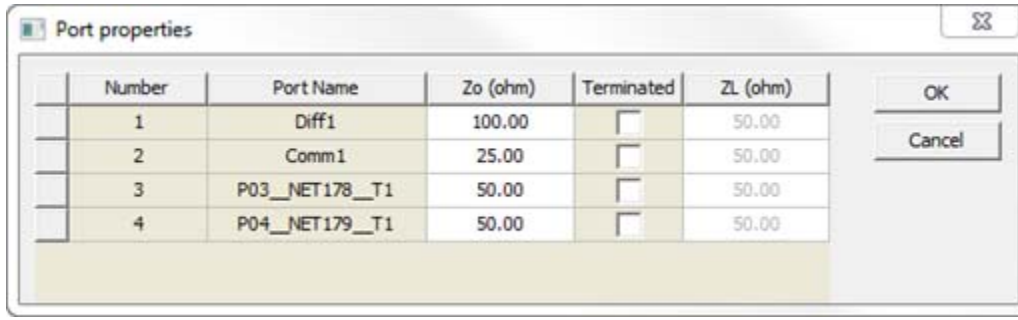
- **Show all** the available cells. There are n-squared choices.
- **Return loss**, show S(i, i). There are n choices.
- **Insertion loss**, show S(i, i+1). There are n choices.
- **Lower triangle**, show S(i, j) for all j < i. There are n(n-1)/2 choices.

Three pin arrangements are recognized:

- **S(i, i+1)** and its **Reverse order**, S(i, i-1)
- **S(i, i+n)** and its **Reverse order**, S(i, i-n)
- **S(i, 2*n-i+1)** and its **Reverse order**, S(i, 2*n-i-1)

Changing Port Properties and Reducing Matrix Size

Access the **Port Properties** dialog through the **Edit Port Properties** option on the **Edit** menu. The normalization impedance, termination, port order, gamma values, and de-embedding distance may all be edited through this dialog.

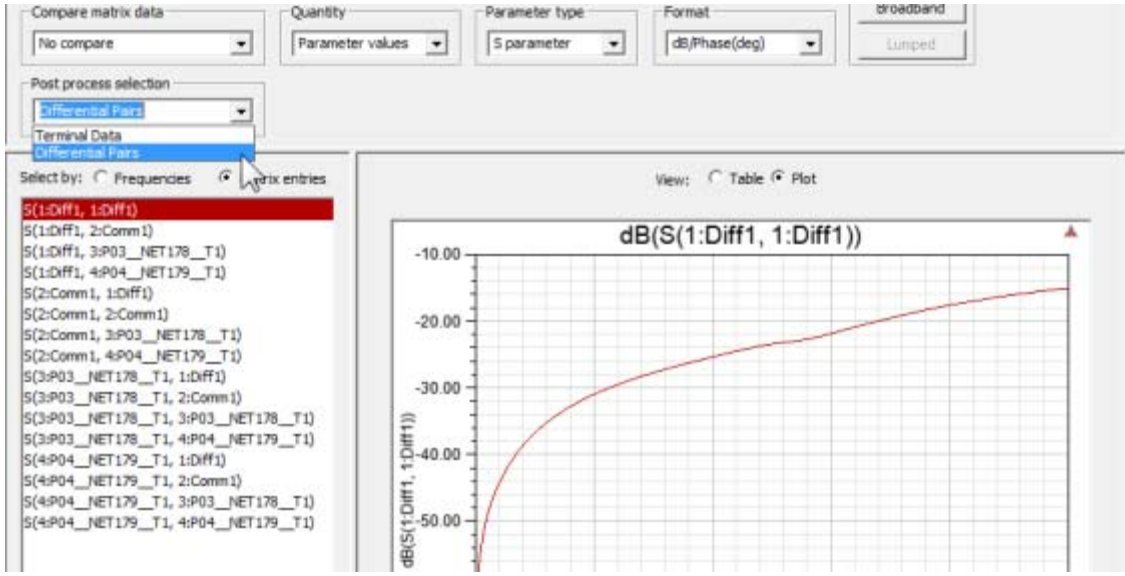


- Impedance values are specified in ohms and may be complex, the accepted syntaxes are:
  - — real, e.g. 50
  - — real + imag i, e.g. 50+5i
  - — mag i, e.g. 5i
- Explicit units may be used with the impedance value (e.g. 0.5kOhm) but will always be shown in ohms (when the dialog is redisplayed).
- Reorder ports by clicking and dragging their row. To see the port names in the data view displays, use the right-click menu option **Full Port Names**.
- Terminated ports are eliminated from the matrix thereby reducing the matrix size. Existing data sets with mismatching port numbers will no longer be available for data comparisons.
- Port properties may be restored to their load time values using the **Reset All Port Properties** option on the **Edit** menu.
- The **de-emebedding** column only appears if gamma values are available (specifically, if **NdExplorer** is invoked from the results menu within Designer). Values are always displayed in the units indicated in the column header, however, values may be entered with specific units, e.g. "1.5in". The default units may be changed via the **Set Display Format** dialog (available on the **Edit** menu or the right-click context menu).

Once the changes have been committed by selecting **OK**, the modified matrix values may be saved to a file by using the **File > Save As** option.

Displaying Mixed-Mode Parameters using Differential Pairs

Mixed-mode parameters are displayed in NdExplorer place of single-ended parameters when differential pairs are both defined and activated in HFSS or HFSS-IE and you select Differential Pairs as the Post Process selection.



Reset All Port Properties

Reset All Port Properties clears reference impedances and terminations, but does not disable all differential pairs. To disable all differential pairs, you must disable them in HFSS or HFSS-IE.

Data Display Pane Context Menus

The Data Display pane presents different right-click menus depending on the context. Some of the commands are the same as those on the Edit menu. Others only appear in the context menus.

- Data Display Pane: Matrix Entries and Table Selected
- Data Display Pane: Frequencies and Table Selected
- Data Display Pane: Matrix Entries and Plot Selected
- Data Display Pane: Frequencies and Plot Selected

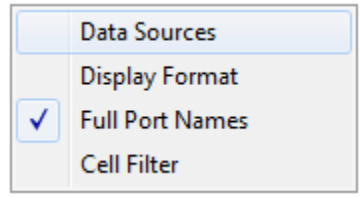
The NdExplorer commands unique to the context menus are:

- Multiple Frequency Statistics
- Highlight Min/Max
- Select Transpose
- Select Compare Variations
- Color Legend Attributes

The **Plot>Log Frequency** menu command also affects the display pane, and lets you toggle to log frequency display of traces and values.

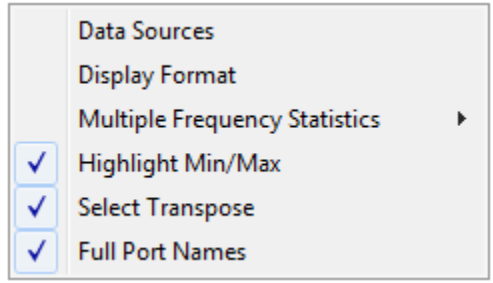
Data Display Pane: Matrix Entries and Table Selected

The shortcut menu only provides commands included in the Edit menu.



Data Display Pane: Frequencies and Table Selected

The shortcut menu provides the following commands.

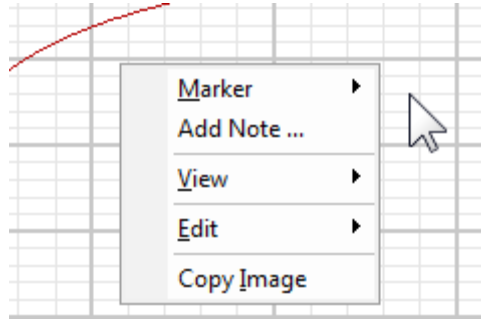


The **Data Sources**, **Display Format**, and **Full Fort Names** commands are the same as in the Edit menu. The commands accessed through this context menu are:

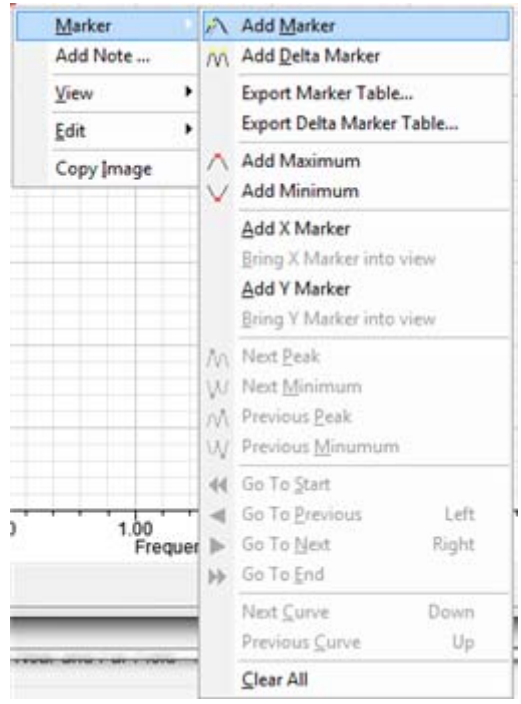
- Multiple Frequency Statistics
- Highlight Min/Max
- Select Transpose

Data Display Pane: Matrix Entries and Plot Selected

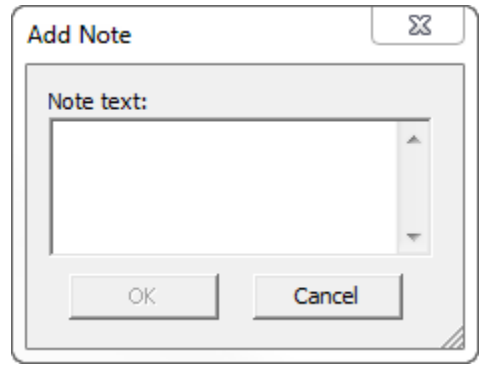
The right-click shortcut menu provides commands and submenus for editing the plot by adding markers, text notes, changing the view, or copying the current image..



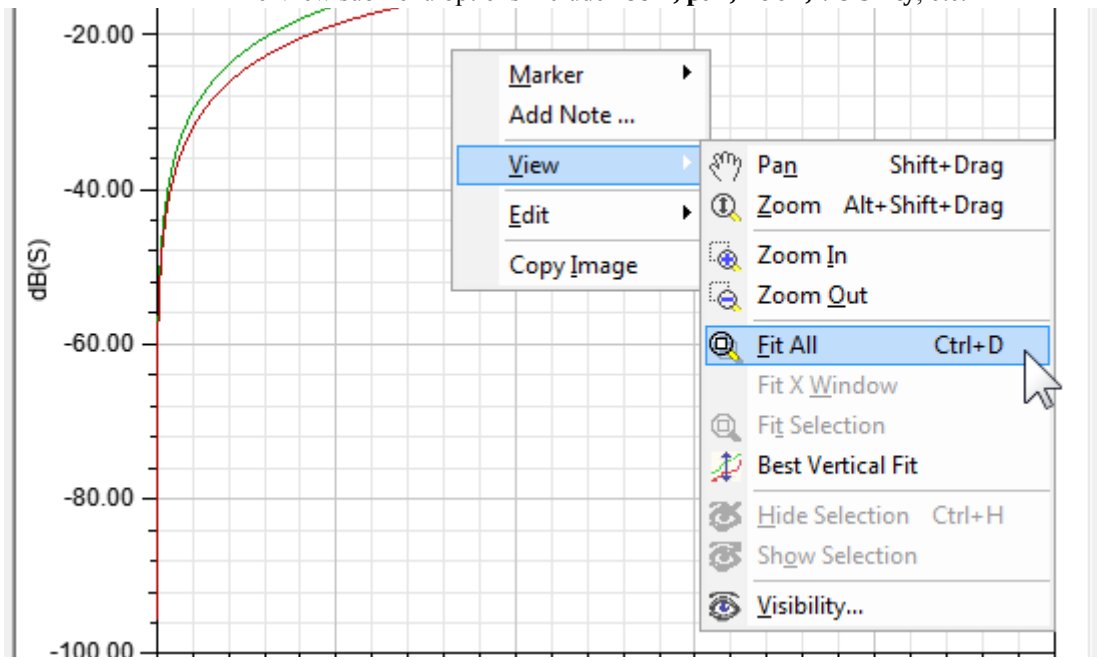
- The Marker submenu options includes the commands for adding markers to plots:



- The Add Note command opens a dialog for adding a text note:

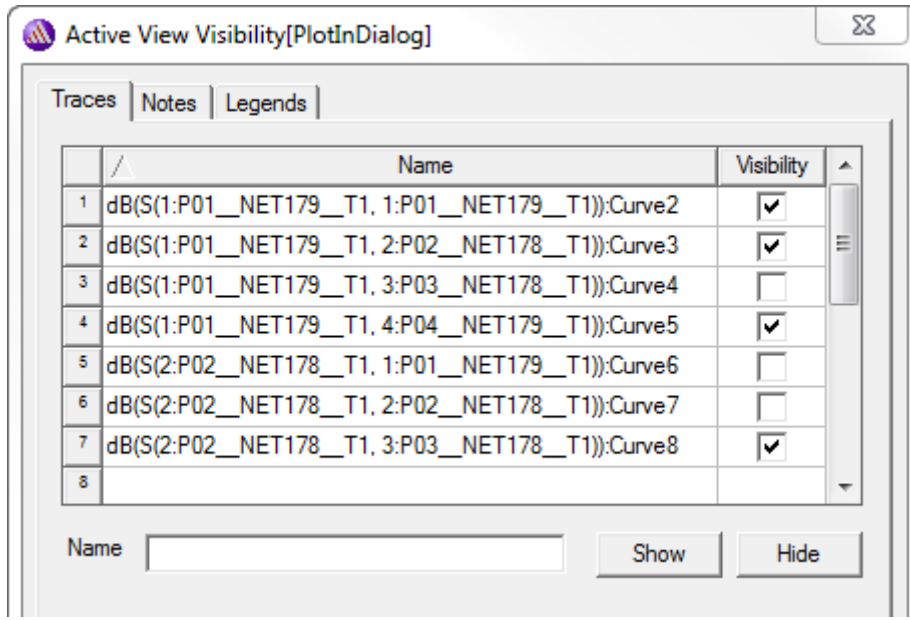


- The View submenu options include **zoom, pan, fit-all, visibility**, etc.

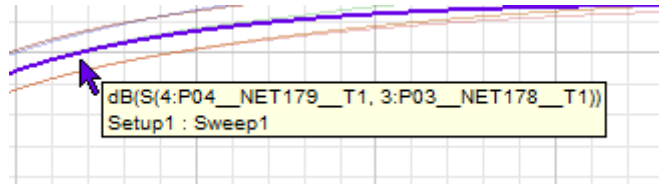


The **Visibility** command lets you select traces, notes, or legends for visibility. You can

also search traces by name and use the **Show** and **Hide** buttons.



- A right click **Copy Image** is also available; the resulting image may be pasted into external documents.
- Holding the cursor over a trace will provide an identifying tool-tip; in the case of multiple variations this is the only way to distinguish between the curves for a particular cell.

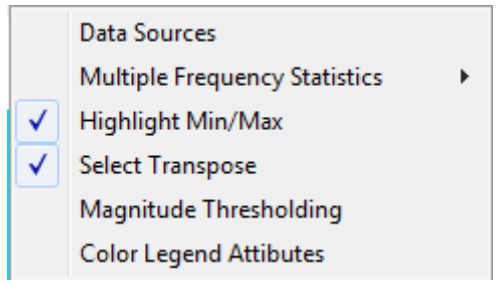


- Selecting **Edit>Properties** opens a Properties dialog for the selected trace or plot object.

Selecting **Table** in the data view pane, switches the view to a tabular view of the trace object.

Data Display Pane: Frequencies and Plot Selected

The shortcut menu provides the following commands.

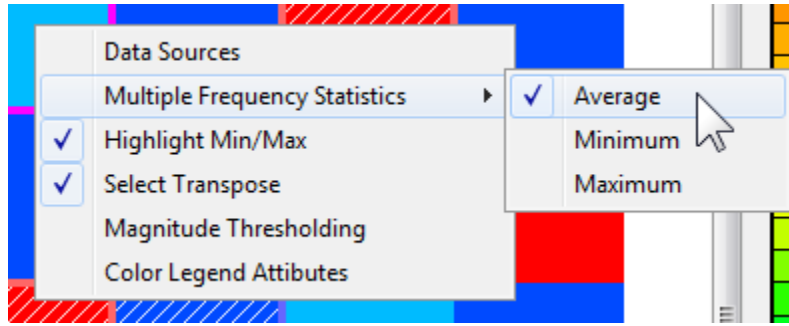


The **Data Sources** command and **Magnitude Thresholding** are the same as on the Edit menu. The other commands are:

- Multiple Frequency Statistics
- Highlight Min/Max
- Select Transpose
- Color Legend Attributes

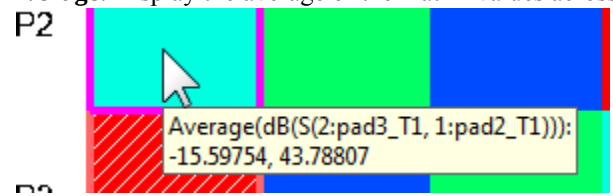
Multiple Frequency Statistics

If you select Frequencies in the Data Control pane, and Plot in the Data View pane, the shortcut menu includes the following commands:



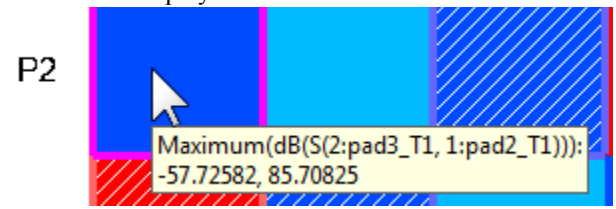
The **Multiple Frequency Statistics** option determines the statistical composite to display when multiple frequencies have been selected for the matrix display. The statistical data is always the first matrix displayed, followed by matrices for each individual frequency. The **Multiple Frequency Statistics** also indicates the data to be used in the colored matrix plot when multiple frequencies have been selected.

- **Average**: Display the average of the matrix values across the selected frequencies.



- **Minimum**: Display the minimum matrix values across the selected frequencies.

- **Maximum**: Display the maximum matrix values across the selected frequencies.



Highlight Min/Max

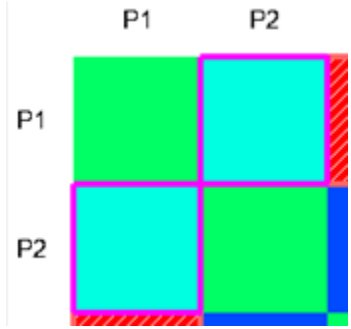
This NdExplorer context menu option determines whether the minimum and maximum matrix entries should be highlighted in the matrix table and color plot view.

Select Transpose

When selecting a cell in the table view of the Frequencies, this NdExplorer **Edit** menu option determines if the transpose cell is to be highlighted as well.

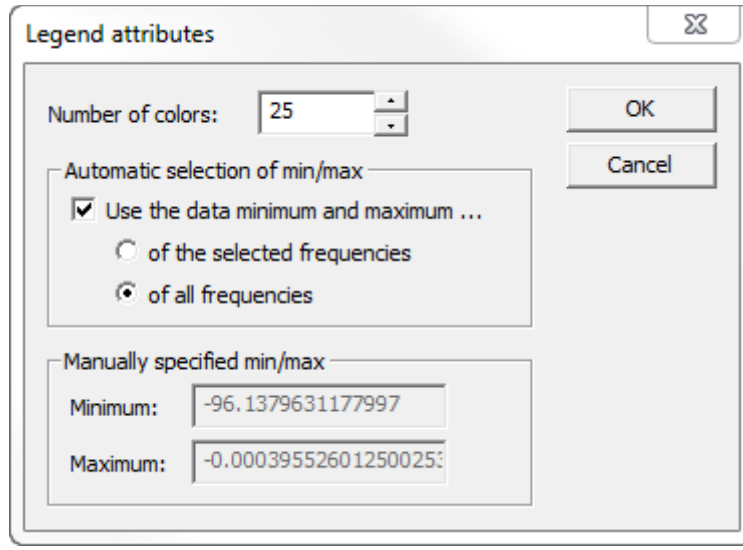
	P4	-96.13785, 180.00000	-0.00040, 0.00000	-95.49173, 0.00000	-87
0.01000GHz	P1	-57.10340, -81.97841	-57.72581, 85.70836	-0.00213, -0.18916	-66
	P2	-57.72582, 85.70825	-57.09556, -81.98749	-66.27113, 99.16916	-0.
	P3	-0.00213, -0.18916	-66.27110, 99.16941	-57.10246, -81.95892	-57
	P4	-66.27112, 99.16942	-0.00213, -0.18929	-57.72556, 85.71465	-57.

This feature also works for the plot view of the frequencies



Changing the Color Scheme for a Matrix Color Plot

Double clicking the matrix color plot legend or right clicking within the plot itself and choosing **Color Legend Attributes**, displays the **Legend Attributes** dialog. The dialog permits you to change the granularity of the color scheme and the value range but not the colors themselves. Using a standard range across all frequencies permits you to quantitatively compare plots; NdExplorer remembers legend settings for each data-type/display-format pair.



- **Number of colors**: number of color entries in the legend, i.e. the number of divisions between the start/end of the data range.
- **Use the data minimum and maximum**: automatically select the data range using the minimum and maximum values from either just the **selected frequencies** or **all frequencies** in the data set.

- **Minimum & Maximum**: when the range is not automatically determined, these fields permit the user to manually enter hard values, e.g. for S parameter data magnitude data, a minimum of 0 and a maximum or 1 could be used.

Exploring Network Data and Modifying the Display

The Network Data Explorer allows you to view data and modify various aspects of the display, including color plots, color coding, viewing across frequencies, and displaying individual statistics.

This section provides examples of various ways of using NdExplorer.

- Viewing the S, Y, or Z Matrix for a Selected Frequency
- Viewing a Color Coded Matrix Plot
- Viewing Matrix Cell Data Across All Frequencies
- Displaying a Graph of a Cell Across All Frequencies
- Displaying Statistics by Frequency
- Displaying Individual Statistics for All Frequencies
- Creating a Statistics Plot
- Comparing Network Data
- Comparing Variations
- Displaying Plot Traces from Multiple Data Sources

Related Topics

Magnitude Thresholding

Smooth All Frequencies

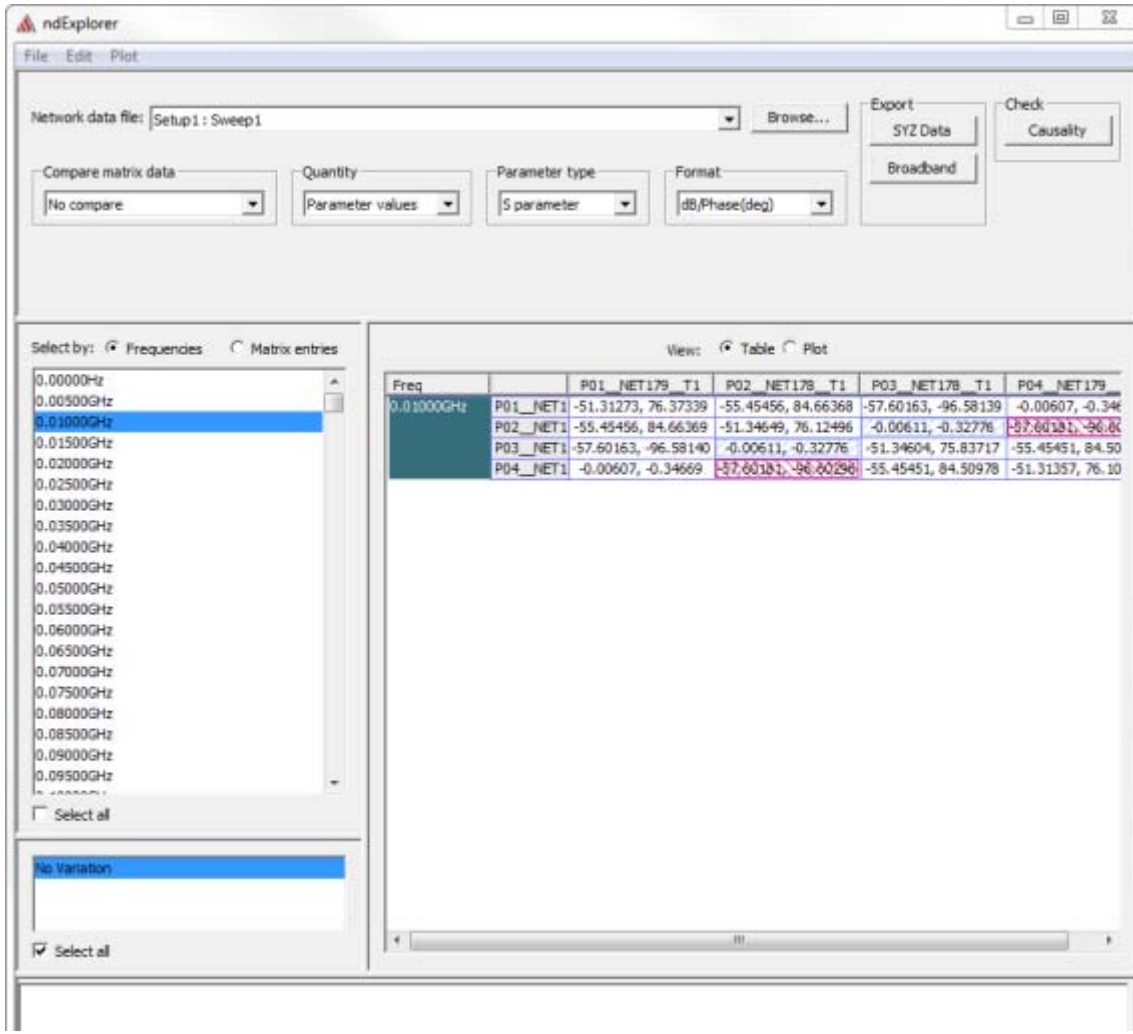
Cell Filtering

Edit Port Properties

Display Differential Pairs as Post Process Selection

Viewing the S, Y, or Z Matrix for a Selected Frequency

1. Select **Parameter values** in the **Quantity** control field.
2. Select **Frequencies** in the data selection pane at left.
3. Select **Table** in the data view pane at right.
4. Choose the frequencies and the variations to display.



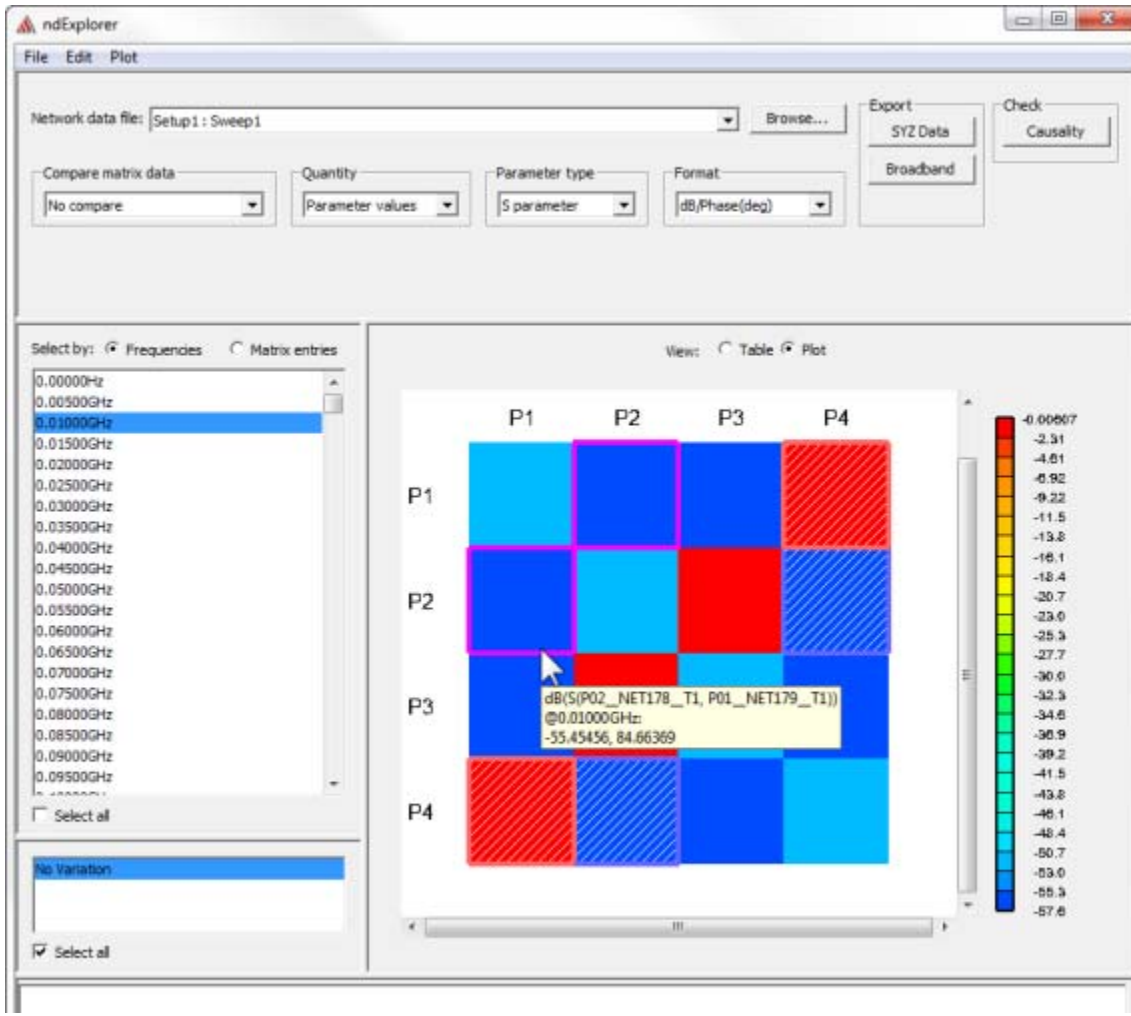
- The maximum value (and its transpose) is highlighted in red and the minimum value in blue. Complex values (regardless of the data format) are compared using their modulus. When multiple frequencies or variations are selected, the first in the data display is either the average, minimum, or maximum of all the frequencies selected — the selection type is set via the right-click menu option, **Multiple Frequency Statistics**.
- Use **Parameter type** in the control panel to choose the data type (S, Y, or Z) and **Format** to specify how the data values are to be displayed (real, imaginary, dB, etc.).
- When the cursor is placed over a cell, tool-tip text reveals detail as to the content of the cell. When multiple variations are selected, use the tooltips to identify the variation being displayed. Clicking in a cell will select it and its transpose. Double clicking a cell takes the

user to a matrix cell view in which values for all frequencies for that particular cell are displayed; the double-clicked frequency is highlighted with solid red shading.

- Selecting **Plot** in the data view pane switches the view to a colored graphical representation of the tabular data.

Color Coded Matrix Plot

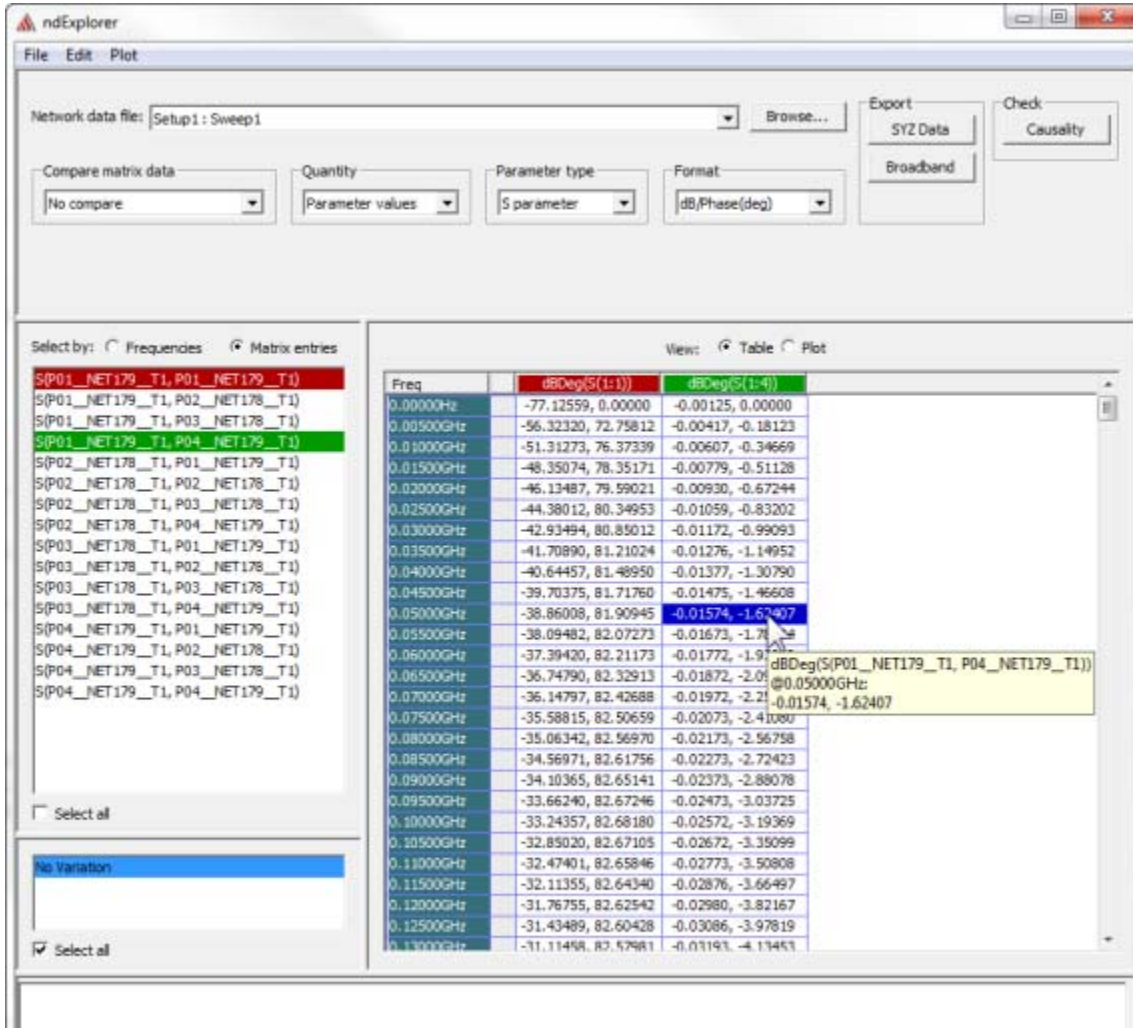
1. Select **Parameter values** in the **Quantity** control field.
2. Select **Frequencies** in the data selection pane at left.
3. Select **Plot** in the data view pane at right.
4. Choose the frequency and variation to display.



- The matrix values are displayed in a color coded grid; if the current **Format** is a complex value, only the real component is used to determine the display color. Use the **Parameter type** selection in the control panel to choose the data type: S, Y, or Z parameter. When multiple frequencies or variations are selected, either the minimum, maximum, or average value is displayed (as chosen with the right-click menu option, **Multiple Frequency Statistics**).
- The **maximum value** (and its transpose) is highlighted in red and the **minimum value** in blue; the real value alone is used to determine these extremes.
- When the cursor is placed over a cell, tool-tip text reveals detail as to the content of the cell. Clicking in a cell will select it and its transpose. Double clicking a cell takes the user to the matrix cell plot in which all frequency values for that matrix cell are displayed as a graph.
- Double clicking the legend or choosing the right-click menu option **Color Legend Attributes** allows you to select the data range and number of colors for each data type and format. See Changing the Color Scheme for a Matrix Color Plot.
- Selecting **Table** in the data view pane, switches the view to a matrix representation of the graphical data.

Viewing Matrix Cell Data Across All Frequencies

1. Select **Parameter values** in the **Quantity** control field.
2. Select **Matrix entries** in the data selection pane at left.
3. Select **Table** in the data view pane at right.
4. Choose the frequency and variation to display.

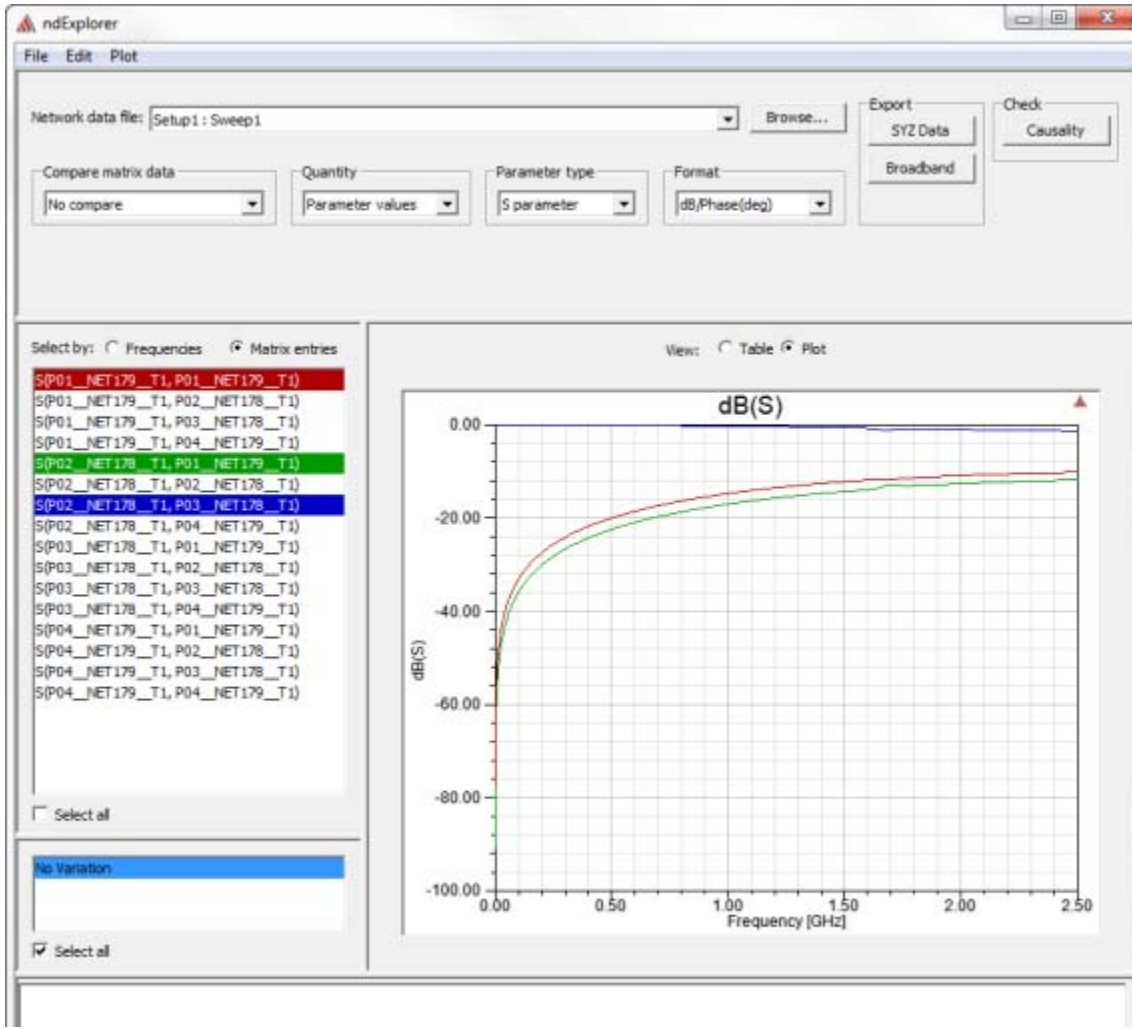


- Values for all frequencies are displayed for each matrix cell selected. The data type and format are determined by the **Parameter type** and the **Format** settings in the control pane.
- When the cursor is placed over a table cell, tool-tip text reveals more detail as to the content of the cell. Clicking in a cell will select it. Double clicking a cell takes the user to the matrix table view in which all cell values for the corresponding frequency are displayed; the double-clicked cell is highlighted with solid red shading.
- Clicking on a column header in the data table highlights the column. The corresponding trace in the plot view is also highlighted.
- Selecting **Plot** in the data view pane, switches the view to a graph of the cell data. The

graph shares the same color coding as the columns; highlighted columns are highlighted in the plot.

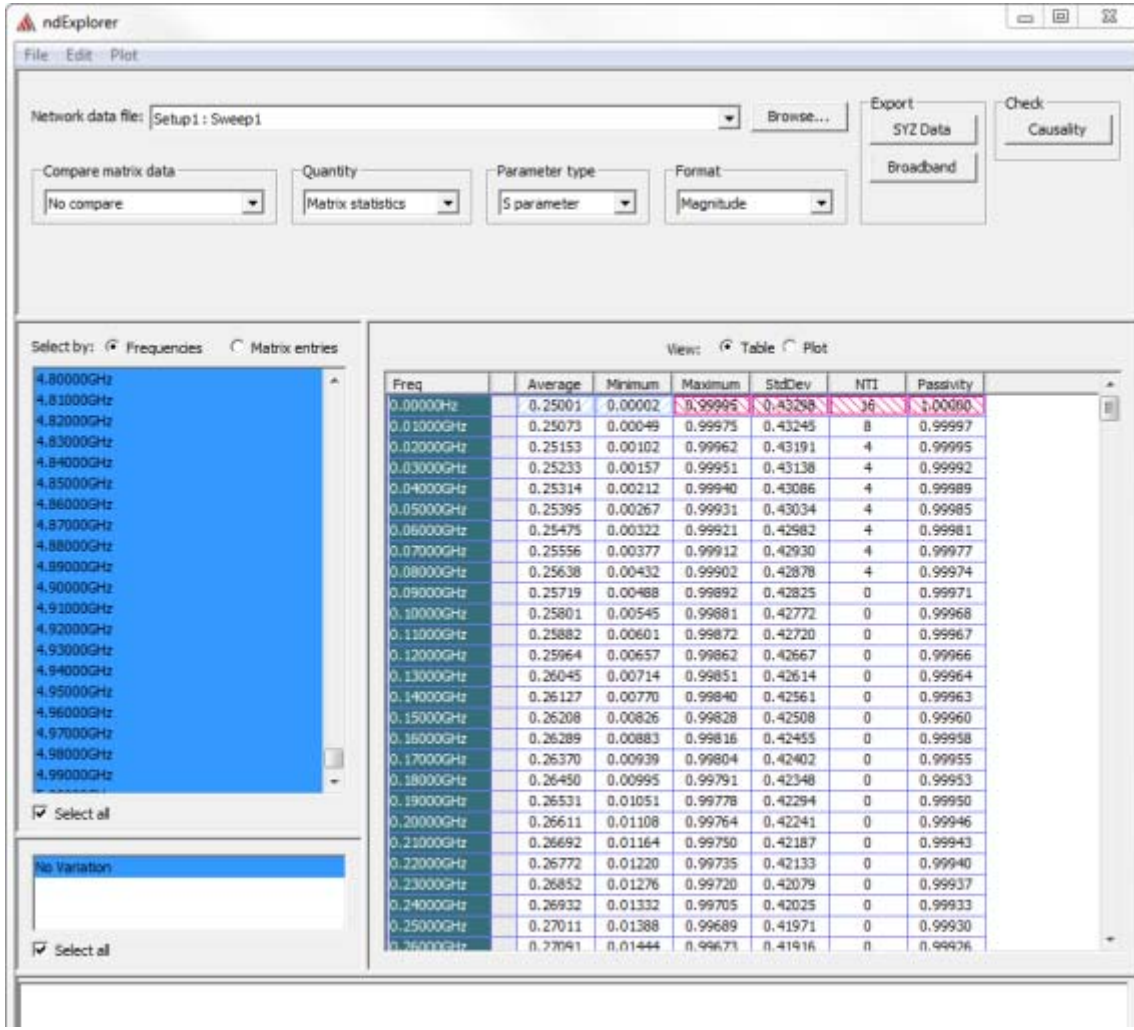
Displaying a Graph of a Cell Across All Frequencies

1. Select **Parameter values** in the **Quantity** control field.
2. Select **Matrix entries** in the data selection pane at left.
3. Select **Plot** in the data view pane at right.
4. Choose the cells and variations to display.
5. Select **Plot > Log Frequency** to switch the X axis to a log scale.



Displaying Statistics by Frequency

1. Select **Matrix statistics** in the **Quantity** control field.
2. Select **Frequencies** in the data selection pane at left.
3. Select **Table** in the data view pane at right.
4. Choose the frequencies and variations to display.



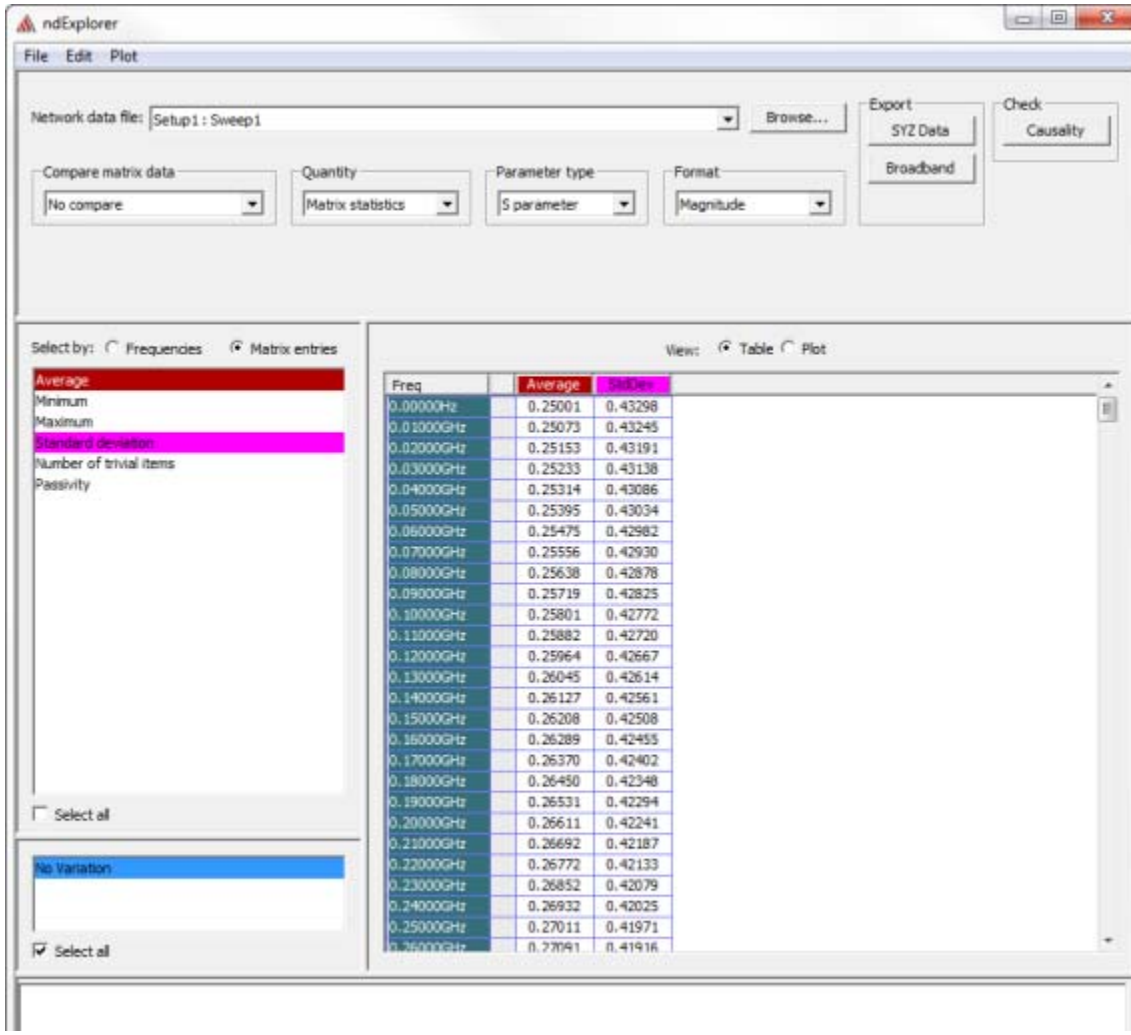
- The various statistical measures for the current **Parameter type** and **Format** are displayed for each frequency selected. Only real (not complex) data formats are offered for statistical analysis. **Passivity** is only available for S-parameter data (comparisons inactive). **NTI** refers to the number of trivial items; for S-parameters, this includes all zeros

and ones; for all other data (and data comparisons), only zeros are counted as trivial. The minimum value for each column is highlighted in blue; the maximum is highlighted with red.

- When the cursor is placed over a cell, tool-tip text indicates the frequency and statistics displayed. Clicking a cell selects the cell. Multiple variations are displayed as separate entries in the table; use the tool-tip to identify the variation for a particular frequency.
- Clicking a column header sorts the data using that column for comparison; clicking again reverses the order of the sort.

Displaying Individual Statistics for All Frequencies

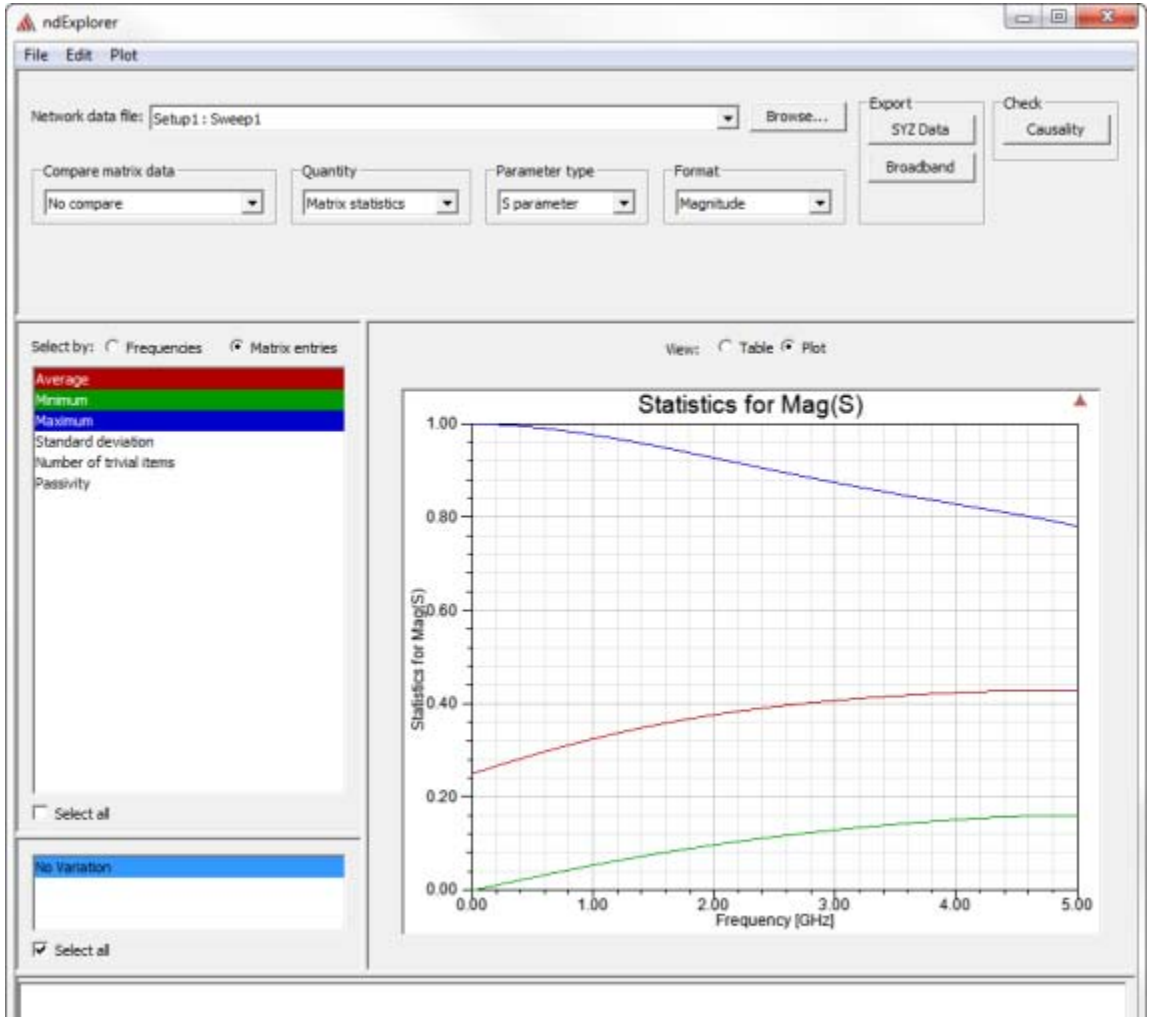
1. Select **Matrix statistics** in the **Quantity** control field.
2. Select **Matrix entries** in the data selection pane at left.
3. Select **Table** in the data view pane at right.
4. Choose the statistics and variations to display.



- The various statistical measures (for the current **Parameter type** and **Format**) are displayed for all frequencies. **Passivity** is only available for S-parameters (comparisons inactive). **NTI** refers to the number of trivial items; for S-parameters, this includes all zeros and ones; for all other data (and data comparisons), only zeros are counted as trivial.
- When the cursor is placed over a cell, tool-tip text indicates the frequency and statistics displayed. Clicking a cell selects the cell. Multiple variations are displayed as separate entries in the table; use the tool-tip to identify the variation for a particular frequency.
- Selecting **Plot** in the data view pane, switches the view to a graph of the statistical data. The graph shares the same color coding as the columns; highlighted columns are highlighted in the plot.

Creating a Statistics Plot

1. Select **Matrix statistics** in the **Quantity** control field.
2. Select **Matrix entries** in the data selection pane at left.
3. Select **Plot** in the data view pane at right.
4. Choose the statistics and variations to display.



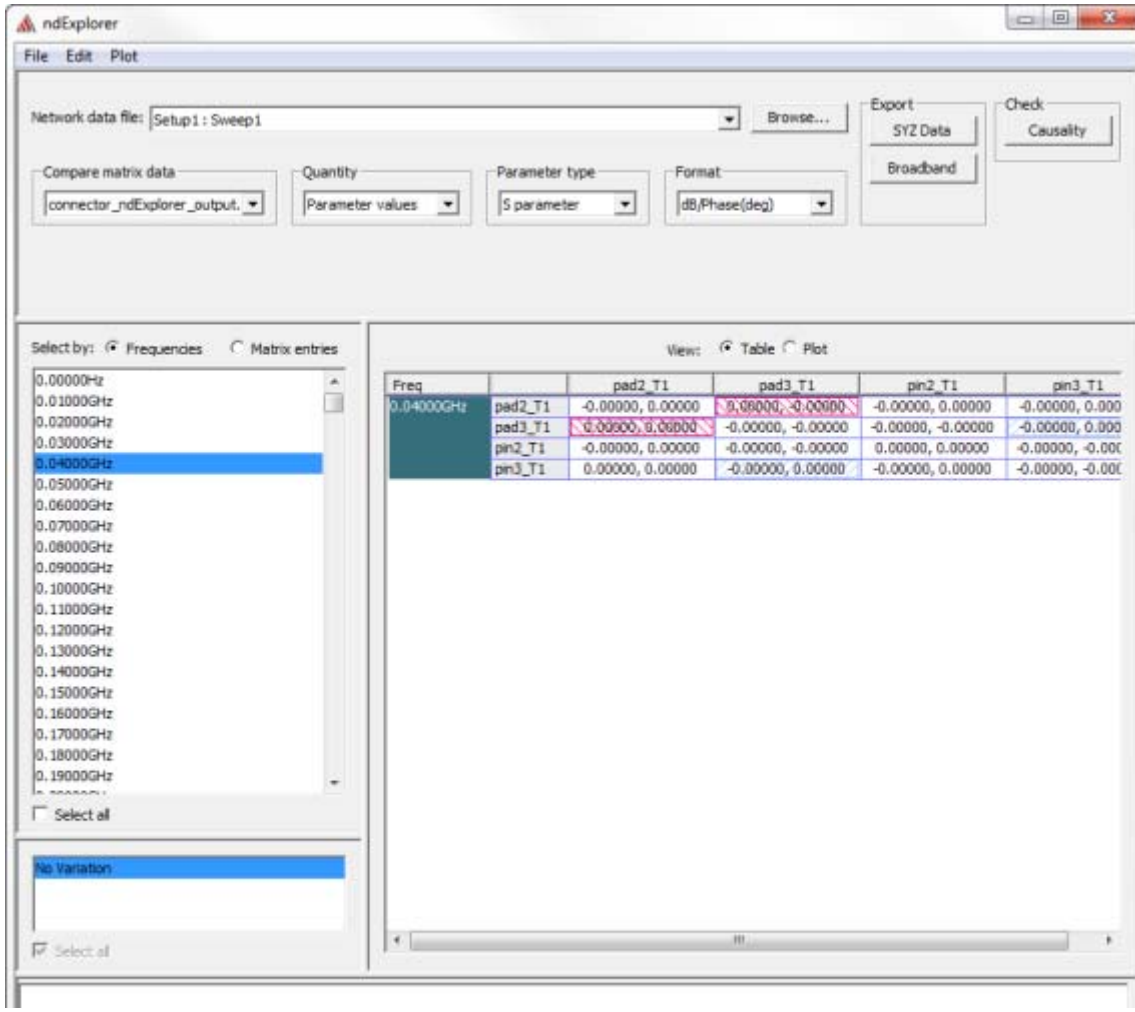
- A graph of each of the statistical measures selected (for the current **Parameter type** and **Format**) is displayed across all frequencies. **Passivity** is only available for S-parameters (comparisons inactive). **NTI** refers to the number of trivial items; for S-parameters, this includes all zeros and ones; for all other data (and data comparisons), only zeros are

counted as trivial.

- When the cursor is placed over a curve, tool-tip text indicates the statistical function, the data format, the data type, and the variation of the graph being displayed.
- Selecting **Table** in the data view pane, switches the view to a tabular presentation of the statistical data. The table columns share the same color coding as the plot traces.

Comparing Network Data

1. Select **Browse** in the **Compare matrix data** control field or the **Compare** menu option on the **File** menu.
2. Load a network data file to be compared with the currently active data. Multiple files may be opened from the file browser dialog, however, only one file is compared against the primary data set at any given time.

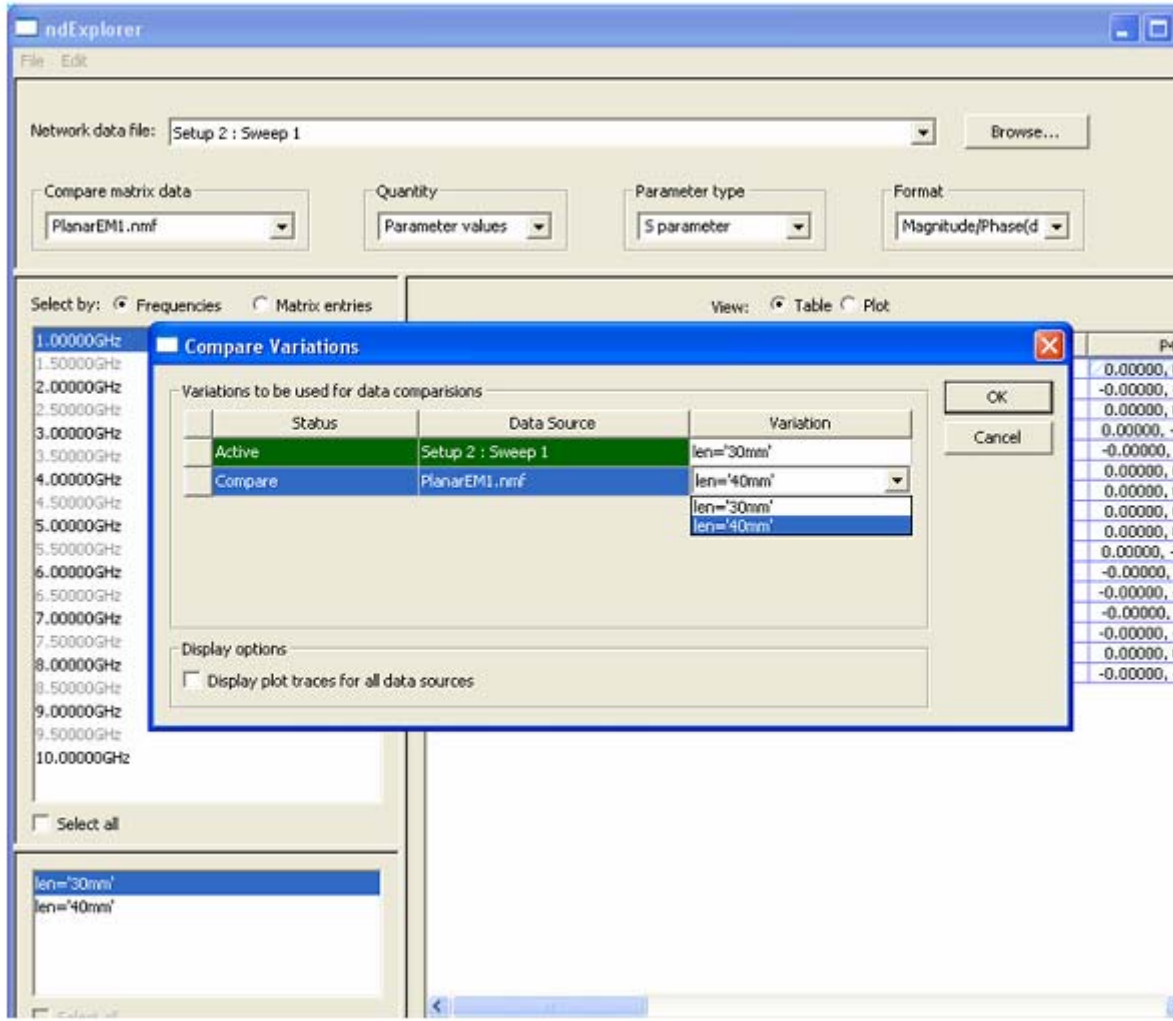


- The data displayed in all views is now of the primary Network data less the values for the compare data set. The data format (magnitude, dB, etc.) is applied either before (pre) or after (post) the subtraction depending on the corresponding setting in the **Display format settings** dialog (available on the **Edit** menu or the right-click context menu).
- Multiple data-sets can be loaded for comparison though only one is active at any one time. The compare data set may be changed by using the **Compare matrix data** control field; comparisons may also be turned off by selecting **No compare**.
- The primary data source may be changed using the **Network data file** drop-down list within the control pane. By so doing, a comparison data set may be made the primary data set, either for viewing as is, or for comparing with other network data.

- The **Compare matrix data** dialog, accessed through the **Select Compare Variations** menu option, may also be used to switch comparison data sets. When a comparison is active, the data sets are restricted to one variation. The variation selected in the **variation list** box applies to the primary data source alone, i.e. the one displayed in the **Network data file** field of the **control** pane.

Comparing Variations

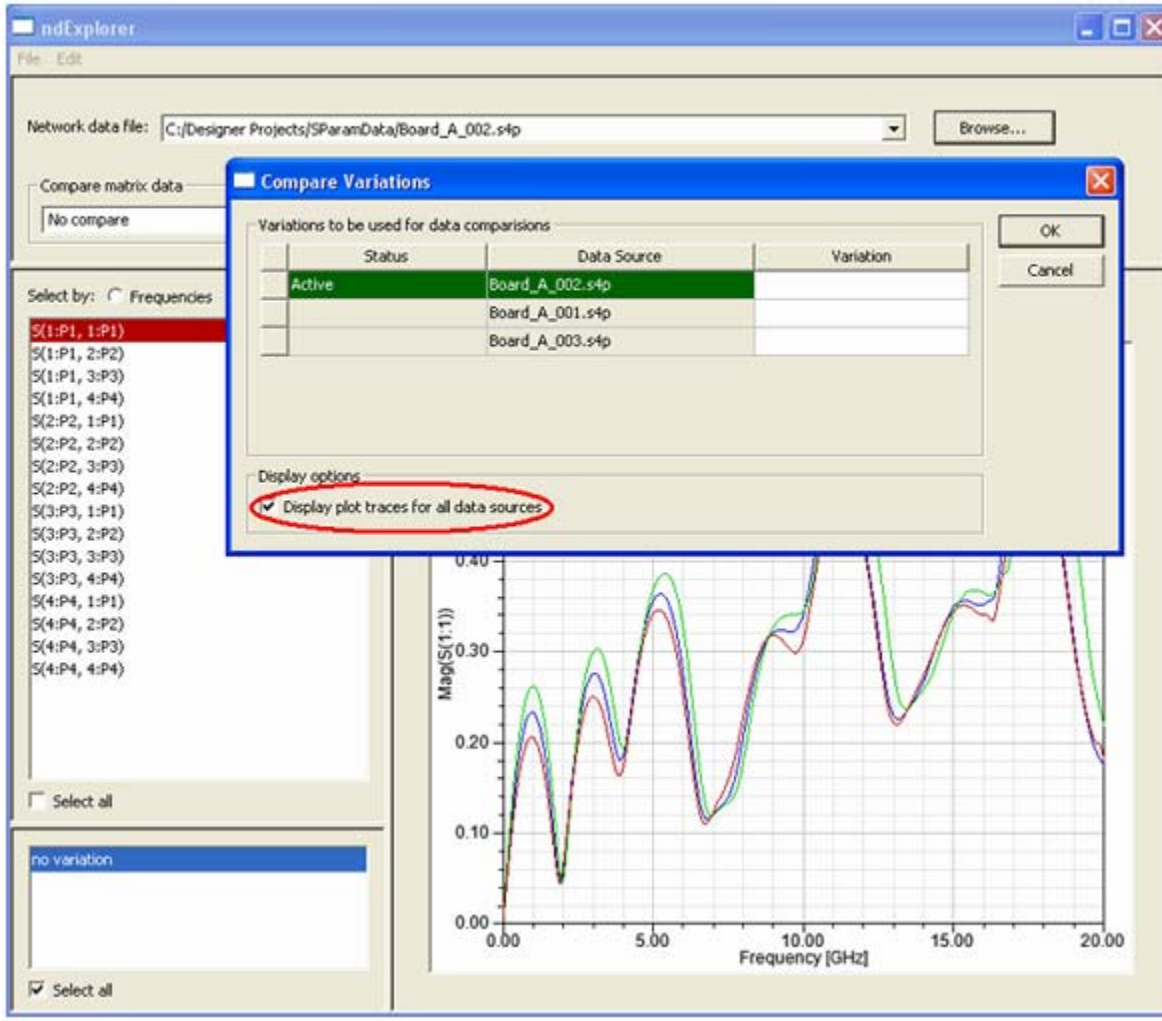
1. Select **Browse** in the **Compare matrix data** control field or the **Compare** menu option on the **File** menu.
2. Load the data for the same (or different) network data file. Please note, that currently only ".nmf" files support multiple variations.
3. Use **Edit>Data Sources** (**Edit** menu or the right-click menu, table view) to choose the variations to compare.



- The **Compare Variations** dialog displays a table of all the loaded data. Selecting a row changes the comparison data set; the data view is updated when the dialog is dismissed with the **OK** button. Note: it is not possible to compare a data set against itself, *unless the data set has been loaded twice*. Selecting the active data set row clears the comparison.
- The variations being used may be changed through the drop-down lists in the **Variation** column. Please note that when comparisons are inactive, all the variations for the active data source may be displayed (where relevant) in the data view; when comparisons are active, only the one selected variation is displayed.

Displaying Plot Traces from Multiple Data Sources

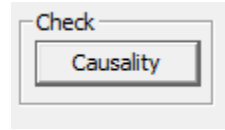
1. Open multiple data sets, either through the main file browser or through the data comparison browser.
2. Use the **Edit>Data Sources** menu or the right-click menu to open the **Select Compare Variations** dialog.
3. Enable the **Display plot traces for all data sources** option.



- Traces for a given cell or statistical measure are displayed for all data sets; you can use tool tips to distinguish between them.
- If a single cell or statistical measure is displayed, different colors are used for each data trace.
- If multiple cells or statistical measures are selected, a single color is used for all data traces for each cell or statistical measure.
- If a data comparison is active, traces will be shown for all data sets compared against it (with the exception of the comparison data set). This is true for both cell and statistical traces.
- Data sets with no variation information are always displayed. With multiple variation data sets, only those with values for the currently selected variations are displayed. (Note that the variation for the comparison data set is fixed by the selection that is made in the **Select Compare Variations** dialog.)

Causality Checking and Plots

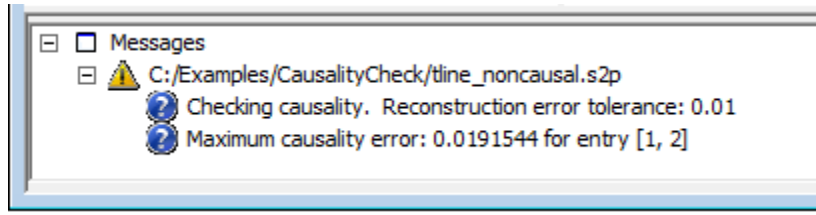
The NdExplorer can perform a causality check on S-parameter data from any source (solution or file), and provide plots of the results in various formats..



When S-parameter data are loaded into NdExplorer, the Check Causality button is enabled.

Click on the button to start the causality check. Depending on the size of the S-parameter data, the causality check may take several minutes or longer to complete.

When the causality check is complete, a summary of the results appears in the Message window.

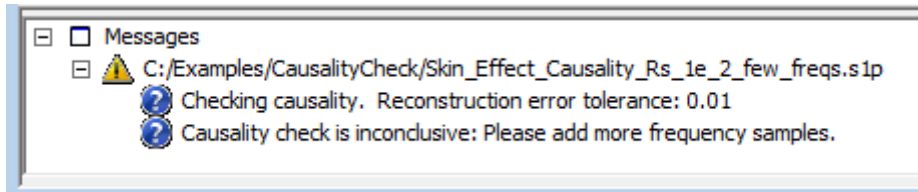


Causality of a frequency response is determined by calculating the generalized Hilbert transform of the data at all frequencies. A causal frequency response is equal to its generalized Hilbert transform. The reconstruction error is the difference between the tabulated data and its transform at a given frequency. The message shows the maximum reconstruction error tolerance for a causal frequency response. The default tolerance of 0.01 is equal to the state-space fitting tolerance.

The second line reports the maximum causality error for all port pairs and all frequencies, along with the matrix indices (port numbers) where the maximum noncausality occurs. A noncausal response is one where all matrix entries can be conclusively analyzed, and at least one entry exceeds the causality tolerance. In the message shown above, the Touchstone file **tline_noncausal.s2p** has a maximum reconstruction error of 0.0191544 in the response involving ports 1 and 2.

When all results are conclusive but no matrix reconstruction error exceeds the tolerance, the maximum causality error is reported as zero, and no matrix entry is listed.

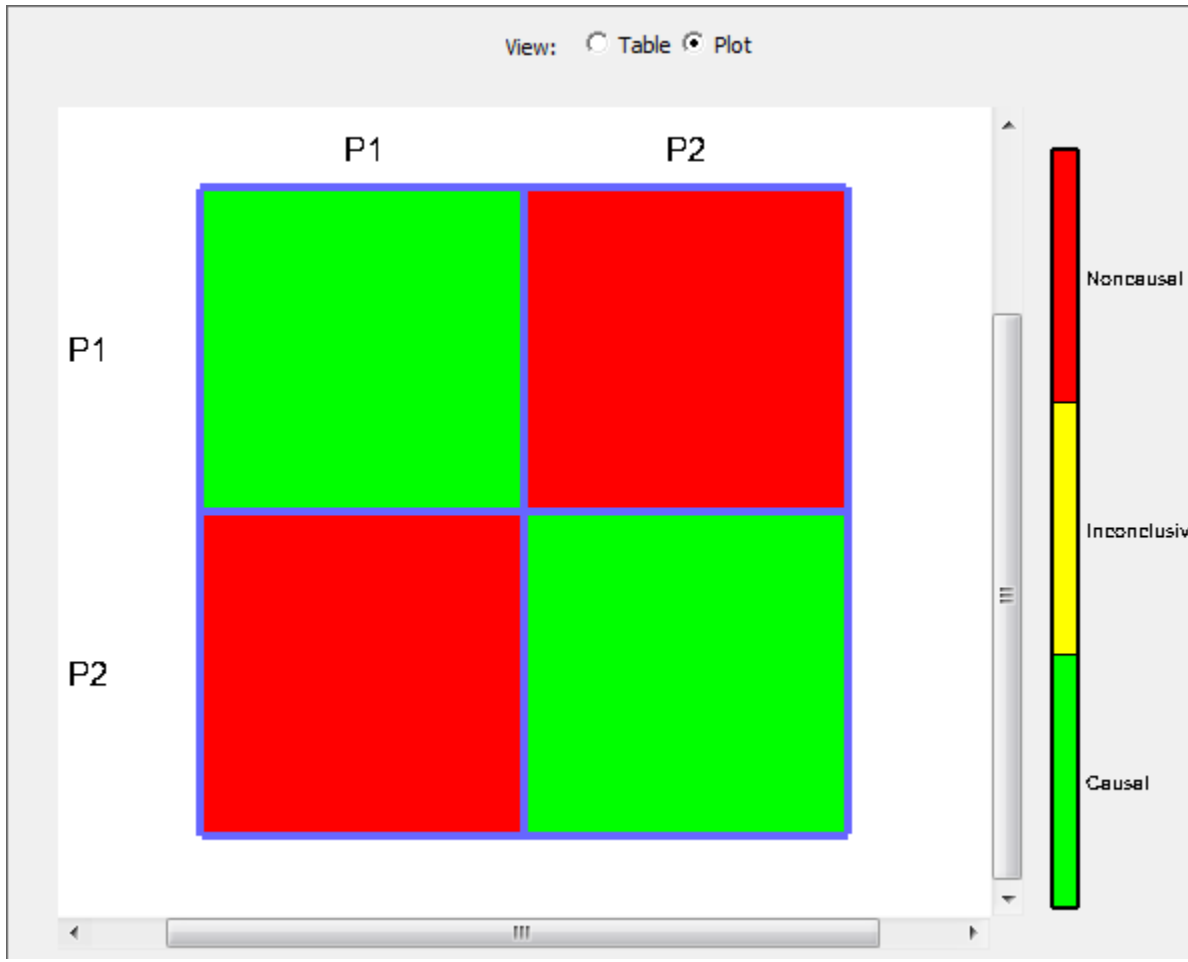
If the data does not contain enough frequency points to determine whether the data is or is not causal, the message will indicate the inconclusive result.



NdExplorer will report the data set as inconclusive if any cells are inconclusive, even if other entries exhibit causality violations.

To see the results of the causality check, click in the **Quantity** field and select **Causality plots**.

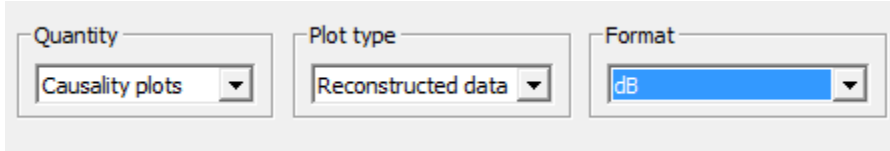
To view a summary diagram of the causality results, set the **Plot Type** to **Causality violation**. The default mode is a rectangular **Plot** of dimension N x N, with color-coding to indicate the causality status of each port pairing:



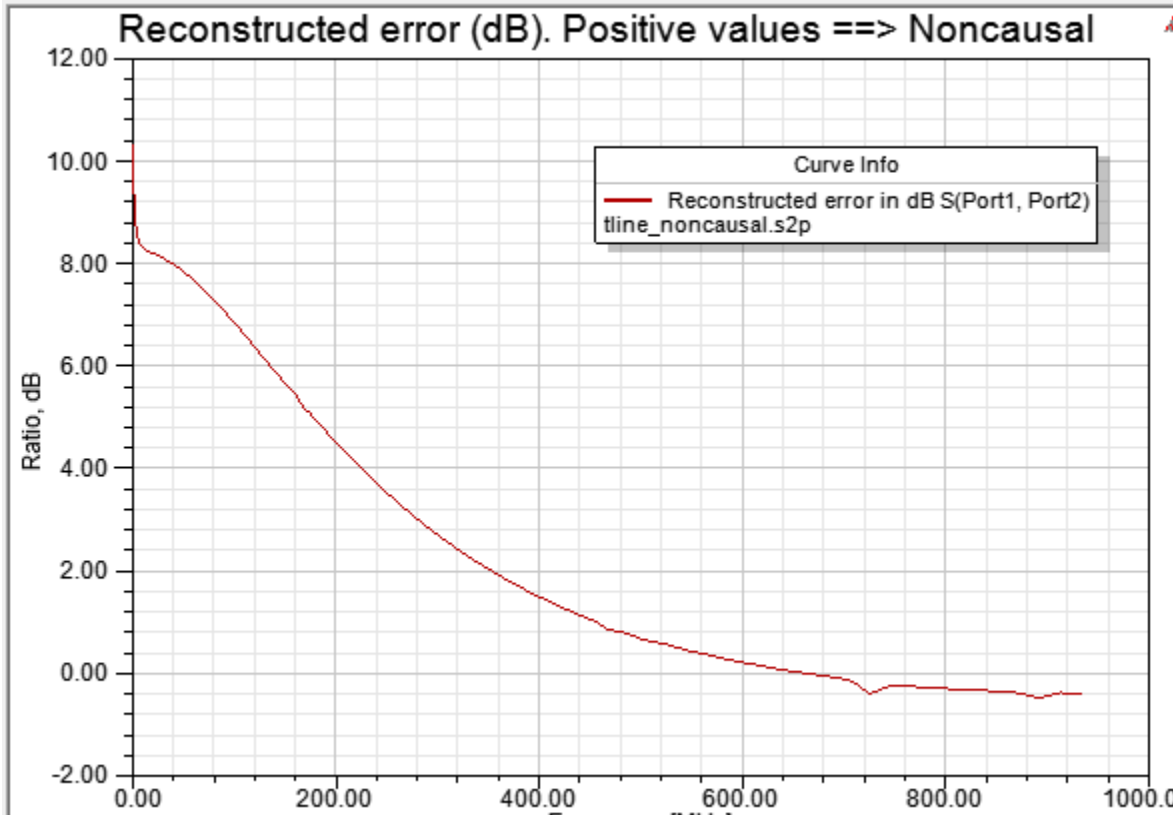
In this plot, the cells go from (Port 1, Port 1) at the upper left to (Port N, Port N) at the lower right. The result shows the causality over all frequencies in the data. In this example, the matrix is symmetric, so that both S12 and S21 are noncausal, while S11 and S22 are causal.

To see the details for each frequency, click **Table** for the **View** instead of **Plot**.

To plot the reconstructed frequency response generated by the causality checker, leave the **Quantity** at **Causality plots** and set the **Plot type** to **Reconstructed data**. The **Format** field appears:

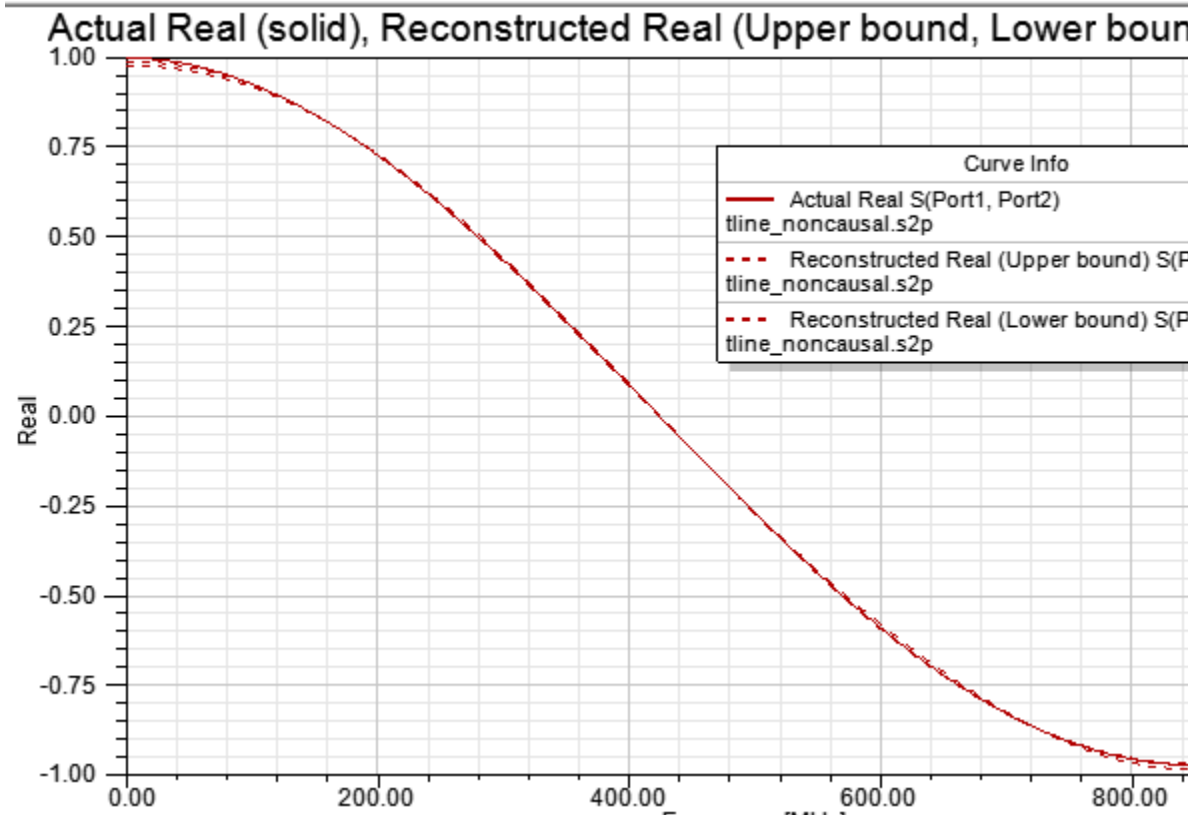


The default **Format** is **dB**. This plots the reconstruction error at each frequency divided by the tolerance, 0.01 in this example. Here is the plot for the noncausal S12 data identified in the Causality plot shown earlier.



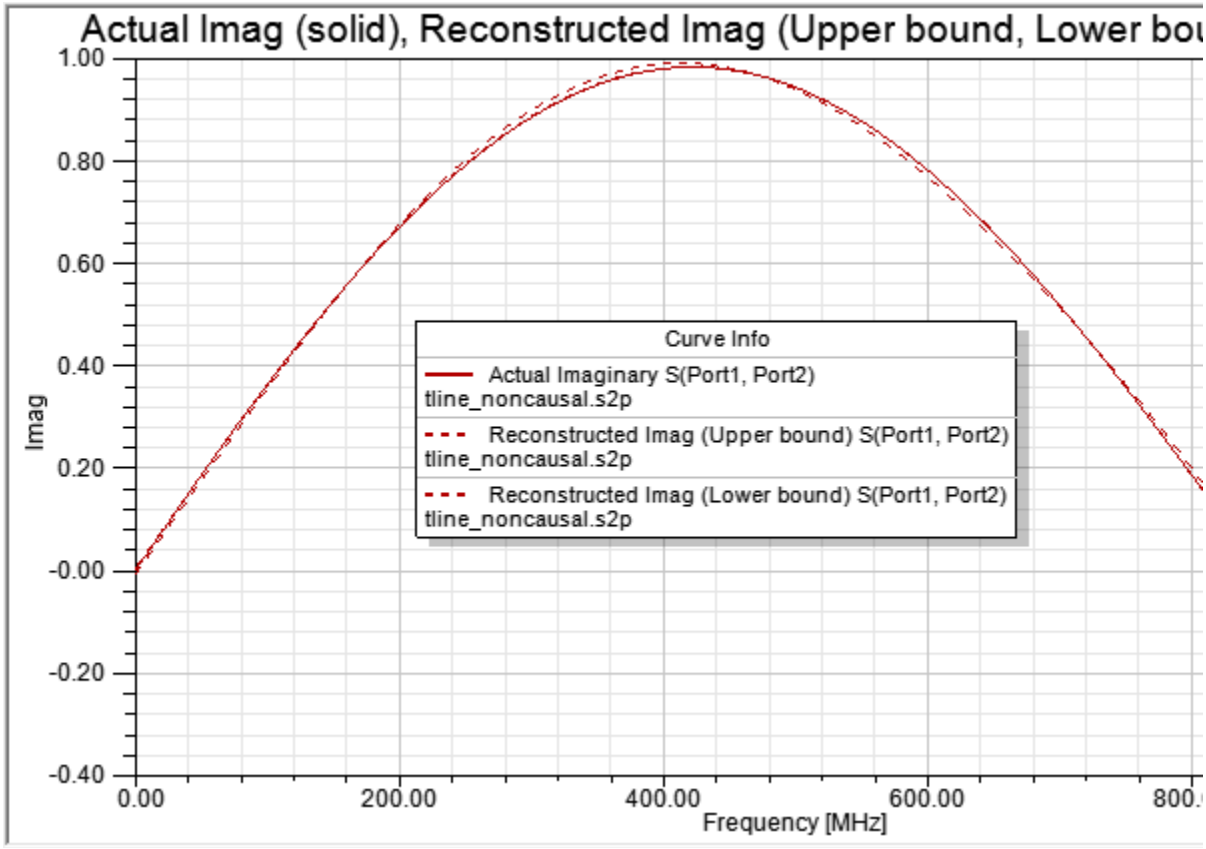
The reconstruction error ratio for parameter S12 is positive for frequencies less than about 680MHz, indicating a broad range of noncausal behavior.

To compare the real part of the reconstructed data to the real part of the actual data, set the **Format** to **Real**.



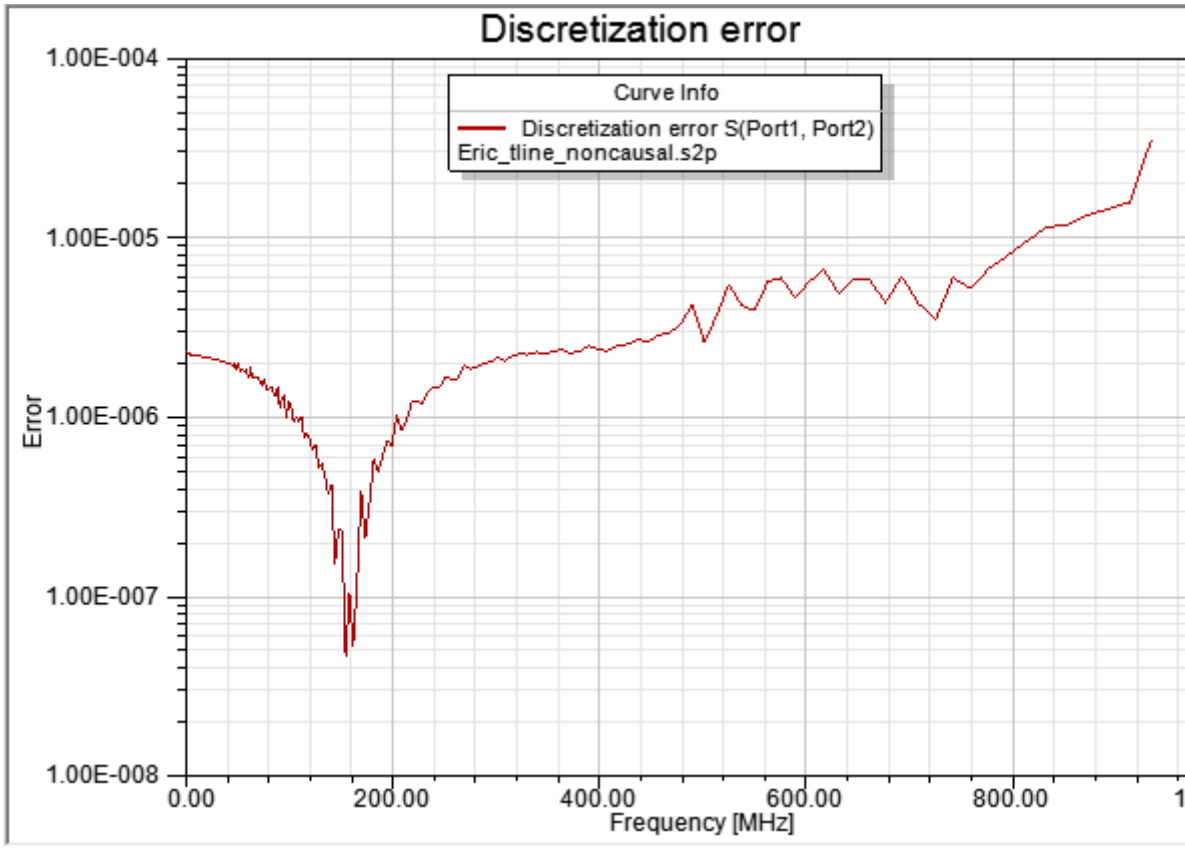
For a causal frequency response, the actual data (solid line) will be within the upper and lower bounds of the reconstruction (dotted lines) at all frequencies.

To compare the imaginary part of the reconstructed data to the imaginary part of the actual data, set the **Format** to **Imaginary**.



For a causal frequency response, the actual data (solid line) will be within the upper and lower bounds of the reconstruction (dotted lines) at all frequencies.

To view the frequency-dependent discretization error, set the **Format** to **Discretization**.



The discretization error is the error that is due to the fact that the data are available only at discrete frequencies rather than for a continuous spectrum. A discretization error near or greater than the causality tolerance renders the causality check inconclusive. Data at more frequencies could reduce the discretization error and render the analysis conclusive. This set of data exhibits low discretization errors (<<0.01) at all frequencies, and the causality check is conclusive (conclusively non-causal in this example).

19

Technical Notes

The simulation technique used to calculate the full 3D electromagnetic field inside a structure is based on the finite element method. Although its implementation is largely transparent, a general understanding of the method is useful in making the most effective use of HFSS.

The HFSS Technical Notes provide an overview of the finite element method and its implementation in HFSS. They also describe how modal S-parameters are computed from the simulated electric and magnetic fields and how they can be converted to "nodal" or "voltage" based pseudo-S-parameters used in circuit theory.

Information is included on the following:

The Finite Element Method

In order to generate an electromagnetic field solution, HFSS employs the finite element method. In general, the finite element method divides the full problem space into thousands of smaller regions and represents the field in each sub-region (element) with a local function.

In HFSS, the geometric model is automatically divided into a large number of tetrahedra, where a single tetrahedron is a four-sided pyramid. This collection of tetrahedra is referred to as the finite element mesh.

Representation of a Field Quantity

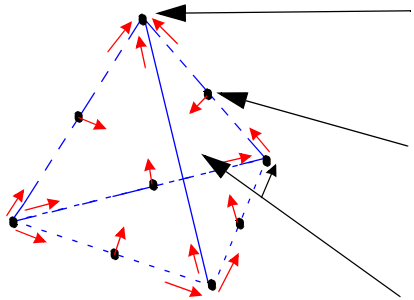
Rectilinear Elements and Curvilinear Elements

Basis Functions

Size of Mesh Vs. Accuracy

Representation of a Field Quantity

The value of a vector field quantity (such as the H-field or E-field) at points inside each tetrahedron is interpolated from the vertices of the tetrahedron. At each vertex, HFSS stores the components of the field that are tangential to the three edges of the tetrahedron. In addition, HFSS can store the component of the vector field at the midpoint of selected edges that is tangential to a face and normal to the edge (as shown below). The field inside each tetrahedron is interpolated from these nodal values.



The components of a field that are tangential to the edges of an element are explicitly stored at the vertices.

The component of a field that is tangential to the face of an element and normal to an edge is explicitly stored at the midpoint of selected edges.

The value of a vector field at an interior point is interpolated from the nodal values.

By representing field quantities in this way, the system can transform Maxwell's equations into matrix equations that are solved using traditional numerical methods.

Related Topics

The Finite Element Method

Rectilinear Elements and Curvilinear Elements

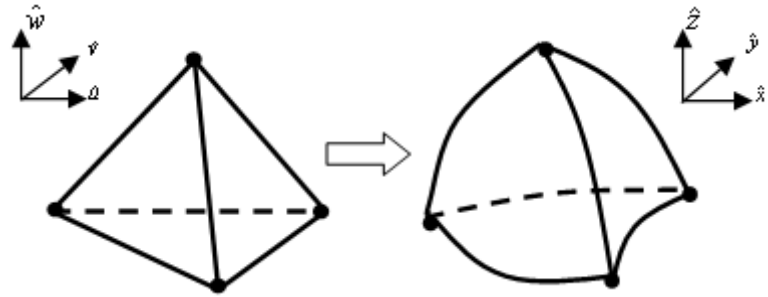
Basis Functions

Size of Mesh Vs. Accuracy

Rectilinear Elements and Curvilinear Elements

In the Finite Element Method (FEM), the physical domain is subdivided into many small elements. In the global coordinates (x, y, z), the field quantities to be solved are expanded by a set of basis functions. Since the basis functions are associated with each element, it is more convenient to write them in the local coordinates (u, v, w), which are independent of the shape of the element. The local coordinates and the global coordinates are linked by a transformation through a so-called Jocobian.

$$J = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \\ \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} & \frac{\partial z}{\partial v} \\ \frac{\partial x}{\partial w} & \frac{\partial y}{\partial w} & \frac{\partial z}{\partial w} \end{bmatrix}$$



If the transformation is linear, the elements are called rectilinear elements, with straight edges and planar faces; if the transformation is non-linear, such as quadratic or cubic, the elements are called curvilinear elements, with curved edges and/or curved faces. So the curvilinear elements are a more general type of elements than their rectilinear counterparts. While they may not always conform exactly to the curved boundaries, the curvilinear elements conform better with less number of elements than the rectilinear ones. As a result, they generate more accurate and usually faster FEM solutions.

Related Topics

The Finite Element Method

Representation of a Field Quantity

Basis Functions

Size of Mesh Vs. Accuracy

Basis Functions

Various interpolation schemes, or basis functions, can be used to interpolate field values from nodal values.

- A first order tangential element basis function interpolates field values from both nodal values at vertices and on edges.

  First order tangential elements have 20 unknowns per tetrahedron.

- A zero order basis function makes use of nodal values at vertices only — and therefore assumes that the field varies linearly inside each tetrahedron.

  Zero order tangential elements have six unknowns per tetrahedron.

- A second order tangential element interpolates field values from nodal values at vertices, on edges and on faces.

  Second order tangential elements have 45 unknowns per tetrahedron.

- Mixed order assigns basis function elements based on the need for higher accuracy in different parts of the model.

Related Topics

The Finite Element Method

Representation of a Field Quantity

Rectilinear Elements and Curvilinear Elements

Size of Mesh Vs. Accuracy

Size of Mesh Vs. Accuracy

There is a trade-off among the size of the mesh, the desired level of accuracy, and the amount of available computing resources.

The accuracy of the solution depends on the size of each of the individual elements (tetrahedra). Generally speaking, solutions based on meshes using thousands of elements are more accurate than solutions based on coarse meshes using relatively few elements. To generate a precise description of a field quantity, each element must occupy a region that is small enough for the field to be adequately interpolated from the nodal values.

However, generating a field solution involves inverting a matrix with approximately as many elements as there are tetrahedra nodes. For meshes with a large number of elements, such an inversion requires a significant amount of computing power and memory. Therefore, it is desirable to use a mesh fine enough to obtain an accurate field solution but not so fine that it overwhelms the available computer memory and processing power.

To produce the optimal mesh, HFSS uses an iterative process, called an adaptive analysis, in which the mesh is automatically refined in critical regions. First, it generates a solution based on a coarse initial mesh. Then, it refines the mesh in areas of high error density and generates a new solution. When selected parameters converge to within a desired limit, HFSS breaks out of the loop.

Related Topics

The Finite Element Method

The HFSS Solution Process

To calculate the S-matrix associated with a structure with ports, HFSS does the following:

- Divides the structure into a finite element mesh.
- Computes the modes on each port of the structure that are supported by a transmission line having the same cross-section as the port.
- Computes the full electromagnetic field pattern inside the structure, assuming that one mode is excited at a time.
- Computes the generalized S-matrix from the amount of reflection and transmission that occurs.

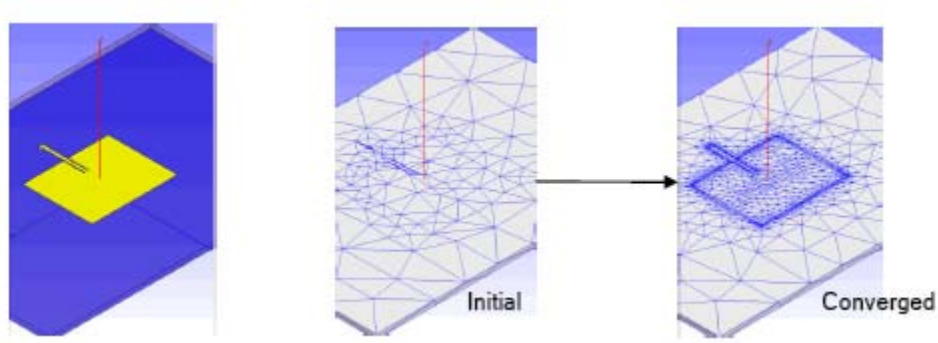
The resulting S-matrix allows the magnitude of transmitted and reflected signals to be computed directly from a given set of input signals, reducing the full 3D electromagnetic behavior of a structure to a set of high frequency circuit parameters.

The Mesh Generation Process

Following is the general mesh generation process:

1. HFSS generates an initial mesh, which includes surface approximation settings. If necessary, the mesher will automatically perform any repairs needed to recover an accurate mesh representation of a model. The solution profile will indicate when mesh repairs have been made, and the results of these repairs will be displayed per object in the mesh statistics panel.
2. If lambda refinement was requested, HFSS refines the initial mesh based on the material-dependent wavelength.
3. Any mesh operations that were defined are used to refine the mesh.
4. If ports were defined, HFSS iteratively refines the 2D mesh at the ports.
5. Using the resulting mesh, HFSS computes the electromagnetic fields that exist inside the structure when it is excited at the solution frequency.
6. If you are performing an adaptive analysis, HFSS uses the current finite element solution to estimate the regions of the problem domain where the exact solution has strong error. Tetrahedra in these regions are refined.
7. HFSS generates another solution using the refined mesh.
8. HFSS recomputes the error, and the iterative process (solve — error analysis — adaptive refinement) repeats until the convergence criteria are satisfied or the maximum number of

adaptive passes is completed.



9. If a frequency sweep is being performed, then HFSS solves the problem at the other frequency points without further refining the mesh. An adaptive solution is performed only at the specified solution frequency.

Note HFSS does not generate an initial mesh each time it starts the solution process. The initial mesh is generated only if a current mesh is unavailable.

Related Topics

Reverting to the Initial Mesh

Seeding the Mesh

Guidelines for Seeding the Mesh

Length-Based Mesh Refinement

Skin Depth-Based Mesh Refinement

Surface Approximation Settings

Guidelines for Modifying Surface Approximation Settings

Meshing Region Vs. Problem Region

Mesh Refinement on Ports

Model Resolution

Seeding the Mesh

In HFSS, mesh operations are optional mesh refinement settings that enable you to provide HFSS with engineering guidance based on your knowledge of the parts of the model geometry that are critical to the structure's electromagnetic performance. Providing such guidance to HFSS prior to beginning the adaptive analysis process can reduce (sometimes extensively) the number of passes necessary to converge upon a field solution as well as the final number of tetrahedra in the mesh for that solution. Although adaptive analysis convergence targets areas where field behavior is found, refining the mesh using more than the standard criteria, such as material characteristics, can result in finding areas of critical field behavior as soon as the first few passes are solved.

The technique of guiding HFSS's mesh construction is referred to as "seeding" the mesh. Seeding is performed using the **Mesh Operations** commands on the **HFSS** menu.

You can instruct HFSS to refine the length of tetrahedral elements on a surface or within a volume until they are below a certain value (length-based mesh refinement) or you can instruct HFSS to refine the surface triangle length of all tetrahedral elements on a surface or volume to within a specified value (skin depth-based mesh refinement.) These types of mesh operations can be defined at any time. If you apply them before the adaptive solution process, they are used to refine the initial mesh after it has been generated. You can also choose to apply mesh operations without generating a solution, in which case the mesh operations are applied to the current mesh.

In a few circumstances, you may also want to define a mesh operation that modifies HFSS's surface approximation settings for one or more faces. Surface approximation settings are only applied to the initial mesh.

Related Topics

Defining Mesh Operations

Technical Notes: The Mesh Generation Process

Guidelines for Seeding the Mesh

While seeding the mesh is not required, it is useful in the following conditions:

- Seeding the mesh inside a volume in the model geometry where regions of strong electric or magnetic fields (with strong capacitive or inductive loading) are expected. Examples include a capacitively loaded gap in a resonant structure, sharp waveguide angles or corners, or gaps between multi-coupled lines in filter structures.
- Seeding the mesh on every face of higher aspect ratio boundaries, such as long PCB traces or on the surfaces of long wires. Spacing the mesh points roughly equal to the trace width of the wire diameter enables you to more accurately capture the behavior of the high-aspect structure from the first adaptive pass.

Related Topics

Defining Mesh Operations

Length-Based Mesh Refinement Process

When you request length-based mesh refinement, you instruct HFSS to refine the length of tetrahedral elements until they are below a specified value. The length of a tetrahedron is defined as the length of its longest edge.

You can specify the maximum length of tetrahedra on faces or inside of objects. You can also specify the maximum number of elements that are added during the refinement. When the initial mesh has been generated, the refinement criteria you specified will be used to refine the initial mesh.

Related Topics

Assigning Length-Based Mesh Refinement on Object Faces

Assigning Length-Based Mesh Refinement Inside Objects

Skin Depth-Based Mesh Refinement

When you request skin depth-based mesh refinement, you instruct HFSS to refine the surface triangle length of all tetrahedral elements on a face to within a specified value. A layered mesh is created based on the surface mesh. The layers are graded based on the skin depth and number of layers you specify.

During skin depth-based mesh refinement, HFSS creates a series of layers that are planes parallel to the object face, and that are spaced within the specified skin depth. For each point on the surface of the face, a series of points (P0, P1, P2, ..., Pn) are added to the mesh, where n is the number of layers. P0 is the point on the surface and the distance from P0 to Pn is the skin depth. The points are spaced in a non-uniform manner, with the distance between them decreasing in a geometric progression, as you move from Pn to P0.

For example, if

Skin Depth:	12 mm
Number of Layers of Elements:	4

then

Distance [P0,P1]:	0.8 mm.
Distance [P1,P2]:	1.6 mm.
Distance [P2,P3]:	3.2 mm.
Distance [P3,P4]:	6.4 mm.
Distance [P0,P4]:	0.8 + 1.6 + 3.2 + 6.4 = 12 mm

The skin depth-based refinement first satisfies the surface triangle edge length criterion, then introduces the series of points to each additional layer. If a limit has been placed on mesh growth, one of the following happens:

- The limit is set high enough to complete the skin depth refinement.
- The limit is set high enough to satisfy the surface triangle edge length criterion, but not high enough to complete the depth seeding.
- The limit is not set high enough to satisfy even the surface triangle edge length criterion.

Because refining by skin depth can add many seeding points, you should first refine the surface of the object using length-based mesh refinement to obtain an accurate count of the number of points HFSS will add when refining by skin depth. This allows you to reach the surface edge length criterion and approximate the number of elements in the mesh and the number of points on the surfaces before proceeding to skin depth seeding.

The refinement criteria you specified are used to refine the current mesh.

Related Topics

Assigning Skin Depth-Based Mesh Refinement on Object Faces

Surface Approximation Settings

Object surfaces in HFSS may be planar, cylindrical or conical, toroidal, spherical, or splines. The original model surfaces are called *true surfaces* . To create a finite element mesh, HFSS first divides all true surfaces into triangles. These triangulated surfaces are called faceted surfaces because a series of straight line segments represents each curved or planar surface.

For planar surfaces, the triangles lie exactly on the model faces; there is no difference in the location or the normal of the true surface and the meshed surface. When an object's surface is non-planar, the faceted triangle faces lie a small distance from the object's true surface. This distance is called the *surface deviation* , and it is measured in the model's units. The surface deviation is greater near the triangle centers and less near the triangle vertices.

The normal of a curved surface is different depending on its location, but it is constant for each triangle. (In this context, "normal" is defined as a line perpendicular to the surface.) The angular difference between the normal of the curved surface and the corresponding mesh surface is called the *normal deviation* and is measured in degrees.

The *aspect ratio* of triangles used in planar surfaces is based on the ratio of circumscribed radius to the in-radius of the triangle. It is unity for an equilateral triangle and approaches infinity as the triangle becomes thinner.

You can modify the surface deviation, the maximum permitted normal deviation, and the maximum aspect ratio of triangles settings on one or more faces at a time in the **Surface Approximation** dialog box. (Click **HFSS>Mesh Operations>Assign>Surface Approximation**.)

The surface approximation settings are applied to the initial mesh.

Note For the initial mesh, all the vertices of the triangles lie on the true surfaces. During adaptive meshing, the vertices are added to the meshed surfaces, not to the true surfaces.

Related Topics

Modifying Surface Approximation Settings

Technical Notes: Guidelines for Modifying Surface Approximation Settings

Technical Notes: The Mesh Generation Process

Guidelines for Modifying Surface Approximation Settings

If you intend to modify the surface approximation settings for an object face or faces, keep the following guidelines in mind:

- When necessary, override the default surface approximation settings to represent curved surfaces more accurately. More accurate representation will increase the mesh size and consume more CPU time and memory. The default settings are adequate for most circumstances.
- If you want to obtain a faster solution by using a cruder representation of curved surfaces, set the coarser setting for the whole object, not just a single face.
- It is difficult for HFSS to satisfy aspect ratio demands if the aspect ratio value is set close to 1 because an arbitrary shape cannot be filled with only equilateral triangles. Therefore,

setting the aspect ratio to 1 can lead to unreasonably large meshes. HFSS limits the aspect ratio to 4 for planar objects and 1.2 for curved objects.

Related Topics

Modifying Surface Approximation Settings

Technical Notes: Surface Approximation Settings

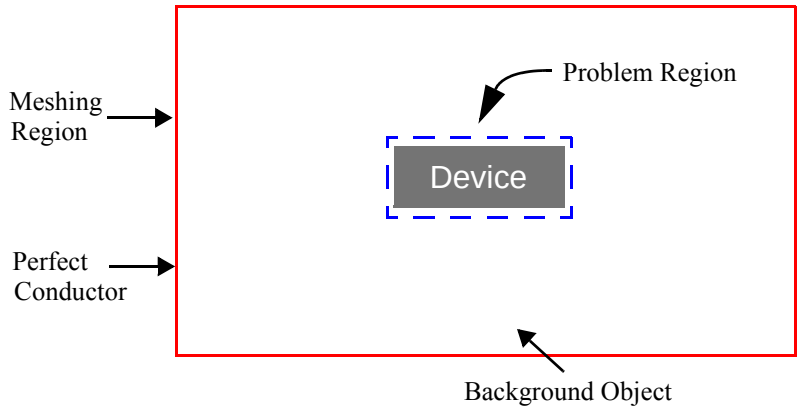
Meshing Region Vs. Problem Region

HFSS distinguishes between the problem region and the meshing region. The *problem region* is the region in which the solution is generated and the mesh is refined. The *meshing region*, which includes the problem region, is the area in which an initial mesh is generated. After an initial mesh is generated, the mesh is refined only in the problem region.

The problem region encompasses an area that is just large enough to include the entire design, but no larger. HFSS automatically defines the problem region during the solution process. If you are interested in effects outside of the structure, such as radiated effects, then you can create a virtual object to expand the size of the problem region to include these areas.

The meshing region, like the problem region, is a box that completely encloses the structure. However the meshing region must be at least 10 times larger than the model. The part of the meshing region not occupied by objects is considered to be the *background* object. The background extends to the boundaries of the meshing region and fills in any voids not occupied by objects. Since the background object is defined as a perfect conductor, no solution is generated inside the background even though an initial mesh is generated for it. HFSS automatically defines the meshing region during the solution process.

The problem region and the meshing region are illustrated below.



Model Resolution

Model Resolution is a setting that determines the smallest details of a model that the mesher should capture and represent in the mesh.

Many times the analysis starts with the geometry already drawn in a different tool for different purpose. Some tools are designed for manufacturing and the resulting models contain lots of extra

details not needed for electromagnetic analysis. If the user removes such details in the original tool the results will be better. But if the user does not have access to the original drawing tool or redrawing the model without these details is not possible, **Model Resolution** is another way to remove the details from analysis.

When the user sets the model resolution length to be L, the mesher will start with a surface representation of the model accurate to the modeler's tolerance limit. Then it will progressively remove edges, move points, merge points etc., within the allowable model resolution limit and simplify the surface mesh. During this process, tiny fillets, rounds, and chamfer protrusions are removed.

Other common model translation anomalies are also handled using **Model Resolution**. For example, some geometry engines will blindly export all of the surfaces as splines. When a user imports such a model for analysis, it would result in very large number of triangles. If the surface can be represented by a smaller set of triangles using Model Resolution, this procedure would reduce the number of triangles in the surface mesh.

The user can start with a model resolution length around 0.1*wavelength. If the model resolution length chosen by the user is too large, the mesher will detect it and report it as an error. The model resolution length is specified in the user units of the modeler. It can be set on selected bodies only. The default value is 100* the tolerance limit of the ACIS modeler.

Related Topics

Specifying Model Resolution

Port Solutions

The excitation field pattern at each port must be calculated before the full 3D electromagnetic field inside a structure can be calculated. HFSS calculates the natural field patterns (or modes) that can exist inside a transmission structure with the same cross-section as the port. The resulting 2D field patterns serve as boundary conditions for the full 3D problem.

- Excitation Fields
- Wave Equation
- Mesh Refinement on Ports
- Modes
- Multiple Ports on the Same Face
- Port Accuracy
- Calculating Characteristic Impedance
- Complex Propagation Constant

Excitation Fields

HFSS assumes that each port is connected to a uniform waveguide that has the same cross-section as the port. A waveguide supports an infinite number of modes with a given field pattern and propagation constant.

Therefore, the excitation field is one of the modes specified for a port where mode *m* can be expressed by:

$$\boldsymbol{E}_m(x, y, z, t) = \Re\left[\boldsymbol{E}_m(x, y)e^{j\omega t - \gamma_m z}\right] \quad (1)$$

where

- $\Re$ is the real part of a complex number or function.
- **E**m(x,y) is the electric field mode pattern of mode *m*.
- $\gamma_m = \alpha_m + j\beta_m$ is the complex propagation constant of mode *m*, where
  - α_m is the attenuation constant of mode m.
  - β_m is the propagation constant of mode m that determines, at a given time *t*, how the phase angle varies with z.
- $\omega = 2\pi f$ is angular frequency of oscillation,
- *j* is the imaginary unit, $\sqrt{-1}$.

In this context, the x- and y-axes are assumed to lie in the cross-section of the port; the z-axis lies along the direction of propagation.

Wave Equation

The field pattern of a traveling wave inside a waveguide can be determined by solving Maxwell's equations. For each mode m, the port solver solves the following equation obtained from Maxwell's equation.

$$\nabla\times\left(\frac{1}{\mu_r}\nabla\times\boldsymbol{E}_m(x, y)e^{-\gamma_m z}\right) - k_0^2\varepsilon_r\boldsymbol{E}_m(x, y)e^{-\gamma_m z} = 0 \quad (1)$$

where

- $k_0 = \omega/c$ is the wave number of free-space
- $\mu_r(x,y)$ is the complex relative permeability.
- $\varepsilon_r(x,y)$ is the complex relative permittivity.

By solving the equation, the electric field mode pattern $E_m(x,y)$ and the propagation constant γ_m are both obtained for all the modes specified for a given port. For the current based impedance calculations, the port solver also solves independently for $H_m(x,y)$ using a corresponding equation for the magnetic field.

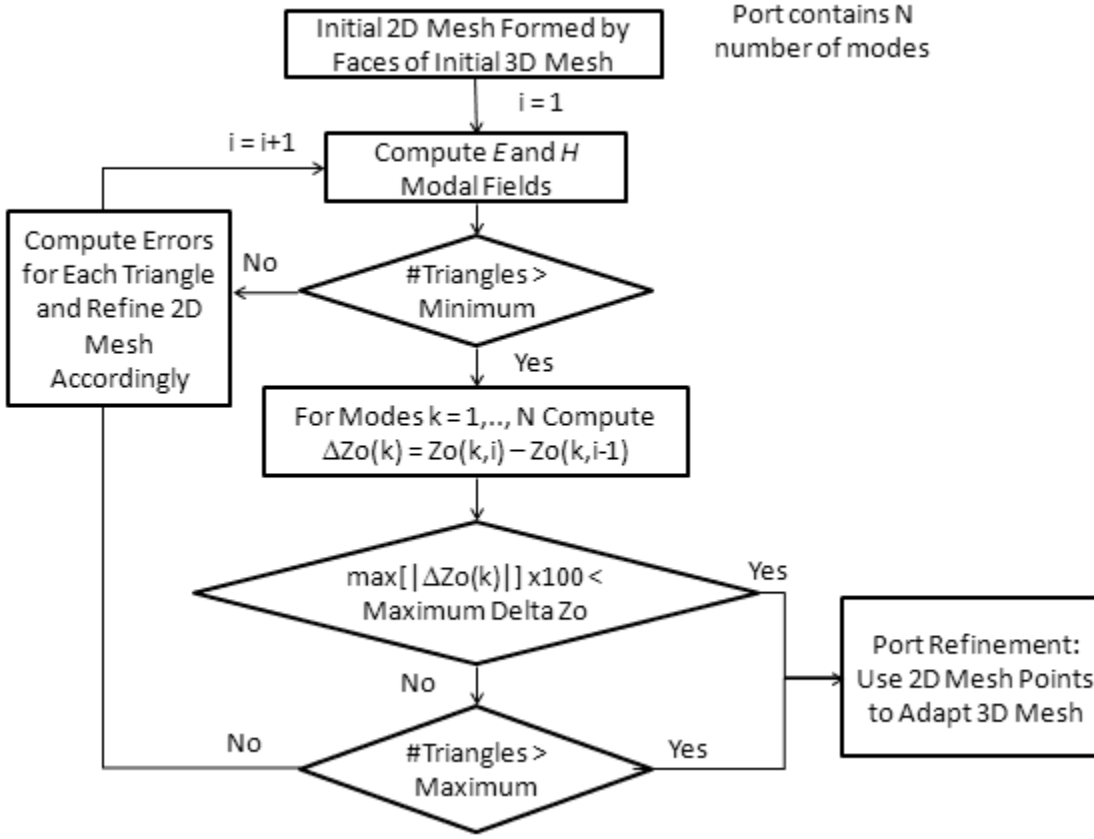
Also note that the excitation field pattern computed by the port solver is valid only at a given frequency. A different excitation field pattern is computed for each frequency point of interest.

Mesh Refinement on Ports

The port solver computes mode patterns and propagation constants based on solutions of a finite element system of equations. The finite element mesh associated with each port is a 2D mesh of triangles corresponding to the faces of the tetrahedra that lie on the planar port surface. The port solver performs an iterative refinement of the triangular mesh.

User settings for the iterative refinement can be found under Port Options in the **Advanced** tab of the **Solution Setup** window. Those settings are reflected in the flow chart below that describe the

refinement procedure where #Triangles refer to the number of triangles of the current mesh in the refinement process. The numerical implementation of $\Delta Zo(k)$ in the flow chart is actually based on relative difference when $|Zo(k)| > 1.0$. Moreover, for terminal projects the comparison is done for each entry of the terminal characteristic impedance matrix rather than the modal impedances such that the worst entry in the impedance matrix satisfies the user specified value.



For a good technical discussion of the port solution procedure, refer to the following:

Jin-Fa Lee, Din-Kow Sun, and Zoltan J. Cendes, "Full-Wave Analysis of Dielectric Waveguides Using Tangential Vector Finite Elements," *IEEE Transactions on Microwave Theory and Techniques*, vol. 39, No 8, August 1991.

Related Topics

Technical Notes: The Mesh Generation Process

Modes

For a waveguide or transmission line with a given cross-section, there is a series of basic field patterns, or modes, that satisfy Maxwell's equations at a specific frequency. Any linear combination of these modes can exist in the waveguide. A user is responsible to specify the number

of modes to be included in a wave port but not for lumped ports since they are restricted to contain only one mode. In general, in order to obtain accurate S-parameters the number of modes defined for a given port should be the same or greater than the number of propagating modes at the highest frequency of interest. Note that the port solver automatically determines what modes should be included based on the specified number of modes. The modes included in a port are based on a sorting criteria to extract the dominant set of modes out of the infinite set of possible modes. If there is no prior knowledge to the number of propagating modes, the user can do a ports only solution while specifying a reasonably large number of modes and view the 'Gamma' values in the Matrix Data panel to determine what modes are propagating.

If a port is defined on a surface where there is a geometrical discontinuity in the normal direction of the port, it may not be enough to include only the propagating modes for accurate S-parameters. If possible, it is recommended to avoid this case by modeling a short section of the port transmission line such that all the higher order modes have decayed and do not contribute to the field on the port surface. The length of the constant cross-section segment that has to be included in the model depends on the value of the mode's attenuation constant, α.

Degenerate Modes

Degenerate modes refer to the case when a set of modes have the same complex propagation constant which means that any linear combination of those modes is still a valid set of natural (or eigen) modes. When you become aware that a wave port contains degenerate modes as a part of the requested modes, you should use the Mode Alignment feature in HFSS.

Multiple Ports on the Same Face

Visualize a port face on a microstrip that contains two conducting strips side by side as two separate ports. If the two ports are defined as being separate, the system simulates the case in which the two ports are connected to uncoupled transmission structures. It is as if a conductive wall separates the excitation waves. However, in actuality, there will be electromagnetic coupling between the two strips.

To model this coupling accurately, analyze the two ports as a single port with multiple modes. In general, if there are N disconnected conductors in the port cross-section, at least N - 1 modes are required for an accurate solution. For example, if the port consists of two adjacent microstrip lines surrounded by a conducting enclosure, N = 3; therefore at least two modes should be defined on the port. Assign an equal number of terminals as modes. Refer to Assigning Wave Ports for Terminal Solutions for more information.

If the multi-conductor port plane is near discontinuities within the 3D model, additional modes beyond N - 1 may be necessary. However, if you define terminals on a multi-conductor port, the presence of non-quasi transverse electromagnetic (TEM) modes will adversely affect the entries of any computed terminal matrices. Therefore, rather than increase the number of modes beyond the required N - 1, extend the port outward until any higher-order modes have sufficient attenuation to be omitted from consideration.

Port Accuracy

Generally, the port accuracy defaults**,** specified under the Port Options in the Advanced tab of the **Solution Setup** window, are adequate. You may want improved port accuracy under the following conditions:

- You are interested primarily in the port impedances. Port impedances are computed as part of the port solution.
- You need to lower the noise floor to catch S-parameters that are expected to be in the –70 dB range.

The port accuracy settings are only used for port adapt. If those settings are modified after port adapt nothing will happen. Therefore, to specify new port accuracy settings, add another solution setup and generate a new solution.

Refining the mesh at the ports causes HFSS to refine the mesh for the entire structure as well. This occurs because it uses the port field solutions as boundary conditions when computing the full 3D solution. Therefore, specifying too small a port field accuracy can result in an unnecessarily complex finite element mesh.

Calculating Characteristic Impedance

Each port in a structure being analyzed can be viewed as a cross-section of a transmission line. HFSS computes the characteristic impedance of each port in three ways — as Zpi, Zpv, and Zvi impedances. You have the option of specifying which impedance will be used in the renormalization calculations.

- For TEM waves, the Zvi impedance converges on the port's actual impedance and should be used.
- When modeling microstrips, it is sometimes more appropriate to use the Zpi impedance.
- For slot-type structures (such as inline or coplanar waveguides), Zpv impedance is the most appropriate.

HFSS will always calculate Zpi impedance, the impedance calculation using power and current, which are well-defined for a port because they are computed over the area of the port. Zpv and Zvi are not calculated by default. This is because V is computed by integrating along a user-defined integration line. To renormalize the solution to a Zpv or Zvi characteristic impedance, you must have defined an impedance line.

Under the **Matrix Data** tab of the **Solution Data** dialog box, the characteristic impedance can be displayed as magnitude/ phase, real/ imaginary, magnitude, phase, real, or imaginary.

For more information on the computation of impedances, refer to the following:

Bruno Bianco, Luigi Panini, Mauro Parodi, and Sandro Ridella, "Some Considerations about the Frequency Dependence of the Characteristic Impedance of Uniform Microstrips," *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-26 No. 3, March 1978.

Edward F. Kuester, David C. Chang, and Leonard Lewin, "Frequency-Dependent Definitions of Microstrip Characteristic Impedance," *International URSI Symposium on Electromagnetic Waves*, Munich, 26-29 August 1980, pp. 335 B/1-3.

Related Topics

Renormalizing S-Matrices

Calculating the PI Impedance

The Z_{pi} impedance is the impedance calculated from values of power (P) and current (I):

$$Z_{pi} = \frac{P}{I \cdot I} \quad (1)$$

The power and current are computed directly from the simulated fields. The power passing through a port is equal to the following:

$$P = \oint_S E \times H ds \quad (2)$$

where the surface integral is over the surface of the port.

The current is computed by applying Ampere's law to a path around the port:

.

$$I = \oint_l H \bullet dl \quad (3)$$

While the net current computed in this way will be near zero, the current of interest is that flowing into the structure, I^-, or that flowing out of the structure, I^+. In integrating around the port, HFSS keeps a running total of the contributions to each and uses the average of the two in the computation of impedances.

Note Zpi calculation is modified when a symmetry E plane touches a port. The current on the symmetry E plane is included in the Zpi calculation for a port with more than one conductor, where as for a single conductor port, such as a waveguide, the current on the symmetry E plane is not included in the Zpi calculation.

Calculating the PV Impedance

The Z_{pv} impedance is the impedance calculated from values of power (P) and voltage (V):

$$Z_{pv} = \frac{V \bullet V}{P} \quad (1)$$

where the power and voltage are computed directly from the simulated fields.

The power is computed in the same way as the Z_{pi} impedance. The voltage is computed as follows:

$$V = \oint_l E \bullet dl \quad (2)$$

The result over which HFSS integrates is referred to as the impedance line — which is defined when the ports are set up. To define the impedance line for a port, select the two points across which the maximum voltage difference occurs. You must define an integration line to specify where the maximum voltage difference will be.

Calculating the VI Impedance

The Z_{vi} impedance is given by:

$$Z_{vi} = \sqrt{Z_{pi}Z_{pv}}$$

For TEM waves, the Z_{pi} and Z_{pv} impedances form upper and lower boundaries to a port's actual characteristic impedance. Therefore, the value of Z_{vi} approaches a port's actual impedance for TEM waves.

Impedance Multipliers

If a symmetry plane has been defined (allowing the model of a structure to be cut in half), the impedance computations must be adjusted by specifying an impedance multiplier. The need for this multiplier can be understood by looking at how the use of symmetry affects the computation of Z_{pv}.

In cases where a perfect E plane of symmetry splits a structure in two, only one-half of the voltage differential and one-half of the power flow can be computed by the system. Therefore, since the Z_{pv} impedance is given by:

$$Z_{pv} = \frac{V \bullet V}{P}$$

the computed value is one-half the desired value. An impedance multiplier of 2 must be specified in such cases.

In cases where a perfect H plane of symmetry splits a structure in two, only one-half of the power flow is seen by the system but the full voltage differential is present. Therefore, structures split in half with perfect H symmetry planes result in computed impedances that are twice those for the full structure. An impedance multiplier of 0.5 must be specified in such cases.

If multiple symmetry planes are used or if only a wedge of a structure is modeled, you must adjust the impedance multiplier accordingly.

If you have defined a symmetry plane, the computed impedances will not be for the full structure. Generally, use one of the following values for the impedance multiplier:

- If the structure has a perfect E plane of symmetry, use 2. Such models have one-half of the voltage differential and one-half of the power flow of the full structure, resulting in impedances that are one-half of those for the full structure.
- If the structure has a perfect H plane of symmetry, enter 0.5. Such models have the same voltage differential but half the power flow of the full structure, resulting in impedances that are twice those for the full structure.
- If the structure has a combination of perfect H and perfect E boundaries, adjust accordingly. For example, you do not have to enter an impedance multiplier for a structure with both a perfect E and perfect H boundary since you would be multiplying by 2 and 0.5.

Related Topics

Setting the Impedance Multiplier

Calculating Terminal Characteristic Impedance Matrix

See Modes to Terminals Conversion.

Complex Propagation Constant

Each port is assumed to be connected to a transmission structure that has the same cross-section as the port. The complex propagation constant, γ, of these transmission lines is computed by HFSS, and is given by $\gamma = \alpha + j\,\beta$, where:

- α is the attenuation constant of a signal in the transmission structure. It is the real component of the propagation constant and has units of nepers per meter.
- β is the phase constant associated with the wave. It is the imaginary component of the propagation constant and has units of radians per meter.

Under the **Matrix Data** tab of the **Display Items Dialog**, gamma can be displayed as magnitude/ phase, real/ imaginary, magnitude, phase, real, or imaginary.

Calculating the Effective Wavelength (Lambda)

The effective wavelength, λ_{eff}, is calculated from

$$\lambda_{eff} = \frac{2\pi}{\beta} \quad (1)$$

where β is the phase constant associated with the wave.

Under the **Matrix Data** tab of the **Solution Data** dialog box, lambda is displayed when **Gamma** is selected as the matrix type.

Calculating the Relative Permittivity (Epsilon)

The relative permittivity, ε_r, is calculated using

$$\lambda_{eff} = \frac{c}{\sqrt{\varepsilon_r} f} \quad (1)$$

where

- λ_{eff} is the effective wavelength given in meters.
- c is the speed of light.
- f is the frequency of the wave.

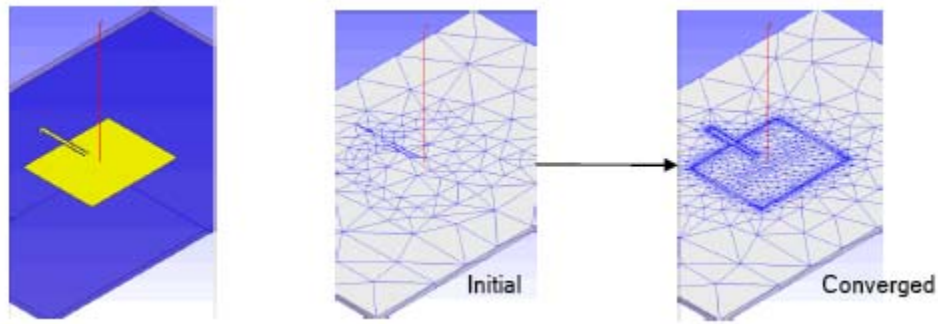
Under the **Matrix Data** tab of the **Solution Data** dialog box, epsilon is displayed when **Gamma** is selected as the matrix type.

The Adaptive Analysis Process

An adaptive analysis is a solution process in which the mesh is refined iteratively in regions where the error is high, which increases the solution's precision. You set the criteria that control mesh refinement during an adaptive field solution. Many problems can be solved using only adaptive refinement.

Following is the general process followed during an adaptive analysis:

1. HFSS generates an initial mesh, and applies lambda refinement and manual refinements.
2. Using this mesh, HFSS computes the electromagnetic fields that exist inside the structure when it is excited at the solution frequency. (If you are running a fast frequency sweep, an adaptive solution is performed only at the specified solution frequency.)
3. Based on the current finite element solution, HFSS estimates the regions of the problem domain where the exact solution has strong error. Tetrahedra in these regions are refined.
4. HFSS generates another solution using the refined mesh.
5. HFSS recomputes the error, and the iterative process (solve — error analysis — refine) repeats until the convergence criteria are satisfied or the requested number of adaptive passes is completed.



6. If a frequency sweep is being performed, HFSS then solves the problem at the other frequency points without further refining the mesh.

Maximum Delta S

For designs with ports.

The delta S is the change in the magnitude of the S-parameters between two consecutive passes. If the magnitude and phase of all S-parameters change by an amount less than the **Maximum Delta S Per Pass** value from one iteration to the next, the adaptive analysis stops. Otherwise, it continues until the requested number of passes is completed.

For example, if you specify **0.1** as the **Maximum Delta S Per Pass**, HFSS continues to refine the mesh until the number of requested passes is completed or until the magnitude of the complex delta of all S-parameters changes by less than **0.1**.

The maximum delta S is defined as

$$Max_{ij}\left|S_{ij}^{N} - S_{ij}^{N-1}\right| \quad (1)$$

where:

- i and j cover all matrix entries.

- N represents the pass number.

Note Delta S is computed on the appropriate S-parameters - modal or terminal - after the S-parameters have been de-embedded and renormalized.

Related Topics

Viewing the Maximum Magnitude of Delta S Between Passes

Maximum Delta E

For designs with voltage sources, current sources, or incident waves.
Not applicable to designs with ports.

The delta E is the difference in the relative energy error from one adaptive solution to the next. It is a measure of the stability of the computed field values from pass to pass. As the solution converges, delta E approaches zero.

The **Maximum Delta E Per Pass** value is a stopping criterion for the adaptive solution. If the delta E falls below this value, the adaptive analysis stops. Otherwise, it continues until the convergence criteria are reached.

The data represents the delta E for all tetrahedra.

Percent of Tetrahedra Refined Per Pass

The value you set for **Percent Refinement Per Pass** determines how many tetrahedra are added at each iteration of the adaptive refinement process. For instance, entering **30** causes the mesh to increase approximately 30 percent each pass. The tetrahedra with the highest error will be refined. If your mesh consisted of 1000 elements, the tetrahedra would be refined so that 300 new elements are added to the mesh. Generally, you can accept the default value.

Magnitude Margin

For solutions in which convergence criteria for specific S-matrix entries were specified.

For each element in the S-matrix, the magnitude margin is the difference between the S-parameter delta magnitude and the target delta magnitude, which was specified in the **Matrix Convergence** dialog box. The magnitude margin reported under the **Convergence** tab is the maximum of these values over the entire matrix. The magnitude margin is defined as

$$Max_{ij}\left[\left|S_{ij}^{N}-S_{ij}^{N-1}\right|-\left|M_{ij}\right|\right] \quad (1)$$

where M_{ij} is the matrix convergence entry. It indicates the solution's proximity to the target delta magnitude. If the solution has converged within the target delta magnitude, a value of zero will be reported for the pass.

Phase Margin

For solutions in which convergence criteria for specific S-matrix entries were specified.

For each element in the matrix, the phase margin is the difference between the S-parameter delta phase and the target delta phase, which was specified in the **Matrix Convergence** dialog box. The phase margin reported under the **Convergence** tab is the maximum of these values over the entire matrix. The phase margin is defined as

$$Max_{ij}\left[\left|phaseS_{ij}^{N} - phaseS_{ij}^{N-1}\right| - phaseM_{ij}\right] \quad (1)$$

where M_{ij} is the matrix convergence entry. The phase margin indicates the solution's proximity to the target delta phase. If the solution has converged within the target delta phase, a value of zero will be reported for the pass.

Note When the Mag S becomes small (near to zero) its phase becomes indefinite and insignificance due to mathematical issue so that Phase Margin will be discarded.

Maximum Delta Frequency

For Eigenmode solutions.

At any time during the solution process, you can view the percent difference in the resonant frequencies from one adaptive solution to the next, or the maximum delta frequency. This is a measure of the stability of the computed frequencies from pass to pass and is available only two or more adaptive passes are completed.

For lossless problems, the maximum delta frequency is the largest percent change in the real part of the frequency for any of the calculated modes. For lossy problems, the maximum delta frequency is the greater of two quantities: the largest percent change in the real part of the frequency over all the modes, and the largest percent change in the imaginary part of the frequency.

Max Delta (Mag S)

For solutions in which convergence criteria for specific S-matrix entries were specified.

The Max delta(Mag S) is the maximum difference of S-Matrix magnitudes between two consecutive passes. If the difference in magnitudes of the S matrices change by an amount less than the **Maximum Delta Mag S** value from one pass to the next, this satisfies the part of the convergence criteria.

$$Max\left|S_{ij}^{N} - S_{ij}^{N-1}\right| \quad (1)$$

Max Delta (Phase S)

For solutions in which convergence criteria for specific S-matrix entries were specified.

The Max delta(Phase S) is the maximum difference of S-Matrix phase between two consecutive passes. If the difference in phase of the S matrices change by an amount less than the **Maximum Delta Phase S** value from one pass to the next, this satisfies this part of the convergence criteria.

$$Max(PhaseS_{n} - PhaseS_{n-1}) \quad (1)$$

Matrix Solvers

HFSS includes two solvers. The default solver and the Iterative Matrix Solver. See Enable Iterative Solver for a discussion of when to use the Iterative Matrix solver. For details on the default solver see Direct Matrix Solver. For details on the Iterative Matrix solver see *Technical Notes*: Iterative Matrix Solver.

Direct Matrix Solver

Direct methods obtain an exact solution to the following linear system of equation

$$Ax = b \quad (1)$$

where A is a matrix, b a right hand side and x the solution. A matrix solver using a direct method is called a direct matrix solver.

The most widely known direct method is Gaussian elimination, which uses elementary matrix operations to compute the solution. Most other direct solution techniques are either a variant of Gaussian elimination or are based upon a particular factorization of the equations that will allow an exact computation. One of the most commonly used such techniques is the LU decomposition, which is introduced below.

LU decomposition is both a factorization approach and closely related to Gaussian elimination. It is based upon the assumption that A can be decomposed into a product of two matrices or factors.

$$A = LU \quad (2)$$

where L is a lower triangular matrix (has elements only on the diagonal and below) and U is an upper triangular matrix (has elements only on the diagonal and above).

With the help of (2), we can equivalently solve (1) by first solving for y such that

$$Ly = b \quad (3)$$

then solving for x such that

$$Ux = y \quad (4)$$

The advantage of decomposition (2) is that the solution of triangular set of equations (3) and (4) is quite trivial. Namely, they can be solved directly by forward substitution and by backward substitution respectively. Furthermore, the factors in (2) are computed only once and can be reused in (3) and (4) for different right hand sides.

In LU decomposition, the major storage is for matrix A, factor L and factor U. The major operation is the decomposition (2), which is roughly equivalent to a matrix-matrix multiplication; while the operation in (3) and (4) combined, is equivalent to a matrix-vector multiplication. The total computational cost is S+mT with S>>T in general, where S is the number of operations for decomposition (2), m the number of right hand sides, and T the number of operations for both forward substitution and for backward substitution.

Direct methods are best used for solving system matrix equations with moderate size or with a large number of right hand sides. As the system to be solved becomes larger, the overhead associated with the more complicated iterative methods becomes less of an issue, and the iterative methods

should outperform the direct methods. For sparse systems, the use of direct methods is complicated by the possible introduction of more nonzero entries (fill-in) such that the L and U factors become much denser than the original matrix A.

Iterative Matrix Solver

This section contains information on the Iterative Matrix Solver.

- Guidelines for Using the Iterative Solver
- Multiprocessing and the Iterative Solver
- Iterative Solver Technical Details

Guidelines for Using the Iterative Solver

1. The iterative solver works most efficiently when it is enabled for designs that do not contain many excitations. (For example, the number of excitations is less than twice the number of processors.)
2. If you choose to take advantage of the iterative solver, and your analysis includes interpolating sweeps or discrete sweeps, the adaptive solution should be well converged at the higher end of the frequency band.
3. The **Relative Residual** provides a stopping criteria. The residual measures the convergence of the iterative solver to the solution of the matrix equation. Its value affects the performance of the iterative solver as follows:
   - Default is 1E-4. This gives accurate S-parameters and fields, indistinguishable from those generated by the direct solver. ANSYS recommends this residual.
   - With a larger residual, for example, 1E-3 or 1E-2, the iterative process will stop with fewer iterations and the solution will be less converged. S-parameters won't differ much from those of a direct solution, for example, a difference in third or second digit. Fields and antenna patterns are visually the same.
   - A residual of 0.1 can be used for quick adaptive mesh refinement early in the adaptive process, but S-parameters will be noticeably different.
   - A residual of 1 should never be used. The interface will not allow a residual above 0.1.

Multiprocessing and the Iterative Solver

In computing the pre-conditioner, the iterative solver uses multiple processors, if set under **Tools>Options>HFSS Options**, just like the multi-frontal solver does. The operations involved are very similar. After that, there is one iterative process per excitation. Each iterative process can only use one processor. If you have one excitation and multiple processors, you will see that HF3D.exe uses only one processor at this stage. If you have multiple excitations and multiple processors, HF3D.exe will use multiple processors at this stage. For example, with four processors and eight excitations, HF3D will perform the iterative processes for four excitations first, and for the other set of four excitations next.

Iterative Matrix Solver Technical Details

The iterative solver always saves significant memory - easily a factor two with simulations of intermediate size, and more with larger simulations. It is also faster than the multi-frontal solver with large simulations.

The following table shows asymptotic behavior with large simulations. N = number of unknowns. This shows that the iterative solver has a better asymptotic behavior both in RAM and in time.

	Time	RAM
Multi-frontal Solver	$N^{1.7}$	$N^{1.3}$
Iterative Solver	$N^{1.2}$	$N^{1.0}$

The iterative solver uses a preconditioner that is based on the next-lower order. Therefore, there is no iterative solver option when you solve with zeroth order.

Consider the matrix equation

$$Ax = b \quad (1)$$

where A is a matrix, b a right hand side and x the solution.

When A^{-1} is computationally expensive or the exact solution x is impossible, an alternative is to seek an approximation $\bar{x}$ to x, with an error $e = x - \bar{x}$. The exact solution can therefore be rewritten as

$$x = \bar{x} + e \quad (2)$$

Substituting (2) into (1) results in the so called residual equation

$$Ae = r \quad (3)$$

where r is the residual defined by

$$r = r - A\bar{x} \quad (4)$$

As aforementioned, the exact solution for e in (3) is impossible since it requires A^{-1}. However, if an approximation $M \approx A$ is available, the error e can be approximated in (3) by

$$\bar{e} = M^{-1}r \quad (5)$$

Finally, the approximation $\bar{x}$ is updated by

$$\bar{x} \leftarrow \bar{x} + \bar{e} \quad (6)$$

It is (4)-(6) that form the foundation of the iterative solution method. A matrix solver using the iterative solution method is called an iterative matrix solver. The method starts with an initial guess $\bar{x} = x_0$ and repeats (4)-(6) until the approximation $\bar{x}$ to x is within tolerance, or the number of iterations exceeds a given number. In the former case, it is said the solution converges; while in the latter, it doesn't.

The residual *r* is used for measuring the closeness of $\bar{x}$ to *x*. Since *A* and *b* in (1) can be scaled by the same factor without altering *x*, so does the residual *r* in (4). It typically makes more sense to replace ||*r*|| as the stopping criterion with the **relative residual**:

$$res = \frac{\|r\|}{\|b\|} \quad (7)$$

where $\|\ \|$ stands for vector norm.

M in (5) is called a preconditioner of *A*. A good preconditioner greatly reduces the number of iterations. The following makes *M* a good preconditioner: *M* is very similar to *A* in eigen spectrum and M^{-1} is computationally cheap. Some of the classic iterative matrix methods include: the Jacobi method where *M* is the diagonal of *A*, the Gauss-Seidel method where *M* is the lower triangular or upper triangular matrix of *A* and the successive over-relaxation method (SOR) where *M* is a weighted combination of the lower triangular and upper triangular matrix of *A*.

In the iterative method, the major storage is for matrix *A* and for preconditioner *M*. The major operation is a matrix-vector multiplication in (4). The computational cost is *S*+*mnT*, where *S* is the number of operations for setting up the preconditioner, *m* the number of right hand sides, *n* the average number of iterations per right hand side and *T* the number of operations for each iteration.

Details of the iterative solver in HFSS are given in:

D. K. Sun, J. F. Lee and Z. J. Cendes, "Construction of nearly orthogonal Nedelec bases for rapid convergence with multilevel preconditioned solvers", *SIAM Journal on Scientific Computing*, Vol. 23, No. 4, pp. 1053-1076, 2001.

I. Bardi, G. Peng and Z.J. Cendes, "Improvements in Adaptive Mesh Refinement and Multi-level methods in High Frequency Electromagnetics", ACES Symposium, 2002.

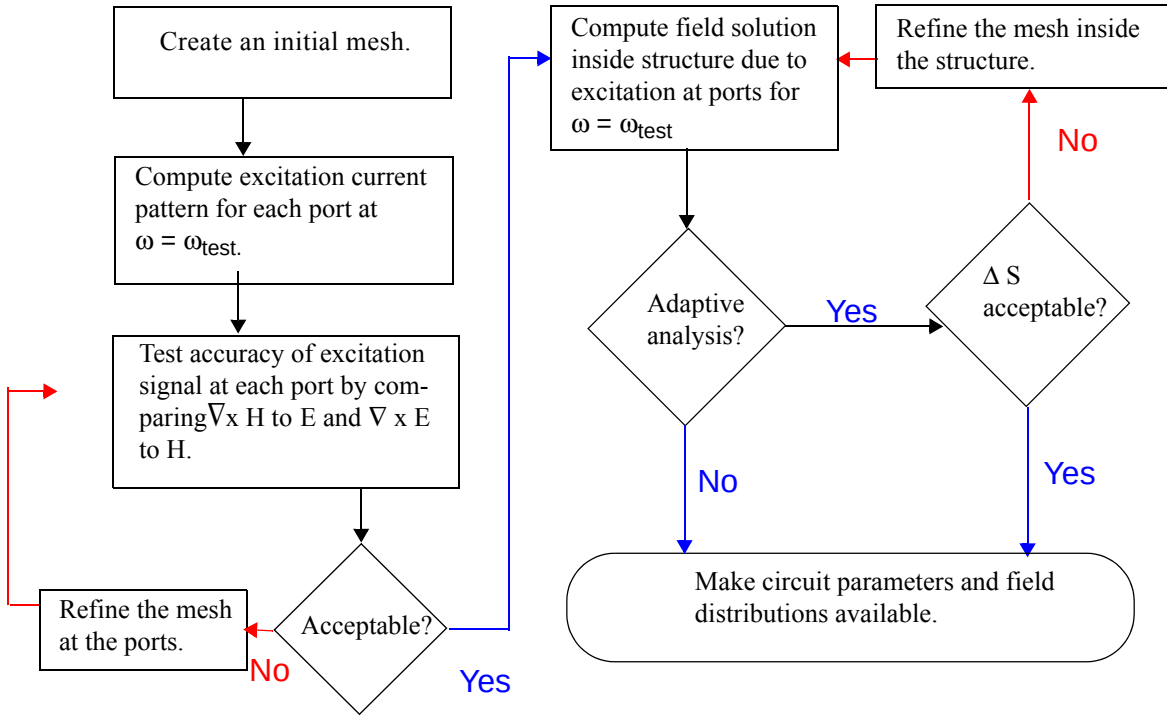
Related Topics

Enable Iterative Solver

Single Frequency Solution

A single frequency solution generates an adaptive or non-adaptive solution at a single frequency, the solution frequency specified in the **Solution Setup** dialog box, and is often the first step in performing a frequency sweep. An adaptive solution is one in which a finite element mesh is created

and automatically refined in the areas of highest error — increasing the accuracy of succeeding adaptive solutions. The procedure for performing a single frequency solution is shown below.



Frequency Sweeps

Perform a frequency sweep when you want to generate a solution across a range of frequencies. You may choose one of the following sweep types:

Fast	Generates a unique full-field solution for each division within a frequency range. Best for models that will abruptly resonate or change operation in the frequency band. A Fast sweep will obtain an accurate representation of the behavior near the resonance.
Discrete	Generates field solutions at specific frequency points in a frequency range. Best when only a few frequency points are necessary to accurately represent the results in a frequency range.
Interpolating	Estimates a solution for an entire frequency range. Best when the frequency range is wide and the frequency response is smooth, or if the memory requirements of a Fast sweep exceed your resources.

Fast Frequency Sweeps

A Fast sweep generates a unique full-field solution for each division within a frequency range. Choose a Fast sweep if the model will abruptly resonate or change operation in the frequency band. A Fast sweep will obtain an accurate representation of the behavior near the resonance.

HFSS uses the center frequency of the frequency range to select an appropriate eigenvalue problem with which to generate a solution for the entire Fast sweep. It then uses an Adaptive Lanczos-Padé Sweep (ALPS)- based solver to extrapolate the field solution across the requested frequency range from the center frequency field solution.

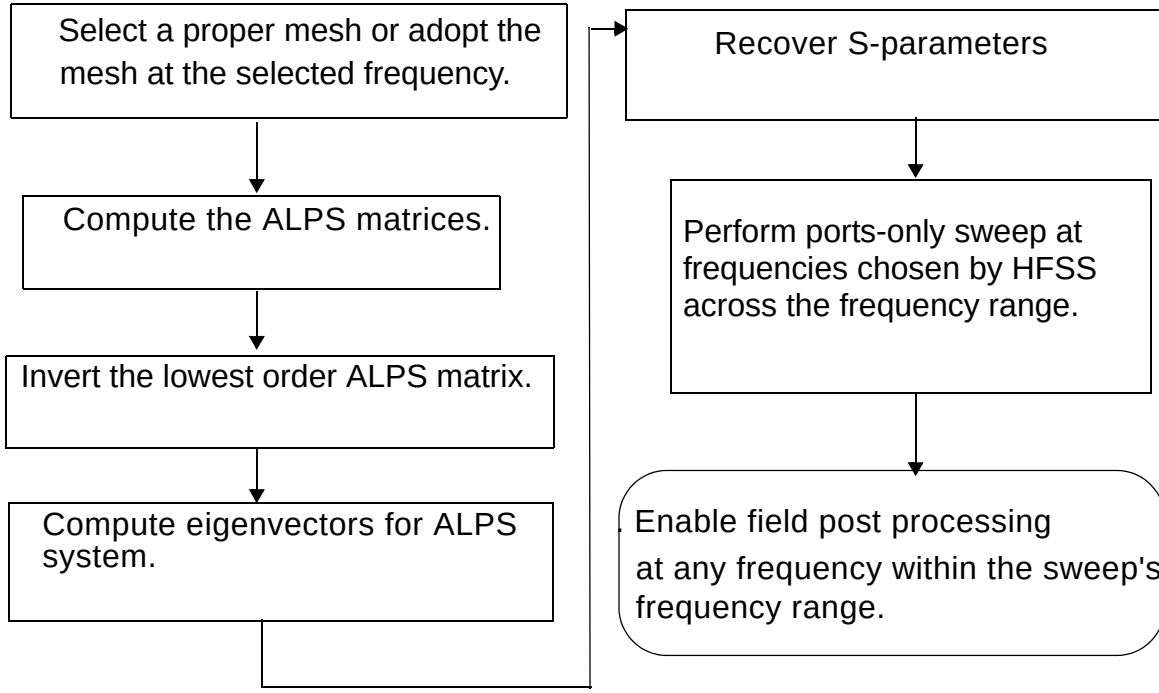
HFSS always uses the middle of the frequency range and the center frequency.

Be aware that HFSS uses the finite element mesh refined during an adaptive solution at the solution frequency or, if you did not request an adaptive solution, the initial mesh generated for the problem. It uses this mesh without further refinement. Also, the field solution at the center frequency is the most accurate. Depending upon the desired level of accuracy you require throughout the frequency range, you may wish to perform additional Fast sweeps at other center frequencies.

The full-field solution is saved only at the center frequency, while the S-parameters are saved for every frequency point; however, the Fast sweep allows the you to post process fields for any frequency entries to the sweep range.

The time and memory required for a Fast sweep may be significantly greater than the time and memory required for a single frequency solution.

The procedure for a Fast frequency sweep is shown below.



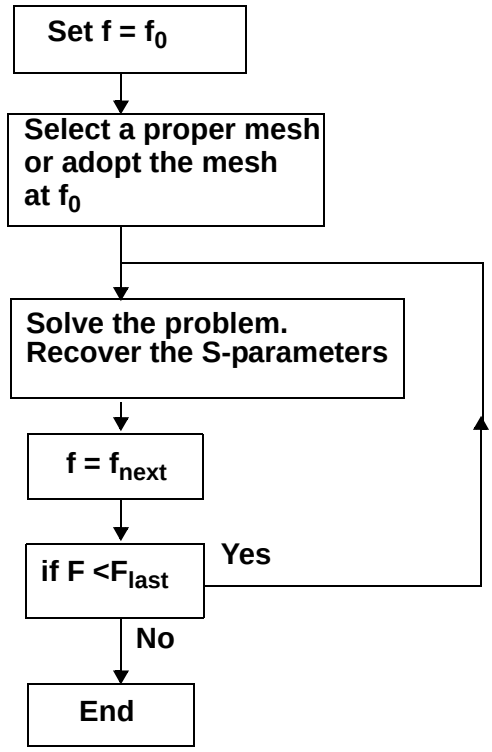
Discrete Frequency Sweeps

A Discrete sweep generates field solutions at specific frequency points in a frequency range. For example, if you specify a range of 1000 MHz to 2000 MHz, then a Step Size of 2.5, the result would be solutions at 1000, 1250, 1500, 1750, and 2000 MHz. By default, the field solution is only saved for the final frequency point computed, which would be at 2000 MHz in this case. Select the **Save Fields** option when setting up the points to solve if you want to save the field solution for a specific point. The S-parameters are saved for every frequency point. The more steps you request, the longer it takes to complete the frequency sweep.

Choose a Discrete sweep if only a few frequency points are necessary to accurately represent the results in a frequency range.

Be aware that HFSS uses the finite element mesh refined during an adaptive solution at the solution frequency or, if you did not request an adaptive solution, the initial mesh generated for the problem. It uses this mesh without further refinement. Because the mesh for the adaptive solution is optimized only for the solution frequency, it is possible that the accuracy of the results could vary at frequencies significantly far away from this frequency. If you wish to minimize the variance, you can opt to use the center of the frequency range as the solution frequency. Then, after inspecting the results, run additional solutions with the solution frequency set to the critical frequencies.

The procedure for a Discrete frequency sweep is shown below, where *n* equally spaced frequencies are included in the sweep.



Interpolating Frequency Sweeps

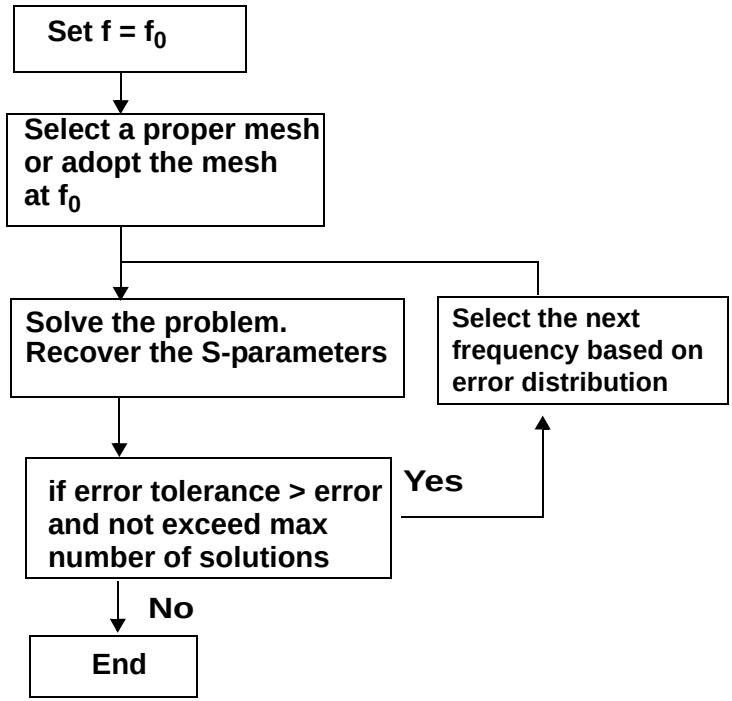
An Interpolating sweep estimates a solution for an entire frequency range. HFSS chooses the frequency points at which to solve the field solution so that the entire interpolated solution lies within a specified error tolerance. The sweep is complete when the solution meets the error tolerance criterion or generates the maximum number of solutions. To view more information about the solution, increase the number of steps and perform the sweep again.

The field solution for each point is deleted so that a new field solution can be generated for the next point. The full-field solution is only saved for the final frequency point computed. The S-parameters are saved for every solved frequency point.

Choose an Interpolating sweep if the frequency range is wide and the frequency response is smooth with the exception of a few resonances, or if the memory requirements of a Fast sweep exceed your resources. An Interpolating sweep's time requirement may be much less than a Discrete sweep's because a solution for the entire frequency range is interpolated based on solutions for a minimal number of frequency points. The maximum time required for an Interpolating sweep is the time required for a single frequency solution multiplied by the maximum number of solutions.

Be aware that HFSS uses the finite element mesh refined during an adaptive solution at the solution frequency or, if you did not request an adaptive solution, the initial mesh generated for the problem. It uses this mesh without further refinement.

The procedure for an Interpolating sweep is shown below, where *n* frequencies determined by the system are included in the sweep.



Solution Types

Driven Modal Solution

Choose the **Driven Modal** solution type when you want HFSS to calculate the modal-based S-parameters of passive, high-frequency structures such as microstrips, waveguides, and transmission lines. The S-matrix solutions will be expressed in terms of the incident and reflected powers of waveguide modes. **Network Analysis** is the default and functions as before.

Composite Excitation provides a method for solving fields in a large frequency domain problem.

Driven Terminal Solution

Choose the **Driven Terminal** solution type when you want HFSS to calculate the terminal-based S-parameters of single and multi-conductor transmission line ports. The S-matrix solutions will be expressed in terms of voltages and currents on the terminals. **Network Analysis** is the default and functions as before.

Composite Excitation provides a method for solving fields in a large frequency domain problem.

Eigenmode Solution

Choose the Eigenmode solution type to calculate the eigenmodes, or resonances, of a structure. The Eigenmode solver finds the resonant frequencies of the structure and the fields at those resonant frequencies.

Transient Solution

For calculating problems in the time domain. It employs a time-domain ("transient") solver. Selecting **Transient** enables radio buttons for **Network Analysis** or **Composite Excitation**.

Transient Network Analysis solution types - excitations are identical on all active ports and are simulated one at a time to facilitate the data collection.Your choice affects the options for the setup. If you select **Network Analysis** the setup includes an Input Signal tab for the simulation.

Composite Excitation solution types - Different ports can have different excitations. All Active excitations are launched in one simulation.

Typical Transient applications include, but are not limited to: Simulations with pulsed excitations, such as ultra-wideband antennas, lightning strikes, electro-static discharge; field visualization employing short-duration excitations; time-domain reflectometry.

Eigenmode Solutions

The Eigenmode solver can find the Eigenmodes of lossy as well as lossless structures, and can calculate the unloaded Q of a cavity. Q is the quality factor, and is a measure of how much energy is lost in the system. Unloaded Q is the energy lost due to lossy materials and boundary conditions. Because ports and other sources are restricted for eigenmode problems, the Q calculated does not include losses due to those sources.

The following restrictions apply to Eigenmode solution designs:

- No excitations may be defined.
- Radiation boundaries may not be defined.
- Frequency sweeps are not available.
- You may not view or plot the S-matrix data.
- Designs cannot include ferrite materials.

If a material or boundary condition, other than finite conductivity, varies with frequency, the eigenmode solution procedure will evaluate the frequency dependent property at the user specified minimum frequency and use that value during the S and T matrix assembly. This may limit the accuracy of the solved Eigenmodes depending on how significant the property varies from the minimum frequency to the eigen frequency.

Related Topics

Calculating the Resonant Frequency

Calculating the Quality Factor

Q Factor of Resonant Cavities

Calculating the Resonant Frequency

Eigenmodes are the resonances of the structure. The eigenmode solver finds the resonant frequencies of the structure and the fields at those resonant frequencies. For a Driven Solution, HFSS solves the following matrix equation (for a lossless case):

$$Sx + k_o^2 Tx = b \quad (1)$$

where

- S and T are matrices that depend on the geometry, the materials, and the mesh.
- $\boldsymbol{x}$ is the electric field solution.
- k_o is the free-space wave number.
- b is the value of the source defined for the problem.

However, in order to find the resonances of the structure, the eigenmode solver sets b to zero, and solves the equation

$$Sx + k_o^2 Tx = 0 \quad (2)$$

for sets of (ko,$\boldsymbol{x}$), one ko for every $\boldsymbol{x}$. The variable x is still the electric field solution, and ko is the free space wave number corresponding to that mode. The resonant frequency of the eigenmode can be found using:

,

$$f = \frac{k_o c}{2\pi} \quad (3)$$

where

- c is the speed of light.

Calculating the Quality Factor

Q is the unloaded quality factor, and is a measure of how much energy is lost in the structure due to lossy materials. Because ports and other sources are restricted for Eigenmode solutions, the Q calculated does not include losses due to those sources.

HFSS uses the following equation to calculate the approximate quality factor:

$$Q = \left| \frac{Mag(freq)}{2 \cdot Im(freq)} \right| \quad (1)$$

The Fields Calculator can also be used to calculate Q. In general, the equation for Q is

$$Q = (2\pi)(freq)\frac{U}{P} \quad (2)$$

where:

- U is the total energy stored in the cavity.
- P is the power lost, from resistive losses, for example.

Q Factor of Resonant Cavities

Basic Definition

$$Q = \frac{\omega W}{P_{loss}} \qquad \omega = |\omega| \tag{1}$$

In complex case (see examples in textbooks)

Since P_{loss} comes form W

$$Q_{min} = \frac{1}{2} \tag{2}$$

Calculating from complex resonance Frequency

$$\omega = \omega_r + j\omega_i \tag{3}$$

Definition 1 (Correct)

$$Q = \frac{|\omega|}{2\omega_i} \tag{4}$$

Definition 2 (Incorrect)

$$Q = \frac{\omega_r}{2\omega_i} \tag{5}$$

Arguments for using Definition 1

- Collin always uses the right frequency
- Gardiole [1] always uses definition1.
- IEEE papers [2], [3], and [5] use Definition 1.

Example R,L,C Circuit

$$R + j\omega L + \frac{1}{j\omega C} = 0 \tag{6}$$

$$\omega_{res} = \frac{1}{2}\left(j\frac{R}{L} + \sqrt{\frac{4}{LC} - \left(\frac{R}{L}\right)^2}\right) \tag{7}$$

Definition 1 (Correct)

$$Q = \frac{\omega_o L}{R} \qquad \omega_o = \frac{1}{\sqrt{LC}} \tag{8}$$

Definition 2 (Incorrect)

$$Q = \frac{\sqrt{\frac{4}{LC} - \left(\frac{R}{L}\right)^2}}{2\frac{R}{L}} \tag{9}$$

Example Cube Resonator

$$f_{res} = \frac{1}{2\pi} j\frac{\sigma}{2\varepsilon} + \sqrt{-\left(\frac{\sigma}{2\varepsilon}\right)^2 + \frac{1}{\mu\varepsilon}k^2} \quad (10)$$

Definition 1 (Correct)

$$Q = \frac{\omega_o \varepsilon}{\sigma} \qquad \omega_o = \frac{k}{\sqrt{\mu\varepsilon}} \quad (11)$$

Definition 2 (Incorrect)

$$Q = 0.5\frac{\sqrt{\frac{1}{\mu\varepsilon}k^2 - \left(\frac{\sigma}{2\varepsilon}\right)^2}}{\frac{\sigma}{2\varepsilon}} \quad (12)$$

- Both definitions give almost the same answer at low losses
- Definition 2 gives nonphysical values at high losses, so Definition 1 is the correct one

References

[1] Gardiol, F.E, Introduction to Microwaves, Artech House, 1984, pp. 113-115.

[2] M. Tsuji, H. Shigesawa and K. Takiyama, " On the Complex Resonant Frequency of Open Dielectric Resonators, " IEEE Transactions on Microwave Theory and Techniques,Vol. MTT-31, No. 5, May 1983. pp. 392-396. Eq. (6)

[3] IEEE MTT Vol. 30, No. 11, November 1982. pp 1954.

[4] R.T Remsky, "Ansoft HFSS8 Solution of TEAM Workshop Problem 18 (Waveguide Loaded Cavity) Using Both Eigenmode and Driven Solutions Presented at COMPUMAG 2001, Evian, France. Ansoft, Intranet.

[5] C.R. I. Emson and A Ait-Sadi, "Results for TEAM Problem 18 - Waveguide With Iris Proceedings of the TEAM Workshop in the Sixth Round, March 20-21, 1996, Okayama, Japan.

[6] S. Groiss, I. Bardi, O. Biro, K. Preis and K.R. Richter, "Solution of the TEAM Benchmark Problem 18, Waveguide Loaded Cavity" Proceedings of the TEAM Workshop, July 14, 1995, Berlin, Germany.

[7] S. Groiss, I. Bardi, O. Biro, K. Preis and K.R. Richter, "Solution of the TEAM Benchmark Problem 19, Circular Cavity Resonator" Proceedings of the TEAM Workshop in the Sixth Round, March 20-21, 1996, Okayama, Japan.

[8] Bardi et al, "Solution of TEAM Benchmark Problem 18 - Waveguide Loaded Cavity," Proceedings of TEAM Workshop, Aix-les- Bains, France, July, 1994, pp. 3-5.

[9] Bardi, Zs. Badics, "Solution of TEAM Workshop Problem 18 by Maxwell Eminence,

Presented at TEAM Workshop meeting held after COMPUMAG'98, Rio, Brazil.

[10] I. Bardi, "Eigenmode and Q factor analysis of a waveguide loaded cylindrical cavity resonator" Ansoft, Intranet

[11] I Bardi, Z. Cendes, "Rigorous analysis of cavity resonators by the finite element method" Ansoft, Intranet

[12] R. E. Collin, "Foundations for Microwave Engineering ", Second Edition, McGraw Hill, Inc

[13] W. Mueller, R. Schuhmann and T. Weiland, "Calculatin Trapped Modes in Tesla Cavities with Time and Frequency Domain Methods" Proceedings of the 2001 Particle Accelerator Conference, Chicago.

Field Solutions

During the iterative, adaptive solution process, the S-parameters typically stabilize before the full Electric field solution. Therefore, when you are interested in analyzing the field solution or quantities computed from the field solutions associated with a structure, it is advisable to use tighter convergence criteria.

In addition, for any given number of adaptive iterations, the magnetic field (H-field) is less accurate than the solution for the electric field (E-field) because the H-field is computed from the E-field using the relationship: .

$$\boldsymbol{H} = \frac{\nabla \times \boldsymbol{E}}{-j\omega\mu}$$

Field Overlay Plots

In HFSS, field overlays are representations of basic or derived field quantities on surfaces or objects. The objects on which you plot the fields may be pre-existing parts of the model geometry or they may be objects that you draw in post-processing mode.

If you select a surface, HFSS will plot the field quantities on the surface. If you select an object, HFSS will plot the field quantities within the volume of the object.

You can choose to create a scalar plot or a vector plot of the fields. A scalar plot can use several user selectable plot types to illustrate the magnitude of field quantities on surfaces or volumes. A vector plot uses arrows to illustrate the magnitudes of the x-, y-, and z-components of field quantities.

Field Quantities

The default field quantities that can be plotted, their definitions, and associated units are as follows (see the following note for issues in HFSS-IE for plotting magnetic and electric current):

Field Quantity	Definition	Units
Mag E	The magnitude of the electric field, |E|(x,y,z,t).	V/m
Mag H	The magnitude of the magnetic field, |H|(x,y,z,t).	Amps/m
Mag Jvol	The magnitude of the current density, |J|(x,y,z,t), over the volume.	$Amps/m^2$

Mag Jsurf	The magnitude of the current density, |J|(x,y,z,t), on the surface.	Amps/m
Complex Mag E	The complex magnitude of the electric field, |E|(x,y,z).	V/m
Complex Mag H	The complex magnitude of the magnetic field, |H|(x,y,z).	Amps/m
Complex Mag Jvol	The complex magnitude of the current density, |J|(x,y,z), over the volume.	$Amps/m^2$
Complex Mag Jsurf	The complex magnitude of the current density, |☺|(x,y,z), on the surface.	Amps/m
Vector E	The electric field, E(x,y,z,t).	V/m
Vector H	The magnetic field, H(x,y,z,t).	Amps/m
Vector Jvol	The current density, J(x,y,z), over the volume.	$Amps/m^2$
Vector Jsurf	The current density, J(x,y,z), on the surface.	Amps/m
Vector Real Poynting	The Poynting vector, defined as E x H*.	W/m^2
Local SAR	The specific absorption rate.	W/kg
Average SAR	The average specific absorption rate.	W/kg
Certification SAR	The IEEE specific absorption rate certification number.	W/kg

Note

Plotting Magnetic current in HFSS-IE is relatively simple. It is M = Exn on an interface between two non-conducting materials. This is an equivalent current.

Plotting Electric current in HFSS-IE involves three cases:

1. J = nxH on a surface of solid conductor (H is zero inside the conductor). This is a TRUE physical current.
2. J = nx(H1-H2) on a conducting sheet which is inside a region/background (or it is not an interface between two regions). This is a TRUE physical current.
3. J = nxH1 on a conducting boundary between two regions (it doesn't matter if the two regions have the same material). H1 is the field on the positive side of the surface. However, if region 2 is the background, we will display J = nxH2. This is an equivalent current.

Specifying the Phase Angle

Specifying the phase angle at which the field quantity is calculated enables you to compute the real part of the field's magnitude at different points in its cycle. These quantities can be represented in the form

$$\boldsymbol{A}(x, y, z, t) = \boldsymbol{A}(x, y, z)\cos(\omega t + \theta)$$

where

- ω is the angular frequency specified during the solution.
- θ is the phase angle (the offset from a cosine wave that peaks).

Peak Versus RMS Phasors

This section concerns how field quantities are represented within HFSS. Some users will not need this information, such as those who wish to know port S-parameters or relative amplitudes of field solutions. Those that wish to find absolute field values, for example, will need to review the difference between the two types of field representation, peak and RMS.

HFSS solves in the frequency domain and obtains a phasor representation of the steady-state finite element field solution. Physical quantities such as the instantaneous (time domain) electric field are then obtained as derived quantities from the phasor representation.

If *Ex* is the x-component of a "peak" phasor quantity representing a time-harmonic electric field, the physical electric field x-component at time *t*, denoted $E_x(t)$, is computed from

$$E_x(t) = \Re\left[E_x e^{j\omega t}\right] \quad (1)$$

where

- $\Re$ is the real part of a complex number or function.
- ω is angular frequency, 2πf.
- *j* is the imaginary unit, $\sqrt{-1}$.
- *t* is the time.

On the other hand, if $\mathbf{E}_x$ is an "RMS" phasor, an additional factor of $\sqrt{2}$ is required as follows:

$$\boldsymbol{E}_x(t) = \Re\left[\sqrt{2}E_x e^{j\omega t}\right] \quad (2)$$

As a consequence of these equations, the peak physical field, max ($E_x(t)$) observed over a full time cycle is $max(\boldsymbol{E_x}(t)) = |E_x|$ for peak phasors and $max(\boldsymbol{E_x}(t)) = \sqrt{2}\,|E_x|$ for RMS phasors.

Additionally, given field phasors E and H, to compute the time-averaged power flow through a surface, the normal component of the real part of the complex Poynting vector is integrated over the surface. The correct form of the complex Poynting vector *S* depends on which phasor representation is used.

For peak phasors, $S = \frac{1}{2}E \times H^*$.

For RMS phasors, $E = E \times H^*$.

The conventions used by HFSS are as follows:

- Each propagating mode incident on a port contains 1 watt of time-averaged power.
- Circuit gap sources are specified in a peak sense. That is, if a voltage gap source magnitude is 5 volts, then the time domain circuit source behaves as v(t) = 5cosωt. Likewise for a current gap source.

- Plane wave sources are specified in a peak sense. That is, if the plane wave magnitude is 5 V/m, then the plane wave incident field magnitude is $E(t) = 5\cos(\boldsymbol{k} \cdot \boldsymbol{r} + \omega t)$.
- Radiated power, as computed by the fields post processor, is a time-averaged quantity computed using the complex Poynting vector.
- Phasors in the Fields Calculator are peak phasors. The Poynting vector button in the calculator therefore implements the Poynting vector for peak phasors,

  $$S = \frac{1}{2} E \times H^* \quad .$$

  Calculations that compute either average or instantaneous time domain quantities must adhere to the peak phasor conventions.

Calculating the SAR

The specific absorption rate (SAR) is a measure of the amount of electromagnetic energy absorbed in a lossy dielectric material. The SAR is a basic scalar field quantity that can be plotted on surfaces or within objects in HFSS. The **HFSS>Fields>SAR Setting...** command lets you specify the SAR setting. You have a choice of IEEE Standard P1528.4 and Gridless.

For the IEEE STD P1528.4 we subdivide the model space into a grid of voxels properly aligned along the global coordinate system. The density and local SAR values are then mapped over the grid of voxels. Afterwards the two-pass average SAR algorithm is applied to the voxels. Since this two pass algorithm is very time consuming we try to use parallel processing as much as possible. The algorithm is divided into two phases so that the results of the time consuming phase can be reused when fields change, i.e. new combinations from edit sources.

For more details, see IEEE P1528.4™/D1.0 Recommended Practice for Determining the Peak Spatial Average Specific Absorption Rate (SAR) in the Human Body from Wireless Communications Devices, 30 MHz - 6 GHz: Requirements for Using the Finite-Element Method for SAR Calculations, specifically involving Vehicle Mounted Antennas and Personal Wireless Devices.

For Gridless, HFSS uses the following equation to calculate the SAR: $\sigma * E^2/(2\rho)$.

where

- σ = the material's conductivity. This is defined as: $\sigma_{bulk} + \omega\varepsilon_0\varepsilon_r tg\delta$
- ρ = the mass density of the dielectric material in mass/unit volume.

There are two types of SAR Field Overlay plots available in HFSS: local SAR, and average SAR. When calculating the local SAR, HFSS uses the equation above to calculate the SAR at each mesh point on an overlay plot. HFSS interpolates the values between the mesh points across the plot.

When plotting the average SAR, for each mesh point on the plot, HFSS reports the SAR averaged over a volume that surrounds that point. The volume is determined by the settings for the material's mass density and mass of the material surrounding each mesh point set in the S**pecific Absorption Rate Setting** dialog box. When you use the Gridless method, the volume will not cross object boundaries. The IEEE 1528 method performs averaging over all tissues.

The Certification SAR quantity provided in the Fields Calculator is an efficient way to determine the peak spatial-average SAR. When computing the certification SAR, HFSS assumes the peak spatial-average SAR is located on an object surface. HFSS therefore searches for the peak local

SAR on the surfaces, and creates an averaging volume around that location, completely inside the object with an axis normal to the surface at the location of the peak value of the local SAR.

Related Topics

Modifying SAR Settings

Transient Solution Theory

An important advantage of the Finite-Element Method (FEM) is its ability to solve Maxwell's equations on an unstructured mesh [1]. In such a mesh, the elements have a range of sizes and orientations such that, rather than forming a regular grid, they conform to the geometry of the model, respecting all its details. The use of unstructured meshes, combined with automatic mesh adaptation in HFSS Frequency Domain, has been instrumental in delivering accurate results for complicated geometries.

So far, commercially available solvers based on time-domain methods have relied on the Finite Difference Time Domain method (FDTD) [2] or on the closely related Finite Integration Technique (FIT) [3]. The use of unstructured meshes for both schemes presents technical difficulties that have yet to be overcome, and thus both FDTD and FIT continue to rely on brick-based space partitioning.

Time-domain methods developed for unstructured meshes such as the Finite Volume Time Domain (FVTD) [4, 5] or the Finite Element Time Domain (FETD) [6] methods do not offer an acceptable trade-off between accuracy, speed and memory usage. However, recent advances in finite-element techniques have led to a new scheme based on the Discontinuous Galerkin [7] family of numerical methods. This scheme retains the flexibility, accuracy and reliability of unstructured-mesh finite-element methods but avoids the solution of a large matrix equation at each time step. Instead, in the Discontinuous Galerkin Time Domain (DGTD) method, each mesh element advances in time using its own time step in a synchronous manner. This results in a significant speed-up on unstructured meshes as the largest mesh elements typically take time steps that are two to five orders of magnitude larger than those of the smallest mesh elements.

This chapter provides an explanation of how the DGTD method has been implemented in HFSS Transient.

Discontinuous Galerkin Finite Element Method

The DGTD solver provides the time-domain solution of Maxwell's Equations.

$$\frac{\partial \vec{D}}{\partial t} + \vec{J} = curl(\vec{H}) \qquad \frac{\partial \vec{B}}{\partial t} = -curl(\vec{E}) \qquad (1)$$

$$div(\vec{D}) = \rho \qquad div(\vec{B}) = 0$$

The mathematical framework of the Discontinuous Galerkin family of numerical methods for Maxwell's equations has been presented by Jan Hesthaven and Tim Warburton [8, 9]. To apply the DG method, we choose basis and test functions local to each mesh element *T*, in a similar way to the Finite Element Method:

$$\int_T \frac{\partial \vec{D}}{\partial t} \cdot \vec{\varphi} = \int_T curl(\vec{H}) \cdot \vec{\varphi} - \int_T \vec{J} \cdot \vec{\varphi} = \int_T curl(\vec{\varphi}) \cdot \vec{H} - \int_T div(\vec{\varphi} \times \vec{H}) - \int_T \vec{J} \cdot \vec{\varphi} \qquad (2)$$

$$\int_T \frac{\partial \vec{B}}{\partial t} \cdot \vec{\varphi} = \int_T -curl(\vec{E}) \cdot \vec{\varphi} - \int_T curl(\vec{\varphi}) \cdot \vec{E} + \int_T div(\vec{\varphi} \times \vec{E})$$

As is the case with the frequency-domain finite-element solver, the DGTD method is free of spurious solutions [10] and supports higher-order basis functions. In the time domain, the use of high orders minimizes dispersion errors occurring during the simulation of electrically large problems. The DGTD solver not only employs high-order elements but allows mixed element orders in the same mesh, known as hp-adaptivity, to optimize accuracy and solution times.

Following the same approach as in the Finite Volume Method, the divergence term containing volume integrals is transformed into surface integrals:

$$\int_T \frac{\partial \vec{D}}{\partial t} \cdot \vec{\varphi} = \int_T curl(\vec{\varphi}) \cdot \vec{H} + \int_{\partial T} (\vec{n} \times \vec{H}) \cdot \vec{\varphi} - \int_T \vec{J} \cdot \vec{\varphi} \qquad (3)$$

$$\int_T \frac{\partial \vec{B}}{\partial t} \cdot \vec{\varphi} = -\int_T curl(\vec{\varphi}) \cdot \vec{E} - \int_{\partial T} (\vec{n} \times \vec{E}) \cdot \vec{\varphi}$$

The tangential fields $\vec{n} \times \vec{E}$ and $\vec{n} \times \vec{H}$ are then evaluated as upwind fluxes at the surface of *T* by using a Riemann solver [11]. Separating the treatment inside the element from its surface provides a robust and flexible framework to take into account boundary conditions. This separation also enables various combinations of locally implicit and locally explicit time-integration schemes for instance to achieve an optimum trade-off between speed and memory, without compromising accuracy and stability.

Related Topics

Transient Solution Theory

Local Time Stepping

Materials in Time Domain

Excitations in Time Domain

Analysis Setup in Time Domain

References for Time Domain

Local Time Stepping

A prime benefit of satisfying Maxwell's equations in a way analogous to the Finite Volume Method is greatly reduced RAM usage. Since the DGTD solver avoids solving the large matrix equation that represents the entire computational domain at each time step, no large matrix needs to be stored.

One of the key features of the DGTD solver is an innovative local time-stepping scheme that advances in time each mesh element independently from its neighbors. This feature efficiently exploits the use of highly unstructured meshes and hp-adaptivity: in the DGTD solver, each mesh element has its own energy-based stability criterion [12], calculated from its size, material properties and basis-function type and order.

Figure 1 illustrates the process. This figure shows a cut-plane through an unstructured mesh model of a ground-penetrating radar. Each element of the mesh is colored according to the time-step values used to locally advance the fields in time. Large elements are colored yellow and use larger time-steps than the smaller elements colored green and or even smaller elements in blue. This results in fewer iterations for most mesh elements and reduced overall simulation times.

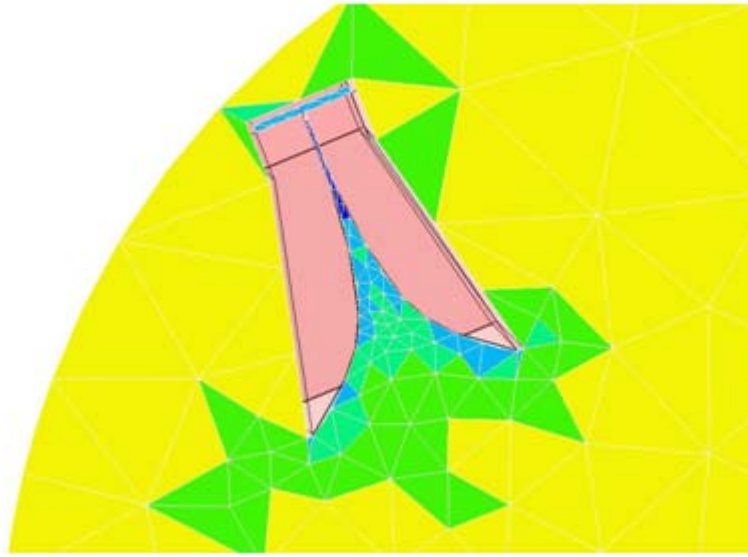


Fig. 1 Local time stepping

The time step of any mesh element is chosen such that at the largest time steps the entire simulation is "in sync." From a user point of view however, the local time-stepping scheme is transparent and all time-domain data are provided or rendered synchronously "on the fly" as needed. In contrast to FIT or FDTD, the DGTD solver doesn't advance in time uniformly using the smallest time step but uses an optimal range of time steps that minimize simulation times, ensures stability and minimizes the dispersion error.

One last important feature of DGTD solver is its affinity for parallel computations. This is a direct consequence of the local nature of its formulation that yields a spatially compact numerical scheme. Parallelism is naturally exploited with this scheme by partitioning the computational domain into sub-domains and solving each sub-domain concurrently on multi-core or many-core systems.

Related Topics

Transient Solution Theory

Discontinuous Galerkin Finite Element Method

Materials in Time Domain

In any time-domain method, handling frequency-dependent material parameters is not straightforward. Two important cases, frequently encountered in electromagnetic simulations, are (1) non-perfectly conducting metals and (2) lossy dielectric materials.

In most high-frequency applications, one does not solve for fields inside non-perfectly conducting metals. Rather, the skin effect is expressed with a boundary condition on the surfaces of the metal. Since the skin impedance is frequency dependent, containing a factor √(frequency), a suitable way has to be found to approximate this in the time domain. In HFSS Transient, the Padé approximation [ ] is used. With this approximation one can expect accurate results over a very wide band which includes the highest frequency of interest. Near DC, a modest reduction in accuracy is to be expected.

Lossy dielectric materials can be described by a bulk conductivity σ or by a loss tangent tan(δ). If the conductivity is the only loss mechanism, the relation between the two is $\tan(\delta) = \sigma/(2\pi \times \text{frequency} \times \varepsilon_0 \varepsilon_r)$. In any case, the loss tangent of a material is frequency dependent, and a suitable way has to be found to approximate it in the time domain. In HFSS Transient, when a lossy dielectric is specified with a certain loss tangent at the highest frequency of interest, the Debye model is used to describe the material for all frequencies. The Debye model is more realistic than the assumption of a constant loss tangent. If one uses the Debye model in the frequency domain as well, one can expect the same results as in the time domain, except when the material touches a port. That case is discussed in the next section.

Related Topics

Transient Solution Theory
Discontinuous Galerkin Finite Element Method
Local Time Stepping
Excitations in Time Domain
Analysis Setup in Time Domain
References for Time Domain

Excitations in Time Domain

Excitations available in HFSS Transient are wave ports, lumped ports, voltage sources, current sources and incident waves. In the case of ports, the modal port solution is provided by the same 2D port solver as is used in HFSS Frequency Domain. If a lossy dielectric or a non-perfectly conducting metal touches the port, the port impedance will be frequency dependent. A frequency-dependent excitation poses a problem to any time-domain solver. HFSS Transient, as well as some competing time-domain codes, uses the mode belonging to the lossless case as an excitation in the port.

In order to deal with higher-order modes or with frequency-dependent (lossless) modes, e.g. in wave guides, HFSS Transient takes advantage of the trans-finite element method [ ], which has always been an important technique in HFSS Frequency Domain. In HFSS Transient, elements touching the ports are set aside in an implicit time-stepping scheme that is close to the FETD method. Since this is a small fraction of the total number of elements, the penalty for solving the matrix equation is negligible, while the implicit method allows large time steps. The upshot is that higher-order modes and frequency-dependent (lossless) modes are handled accurately.

In a standard HFSS simulation a port excitation is the equivalent of an excited infinite transmission line connected to the port face. For a modal project this is easy to visualize but in a terminal project the user defines the input voltage. This raises the question of how this input voltage is applied to the port for terminal designs. While the following discussion is directed to transient simulations, it is also applicable to HFSS terminal designs where the Incident Voltage option in Edit Sources is used.

To derive the correct model for a terminal port, first consider the transmission line section shown in Figure 1. The port is at the point 0-0 and to the left is the equivalent infinitely long feed line.

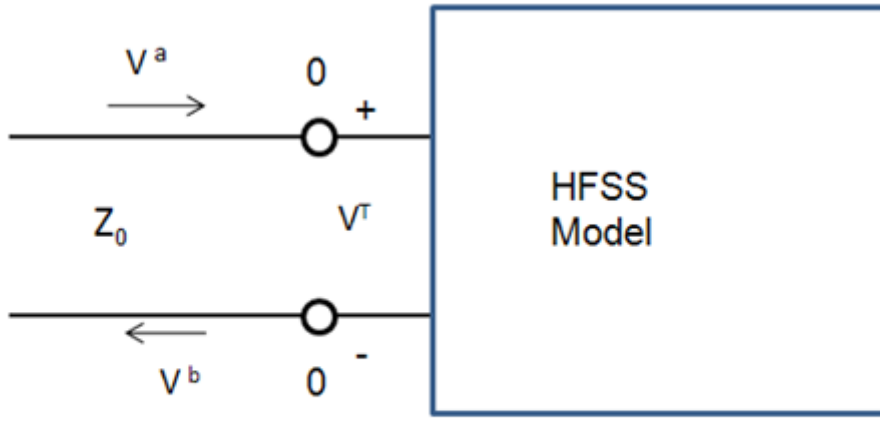


Figure 1

For this model then V^a is the incident voltage and V^b is the reflected voltage seen at the port. We know from the definition of voltage and current on a transmission line that for the total voltage, V^T, the following is true:

$$V^T = V^a + V^b \quad (1)$$

If the characteristic impedance of the line is equal to Z0 then we also know:

$$Z_0 I = V^a - V^b \quad (2)$$

If we add equations (1) and (2) we find that at point 0-0:

$$V^T + Z_0 I = 2V^a \quad (3)$$

If we apply Kicrchkoff's voltage law to the equivalent circuit shown in Figure 2 we see it yields the result shown in (3). This is the port model for a terminal port in HFSS.

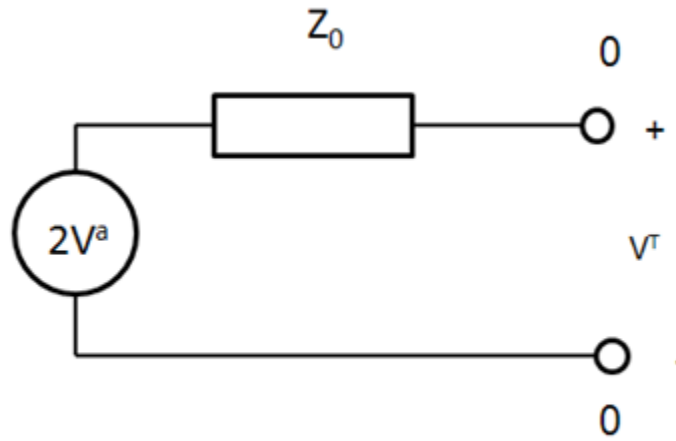


Figure 2

To be consistent then the same terminal port excitation is used in HFSS Transient. Therefore if you define an incident voltage signal in the transient setup that signal is applied to the 3D model as a voltage source of twice the user defined amplitude in series with the terminal impedance.

Note As a consequence of the above discussion, one needs to be careful when comparing results with a circuit tool or textbooks using the voltage source shown in Figure 2. If one volt is used for the voltage source in the circuit simulator, this should be compared with a HFSS transient simulation with edit sources set to 0.5 volt.

Before considering an example one more point needs to be made. In the reporter for HFSS Transient when plotting the voltage at a given port the options are Input( ) and Output( ). Input is the user defined incident voltage (Va in the Figure 2) and Output is the reflected voltage (Vb = VT - Va). We can see then that to view the total voltage at the port terminals one needs to create an output variable that is the sum of Input ( ) and Output( ). We will consider two examples to illustrate this.

Consider the HFSS Transient model of a simple coax shown in the Figure 3. It is an air filled 50Ω line that has a center section with a smaller inner radius that results in an impedance of 91.5 ohm in that part of the line. Each section of this air filled coax (the two 50 Ω and the one 91.5 ohm sec-

tions) has a length of 100mm. This length has a time delay of 1/3 nS. This was simulated in HFSS Transient using the incident voltage signal shown in the inset for Figure 3.

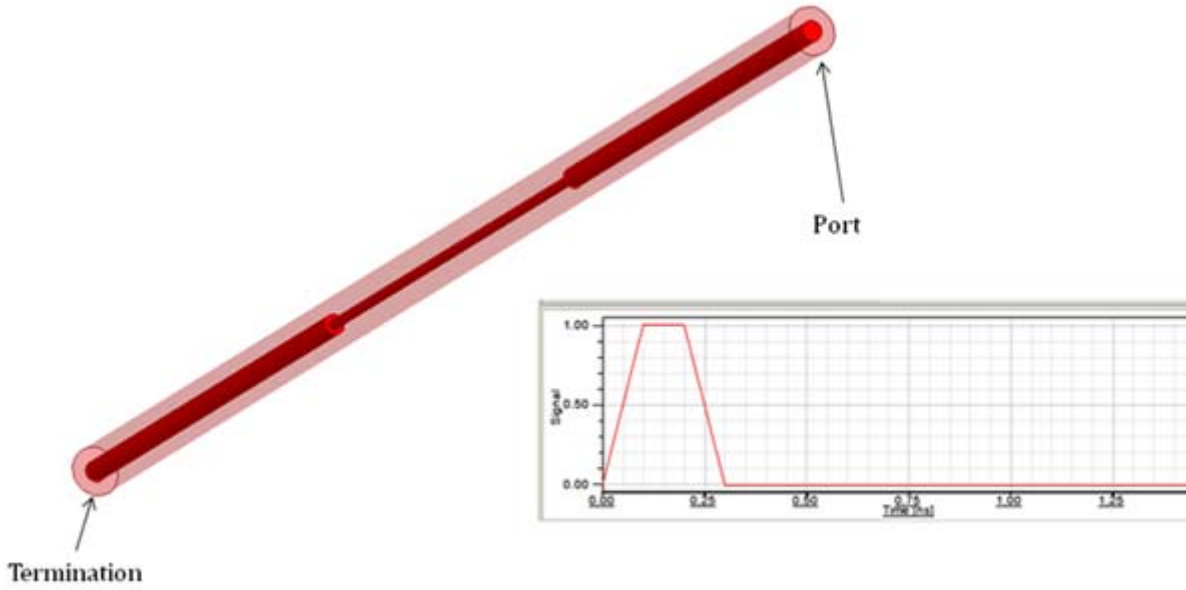


Figure 3

This model was simulated with a single 50 Ω terminal port on one end and a matched termination on the opposite end. The response seen at the port is shown below in Figure 4.

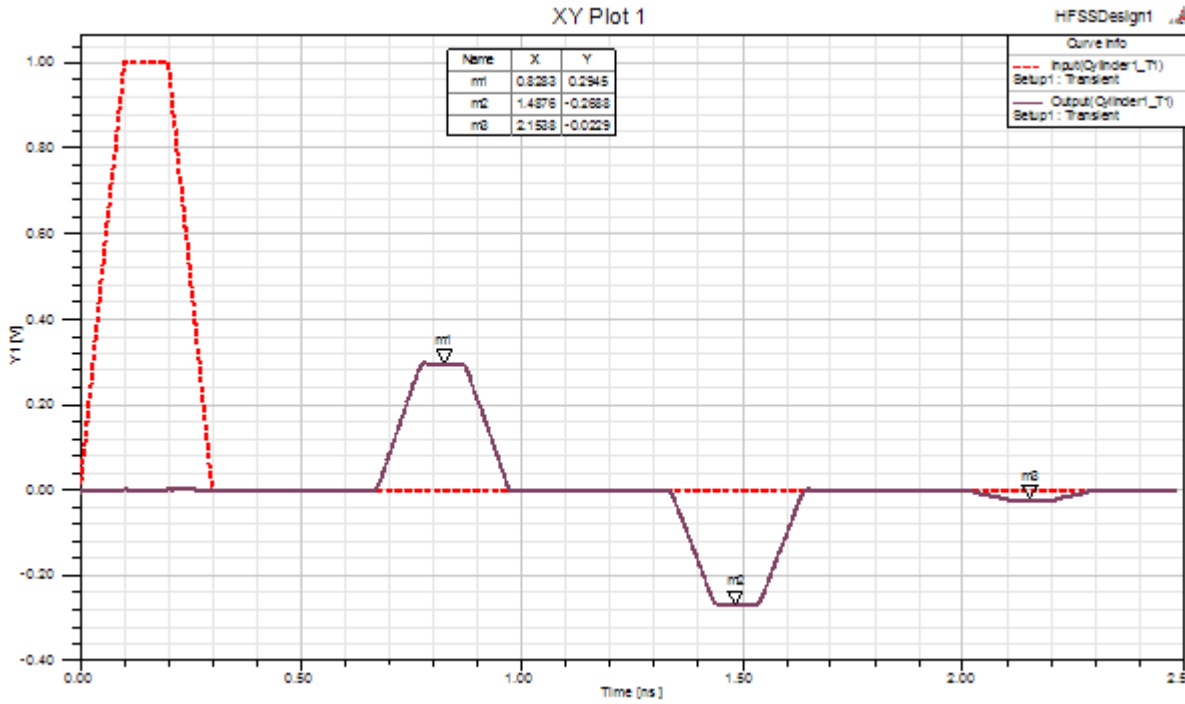


Figure 4

Figure 4 shows the Input and Output signals at the port terminal (Cylinder1_T1). Remember that Input( ) is the incident signal as shown in Figure 3. If one uses the equations for reflection and transmission to compute the amplitude of the signal at the 3 points shown in the figure these text-book equations would give 0.29, -0.27,-.02 respectively. We can see that the reflected voltage is accurately computed for the applied signal.

For an example where one might be more interested in the total voltage a second case was consid-ered. The same HFSS Transient model was simulated with the same incident signal, but this time

the port terminal impedance was set to 30 Ω. The HFSS Transient response for this simulation is shown in the Figure 5.

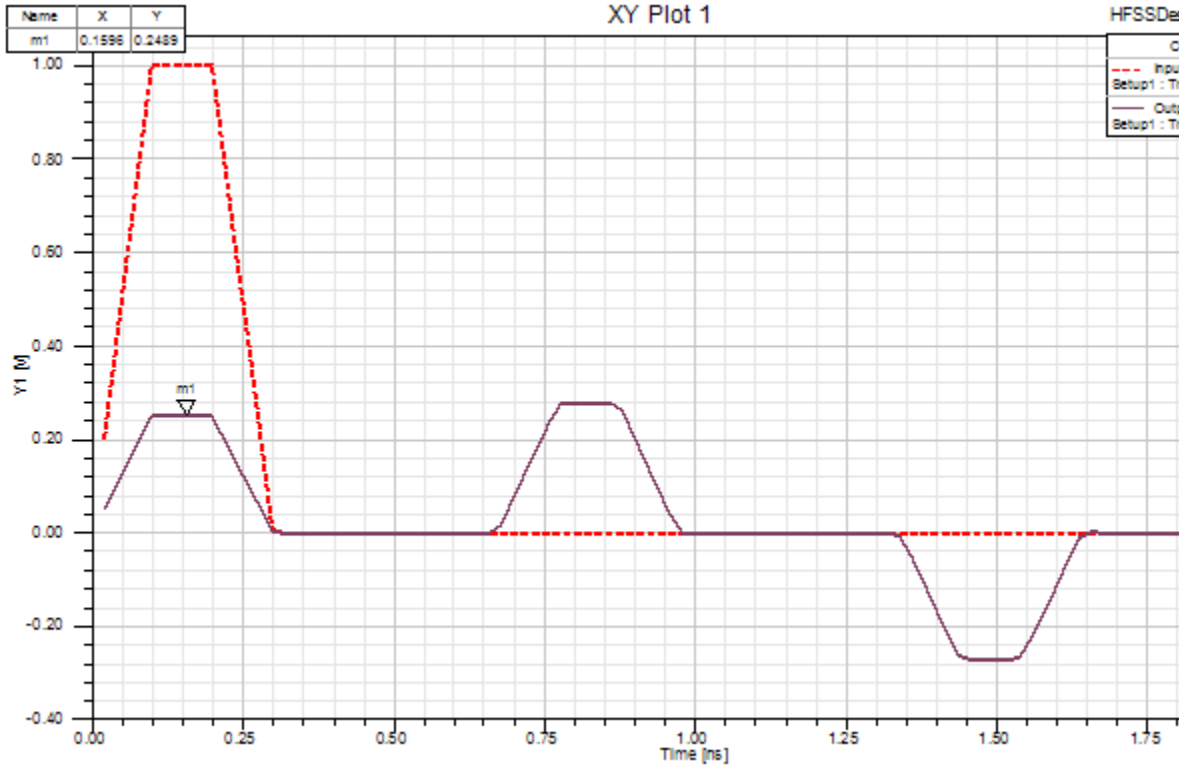


Figure 5

For the time between 0.1 - 0.2 nS the textbook response says that V^T should be 1.25V. Again that is the total voltage and if you add together the 2 curves shown in the plot you would get the correct response of 1.25V in that range.

Related Topics

Transient Solution Theory

Discontinuous Galerkin Finite Element Method

Local Time Stepping

Materials in Time Domain

Analysis Setup in Time Domain

References for Time Domain

Analysis Setup in Time Domain

Two classes of analyses exist: those focused on device characterization, and more general analyses. In the case of device characterization, one is interested in quantities like S-parameters and/or time-

domain output at ports due to excitation of a specific port. This requires that excitations be "on" one at a time. For each active port and mode, a separate simulation is done. These simulations are independent and can be distributed over a network, to be executed in parallel. Each active port and mode will employ an excitation with the same time profile.

In the more general analysis type, all active ports and modes are "on" simultaneously. Each one can have an excitation with a different time profile. This type of analysis gives you more freedom, but S-parameters cannot be extracted.

In both classes of simulations, fields can be saved and visualized.

Related Topics

Transient Solution Theory

Discontinuous Galerkin Finite Element Method

Local Time Stepping

Materials in Time Domain

Excitations in Time Domain

References for Time Domain

Target Residual for Transient Analysis Duration

The residual is defined to be the ratio of the global RMS value of the electric field at time t to the maximum RMS value in the time range zero to time t :

$$E_{rms}(t) = \sqrt{\frac{\sum_{i=1}^{N} \left|\vec{E}_i t\right|^2}{N}} \quad (1)$$

$$Residual(t) = \frac{E_{rms}(t)}{max\{E_{rms}(\tau):0 \le \tau \le t\}} \quad (2)$$

where $\vec{E}_i$ is the electric field at the ith interpolation point in the mesh which contains a total of N interpolation points. Note that the number of interpolation points is typically much larger than the number of mesh vertices. The time duration of a simulation can be controlled by specifying a target residual in db20 located in the Duration tab of the solution setup.

Related Topics

Transient Solution Theory

References for Time Domain

[1] Z. Cendes, D.N. Shenton, and H. Shahnasser. "Magnetic field computation using Delaunay triangulation and complementary finite element method", IEEE Trans. on Magnetics, 19(6):2551, 1983.

[2] A. Taflove, "Application of the Finite-Difference Time-Domain Method to Sinusoidal Steady-State Electromagnetic-Penetration Problems, Electromagnetic Compatibility", IEEE Transactions on Volume EMC-22, Issue 3, Aug. 1980 Page(s):191 - 202

[3] T. Weiland, "A discretization method for the solution of Maxwell's equations for six-component Fields", Electronics and Communications, vol. 31, no. 3, pp. 116-120, 1977

[4] P. Bonnet, X. Ferrières, B. Michielsen, and P. Klotz, "Time Domain Electromagnetics", S. M. Rao, Ed., Academic Press, 1997, ch. 9, pp. 307-367.

[5] D. Baumann, C. Fumeaux, and R. Vahldieck, "Field-Based Scattering-Matrix Extraction Scheme for the FVTD Method Exploiting a Flux-Splitting Algorithm," IEEE Trans. on Microwave Theory and Tech., vol. 53, no. 11, Nov. 2005.

[6] J-F. Lee, R. Lee, and A.C. Cangellaris, "Time-Domain Finite-Element Methods", IEEE Trans.Antennas Propag., vol. 45, pp. 430-442, Mar. 1997.

[7] B. Cockburn, G. Karniadakis, and C.-W Shu, "Discontinuous Galerkin Methods: Theory, Computation and Applications", Lecture Notes in Computational Science and Engineering (Springer-Verlag, New York, 2000), Vol. 11.

[8] J. Hesthaven,T. Warburton, "Nodal high-order methods on unstructured grids", Journal of Computational Physics, v.181 n.1, p.186-221, September 1 2002.

[9] J. Hesthaven and T. Warburton, "Nodal Discontinuous Galerkin Methods Algorithms, Analysis, and Applications", Springer series: Texts In Applied Mathematics, vol. 54, 2007.

[10] A. Buffa, I. Perugia, "Discontinuous Galerkin approximation of the Maxwell eigenproblem",SIAM J. Numer. Anal. 44, 2198, 2006.

[11] Leveque, Randall, "Finite Volume Methods for Hyperbolic Problems", Cambridge University Press, 2002.

[12] L. Fezoui, S. Lanteri, S. Lohrengel and S. Piperno, "Convergence and stability of a Discontinuous Galerkin time-domain method for the 3D heterogeneous Maxwell equations on unstructured meshes", ESAIM: Math. Model. and Numer. Anal., Vol. 39, No. 6, pp. 1149-1176, 2005.

[13] R.A. Chilton and R. Lee, "The discrete origin of FETD-Newmark late time instability, and a correction scheme," J. Comp. Physics, vol. 224, pp. 1293-1306, 2007.

Domain Decomposition Method

HFSS solves a linear matrix equation of the form:

$$\mathbf{Ax} = \mathbf{b} \qquad (1)$$

where the excitation vector **b** is commonly called the right-hand-side or RHS. The number of RHS vectors depends on the total number of modes used to excite the problem.

HFSS provides three distinct solution methods. By default, a direct solver is used and is the preferred method for small to medium size problems and/or problems with a large number of modes. A major strength of the direct solver is the ability to provide a solution for a large number of modes with relatively small computational overhead for the multiple RHS. However, if the problem size is too large for the machine memory, the second option is to use the iterative solver which can handle larger problems but requires a complete solution for each RHS. The iterative solver modifies the matrix equation (1) by multiplying both sides with a pre-conditioner, M, in order to enable a faster solution:

$$\mathbf{MAx} = \mathbf{Mb} \qquad (2)$$

Both the direct solver and the iterative solver work only with a single machine's shared memory. This limitation is addressed in HFSS with the introduction of an advanced solving technique called the Domain Decomposition Method (DDM) which enables the use of distributed memory across many different machines to solve very large problems.

- Basic DDM Theory
- Computational Memory
- Computational Time

Basic DDM Theory

DDM has emerged as a powerful and attractive technique due to its inherent parallelism which enables the use of distributed memory. DDM is based on a divide-and-conquer philosophy where instead of solving a large and complex problem directly, the original problem as defined by the mesh is partitioned into smaller, possibly repetitive, and easier to solve sub meshes or sub-domains. In this approach it is critical to enforce continuity of electromagnetic fields at the interfaces between adjacent sub-domains through some suitable boundary conditions.

To illustrate the basic idea of DDM, we solve (1) by decomposing the original problem into two domains. Subsequently, we have:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \qquad (3)$$

where $\mathbf{A}_{ii}$, $\mathbf{x}_i$ and $\mathbf{b}_i$, i=1,2 are system matrix, solution vector and RHS for domain *i*, respectively and $\mathbf{A}_{12}$, $\mathbf{A}_{21}$ are the coupling matrices between the two domains. To solve in parallel sub-domain problems, one popular domain decomposition algorithm is of Jacobi type:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{0} \end{bmatrix}. \quad (4)$$

Using (4) and applying iteration, (3) can be solved as

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}^{(n)} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}^{(n-1)}. \quad (5)$$

Through some simple algebra, equation (5) leads to two coupled systems:

$$\mathbf{A}_{11}\mathbf{x}_1^{(n)} = \mathbf{b}_1 - \mathbf{A}_{12}\mathbf{x}_2^{(n-1)} \qquad \mathbf{A}_{22}\mathbf{x}_2^{(n)} = \mathbf{b}_2 - \mathbf{A}_{21}\mathbf{x}_1^{(n-1)} \quad (6)$$

By setting $\mathbf{x}_1^{(0)} = 0$ and $\mathbf{x}_2^{(0)} = 0$ and, we have initial guess $\mathbf{x}_i^{(1)} = \mathbf{A}_{ii}^{-1}\mathbf{b}_i$ which is the local solution of each sub-domain problem. These initial solutions are then refined through coupling matrices $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ until equilibrium is reached.

The approach (5) is known as a stationary Jacobi solution of (3) which unfortunately converges rather slowly. A more advanced approach is to apply as a preconditioner

leading to:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}, \quad (7)$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{A}_{11}^{-1}\mathbf{A}_{12} \\ \mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}^{-1}\mathbf{b}_1 \\ \mathbf{A}_{22}^{-1}\mathbf{b}_2 \end{bmatrix}. \quad (8)$$

Related Topics

Domain Decomposition Method

Basic DDM Theory

Computational Memory

Computational Time

Computational Memory

Using domain decomposition we need to solve (factorize) $\mathbf{A}_{11}$, $\mathbf{A}_{22}$ compared to solving $\mathbf{A}$ using the standard method. When a direct solver is used, DDM is inherently more memory efficient since a series of smaller matrices are needed to be solved compared to a large matrix. This is evident by assuming that domains are equally partitioned so that (e.g. two domains) $N_1 = N_2 = N/2$ where N_1, N_2, and N are numbers of unknown for the sub-domains and the original problem respectively. Memory requirement for original problem typically scales as $C_1 N^{\alpha}$ where C_1 is a constant and $\alpha > 1$. For an M domain decomposition solve, the total memory requirement is

$$MC_1\left(\frac{N}{M}\right)^{\alpha} = M^{1-\alpha}C_1 N^{\alpha}$$

For example, if $\alpha = 1.5$, using 2-domains we save about 30% of total memory. Also keep in mind that the memory can be distributed across different machines enabling a problem to be solved which is not possible on a single machine.

Related Topics

Domain Decomposition Method

Basic DDM Theory

Computational Time

Computational Time

A domain decomposition solution consists of three steps:

1. Mesh partitioning, a relatively fast process. HFSS will automatically partition the mesh into roughly equivalent sized mesh sub-domains to balance memory and solution times for the domains.
2. Matrix-assembly/solve where both **A** and **M** are solved in parallel
3. Domain iteration where we solve **MAx=Mb** via an iterative solver.

The solution time for solving the entire system (1) without DDM requires $C_2 N^{\beta}$ operations with a direct solver where C_2 is a constant and $\beta > 1$ (and typically larger than α). For an M domain DDM solve, the factorization time of step 2 above is dominated by the most time consuming (i.e. largest) domains, given as:

$$C_2\left(\frac{N}{M}\right)^{\beta} = M^{-\beta} C_2 N^{\beta}$$

Therefore, given $\beta > 1$, this stage exhibits a super-linear speed-up. This type of speed-up is commonly referred to as "scalable".

For step 3 the major computation is on the matrix-vector multiplication (**MA**)**x** which can be parallelized. In each iteration, the master node produces a new solution vector **x** and then sends its appropriate components to each domain (cpu/node/machine). Each domain then sends back to the master the vector **y=MAx** which is used by the master to produce a new solution vector **x**. This process goes on until **x** converges. In this step, the cost of matrix-vector multiplication per iteration is also scalable with respect to number of domains. Therefore if the iteration number remains relatively unchanged as number of domains increases, the domain decomposition solver can exhibit overall scalable performance.

Related Topics

Domain Decomposition Method

Computational Memory

Basic DDM Theory

Integral Equation Method Used in HFSS-IE

The most commonly used electromagnetic integral equation is the Electric Field Integral Equation, or more commonly referred to by its acronym, EFIE. Consider a perfect conductor that is illuminated by a plane wave or any other source. On the surface of a perfect conductor the total tangential electric field must be zero. The total electric field comprises the incident electric field and the scattered electric field; hence, on the conducting surface.

The above relation only specifies the scattered field on the surface of the conductor. To fully understand the system, we need to be able to calculate the scattered field everywhere; one way to do this is to calculate the induced current which generates the scattered electric field.

$$\hat{n} \times (\vec{E}^{inc} + \vec{E}^{scat}) = 0$$

The field generated by a current is

where G is the Green's function defined as

$$\vec{E}(\vec{r}) = \frac{1}{j\omega\varepsilon}(\nabla\nabla \bullet + k^2)\int J(\vec{r}')G(|\vec{r} - \vec{r}'|)d\vec{r}'$$

and J is the surface current density. Combining the relationship between current and scattered field

$$G(r) = \frac{1}{4\pi}\frac{e^{-jkr}}{r}$$

with the boundary condition yields the EFIE

$$\hat{n} \times \vec{E}^{inc} = \hat{n} \times \left[\frac{1}{j\omega\varepsilon}(\nabla\nabla \bullet + k^2)\int J(\vec{r}')G(|\vec{r} - \vec{r}'|)d\vec{r}'\right]$$

The unknown current density is within the integral operator; these types of equations are known as integral equations. Also note the integral is a convolution. Electromagnetic integral equations are convolutions, and this allows us to make an analogy to system theory. The Green's function is the transfer function, or the impulse response of the system, and the incident electric field is the output. Our task is to find the input, or the current density, from the output knowing the transfer function.

Solving the EFIE is typically done through a finite element technique. We divide the surface of the conductor into many, many triangles, and on each triangle we assume a current distribution. This converts EFIE into a matrix equation, and in the electromagnetics literature this technique is called the method of moments (MoM).

The EFIE is just one of many different integral equations used in electromagnetics. We can vary the boundary conditions we enforce, or the method we use to specify the field everywhere. For example, if on the surface of the conductor, we enforce the continuity of the magnetic field, following a similar procedure yields the magnetic field integral equation (HFIE). All of them are similar in form to the EFIE and are solved in a similar manner.

References:

1. Roger Harrington, *Time Harmonic Electromagnetic Fields*, McGraw-Hill, New York, NY 1961.
2. Andrew Peterson, Scott Ray, Raj Mittra, *Computational Methods for Electromagnetics*, IEEE

Press, New York, NY 1998.

S-Parameters

One of the most important features of HFSS is the capability to categorize an electromagnetic structure in a compact manner using the concept of S-parameters. S-parameters are available for projects that contain ports and this document focuses on the case when the S-parameters are related to the eigen modes of the ports. The use of S-parameters for terminals are described in Modes to Terminals Conversion.

At each port, the modal representations of the electric and magnetic fields assuming K modes are:

$$\mathbf{E} = \sum_{m=1}^{K} (a_m + b_m)\mathbf{e}_m \quad (1)$$

$$\mathbf{H} = \sum_{m=1}^{K} (a_m - b_m)\mathbf{h}_m \quad (2)$$

where a_m and b_m are unitless complex amplitudes of the incident and reflected modal fields, respectively. Given a particular electromagnetic structure, it can be categorized in terms of the incident and reflected/transmitted modal amplitudes using the NxN generalized S-matrix, S:

$$\mathbf{b} = S\mathbf{a} \quad (3)$$

where **a** and **b** are unitless complex modal coefficient vectors. The size of these vectors, N, is the total number of modes obtained by adding up the number of the modes on all ports.

An entry of the S-matrix, S_{ij}, specifies the multiplication factor to be applied to the incident modal amplitude a_j to get the reflected/transmitted modal amplitude b_i given that all the other incident modes are turned off. This can be described mathematically as:

$$S_{ij} = \left.\frac{b_i}{a_j}\right|_{a_k = 0,\ k \neq j} \quad (4)$$

HFSS automatically provides an S-matrix when a project contains ports. The following links contain information on various topics related to S-matrices.

- Post Processed S-Matrices
- Passivity
- Alternative Matrix Characterizations

Post Processed S-Matrices

HFSS automatically computes the generalized S-matrix during the solve time when a project contains ports. The S-matrix can be modified after solution time by applying post processing transformations. The following post processing transformations are supported by HFSS:

- Renormalized S-Matrices
- De-embedded S-Matrices

Renormalized S-Matrices

Before a structure's generalized S-matrix can be used in a high frequency circuit simulator to compute the reflection and transmission of signals, it must be normalized to the appropriate impedance. For example, if a generalized S-matrix has been normalized to 50 ohms, it can be used to compute reflection and transmission directly from signals that are normalized to 50 ohms.

To renormalize a generalized S-matrix to a specific impedance, HFSS first calculates a unique impedance matrix *Z*, associated with the structure defined as follows:

$$Z = \sqrt{Z_0}(I-S)^{-1}(I+S)\sqrt{Z_0} \quad (1)$$

where

- S is the $n \times n$ generalized S-matrix.
- I is an $n \times n$ identity matrix.
- Z_0 is a diagonal matrix having the characteristic impedance (Z_0) of each port as a diagonal value.

The renormalized S-matrix is then calculated from the unique impedance matrix using this relationship:

$$S_\Omega = \sqrt{Y_\Omega}(Z-Z_\Omega)(Z+Z_\Omega)^{-1}\sqrt{Z_\Omega} \quad (2)$$

where

- Z is the structure's unique impedance matrix.
- Z_Ω and Y_Ω are diagonal matrices with the desired impedance and admittance as diagonal values. For example, if the matrix is being renormalized to 50 ohms, then Z_Ω would have diagonal values of 50.

Visualize the generalized S-matrix as an S-matrix that has been renormalized to the characteristic impedances of the structure. Therefore, if a diagonal matrix containing the characteristic impedances of the structure is used as Z_Ω in the above equation, the result would be the generalized S-matrix again.

HFSS needs to calculate the characteristic impedance of each port in order to compute a renormalized S-matrix.

The S-matrices initially calculated by HFSS are generalized S-matrices that have been normalized to the impedances of each port; however, you can compute S-matrices that are normalized to specific impedances, such as 50 ohms.

To convert a generalized modal S-matrix to a renormalized modal S-matrix, HFSS first needs to compute the characteristic impedance at each port. There are several ways to compute characteristic impedance. Two methods — the Zpv and Zvi methods — require an impedance, or *integration*, line.

HFSS will always calculate Zpi impedance, the impedance calculation using power and current, which are well-defined for a port because they are computed over the area of the port. Zpv and Zvi are not calculated by default. This is because v is computed by integrating along a user-defined integration line. If Zpi is very small (less than 10^-14) HFSS uses Zpv (if it is non-zero). In this

case, a warning says that the design has too few conductors touching the lumped port. To renormalize the solution to a Zpv or Zvi characteristic impedance, you must define an integration line.

De-embedded S-Matrices

If a uniform length of transmission line is added to (or removed from) a port, the S-matrix of the modified structure can be calculated using the following relationship $[S'] = [e^{\gamma l}][S][e^{\gamma l}]$, where

- $[e^{\gamma l}]$ is a diagonal matrix with the following entries:

$$\begin{bmatrix} e^{\gamma_1 l_1} & 0 & 0 \\ 0 & e^{\gamma_2 l_2} & 0 \\ 0 & 0 & e^{\gamma_3 l_3} \end{bmatrix}$$

- $\gamma = \alpha + j\beta$ is the complex propagation constant, where:
  - α is the attenuation constant of the wave.
  - β is the propagation constant of the uniform transmission line at port i.
- l_i is the length of the uniform transmission line that has been added to or removed from the structure at port *i*. A positive value indicates that a length of transmission line has been removed from the structure.

The value of γ for each port is automatically calculated by HFSS.

Related Topics

De-embedding S-Matrices

Alternative Matrix Characterizations

- Calculating the Z-Matrix
- Calculating the Y-Matrix

Calculating the Z-Matrix

The impedance matrix, *Z*, is calculated from the S-matrix as follows:

$$Z = \sqrt{Z_0}(I - S)^{-1}(I + S)\sqrt{Z_0} \qquad (1)$$

where

- S is the $n \times n$ generalized S-matrix.
- I is an $n \times n$ identity matrix.

- Z_0 is a diagonal matrix having the characteristic impedance (Z_0) of each port as a diagonal value.

Under the **Matrix Data** tab of the **Display Items Dialog**, the Z-matrix can be displayed as magnitude/ phase, real/ imaginary, magnitude, phase, real, or imaginary.

Calculating the Y-Matrix

The admittance matrix, *Y,* is simply the inverse of the impedance matrix, *Z*.

Under the **Matrix Data** tab of the **Solution Data** dialog box, the Y-matrix can be displayed as magnitude/ phase, real/ imaginary, magnitude, phase, real, or imaginary.

Passivity

Passive devices can only dissipate or temporarily store energy, but never generate it. The mathematical definition of passivity is based upon the following condition:

Q = I - conjugate(transpose(S)) * S must be a positive semidefinite matrix.

where:

- S is the S-parameter matrix
- I is an identity matrix.

A positive semidefinite matrix has only non-negative eigenvalues. The passivity test computes the eigenvalues of the matrix Q above at each frequency in the sweep. If any of the eigenvalues is negative, and larger (in magnitude) than the specified passivity tolerance, then a violation of passivity is reported to the user. The default value for passivity tolerance is .0001 tolerance above 1.0 magnitude.

Radiated Fields

When HFSS calculates radiation fields, the values of the fields over the radiation surface are used to compute the fields in the space surrounding the device. This space is typically split into two regions — the near-field region and the far-field region. The near-field region is the region closest to the source. In general, the electric field E(x,y,z) external to the region bounded by a closed surface may be written as

$$\boldsymbol{E}(x,y,z) = \int_s (\langle j\omega\mu_0 \boldsymbol{H}_{tan}\rangle G + \langle \boldsymbol{E}_{tan} \times \nabla G\rangle + \langle \boldsymbol{E}_{normal} \nabla G\rangle) ds \quad (1)$$

where

- s represents the radiation boundary surfaces.
- j is the imaginary unit, $\sqrt{-1}$.
- ω is the angular frequency, $2\pi f$.
- μ_0 is the relative permeability of the free space, $4\pi \times 10^{-7}$ Wb/Am.
- $\boldsymbol{H}_{tan}$ is the component of the magnetic field that is tangential to the surface.
- $\boldsymbol{E}_{normal}$ is the component of the electric field that is normal to the surface.
- $\boldsymbol{E}_{tan}$ is the component of the electric field that is tangential to the surface.
- G is the free space Green's function, given by

$$G = \frac{e^{-jk_0|\boldsymbol{r}-\boldsymbol{r}'|\sqrt{\mu_r \varepsilon_r}}}{|\boldsymbol{r}-\boldsymbol{r}'|} \quad (2)$$

where

- k_0 is the free space wave number, $k_o = \omega\sqrt{\mu_o \varepsilon_o} = \omega / c$
- $\boldsymbol{r}$ and $\boldsymbol{r}'$ represent field points and source points on the surface, respectively.
- ε_0 is the permittivity of free space, $1/(c^2\mu_0)$
- ε_r is the relative permittivity of a dielectric.
- μ_r is the relative permeability of a dielectric.

This r dependence is characteristic of a spherical wave, a key feature of far fields. The far field is a spherical TEM wave with the following equation:

$$\boldsymbol{E} = \eta_o \sqrt{\frac{\mu_r}{\varepsilon_r}} \boldsymbol{H} \times \hat{\boldsymbol{r}} \quad (3)$$

where η_0 is the intrinsic impedance of free space, $\eta_o = \sqrt{\frac{\mu_o}{\varepsilon_o}}$.

You can select the material in which the objects are embedded (the default is vacuum). In that case, the Green's function and the intrinsic impedance of that material will be used, so the far field will be correct. See Global Material Environment.

When calculating the near fields, HFSS uses the general expressions given in (**eq. 1**). You must specify the radial coordinate *r*. Because it can be used to compute fields at an arbitrary radius from the radiating structure, this command can be useful in EMC applications.

Note If HFSS calculates the near fields in a problem containing an incident wave, the radius at which the fields are calculated is very important. If the radius is within the solution region, then the fields calculated are either the total fields or the scattered fields depending upon which is selected. If the radius is outside the solution region, then the fields calculated are only the scattered fields.

When calculating the far fields, the previously discussed far-field approximations are used, and the result is valid only for field points in the far-field region.

Warning A radiation or PML boundary or Floquet port must have been defined in the design for HFSS to calculate radiated fields.

Related Topics

Assigning Boundaries

Assigning Excitations

Spherical Cross-Sections

Maximum Near Field Data

Selecting a Far Field Quantity to Plot

Spherical Cross-Sections

When you set up a spherical surface over which to analyze near or far fields, you specify a range and step size for phi and theta. These indicate the spherical direction in which you want to evaluate the radiated fields. For every value of phi there is a corresponding range of values for theta, and vice versa. This creates a spherical grid. Each grid point indicates a unique direction along a line that extends from the center of the sphere through the grid point. The radiated field is evaluated in this direction. The number of grid points is determined by the step size for phi and theta.

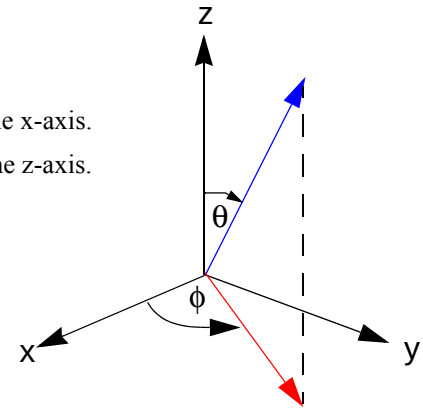
The sphere can be defined according to any defined coordinate system and before or after a solution has been generated.

Note For parts of the sphere outside of the model region, near-field approximation is calculated. However, if parts of the sphere are inside the model region, the model fields are used to compute interpolated values. A section of the sphere is considered to overlap the model if it lies in the enlarged model region after accounting for symmetry planes.

The relationship between phi and theta is shown below.

ϕ is rotated away from the x-axis.

θ is rotated away from the z-axis.



When HFSS evaluates the radiated fields, it needs at least two directions along which to plot the fields. Therefore, if the step size for phi is zero, then the step size for theta must be greater than zero, and vice versa. This ensures that the fields are plotted in at least two directions.

When setting up the sphere, phi and theta angles must be specified between -360 degrees (**deg**) and 360 degrees (**deg**), or the equivalents in radians (**rad**). If **deg** nor **rad** is specified, HFSS assumes the value to be in degrees.

Following are additional guidelines for specifying **Phi** in the **Near Field Radiation Sphere Setup** window or the **Far Field Radiation Sphere Setup** window:

Start	The point where the rotation of phi begins. The **Start** value must be equal to or greater than one.
Stop	The point where the rotation of phi ends. The **Stop** value must be greater than the **Start** value and less than 360. If the **Stop** value is equal to the **Start** value, then HFSS assumes that only one angle should be used and the **Step Size** value will be ignored.
Step Size	The number of degrees or radians (spherical grid points) between the sweep of phi. For example, to divide a sweep from 0° to 180° into 10° increments, you would enter **10**. Entering zero for the **Step Size** causes the sweep to consist of one point, the start value. If the **Step Size** value is zero, then HFSS assumes that only one angle should be used.

Following are additional guidelines for specifying **Theta**:

Start	The point where the rotation of theta begins. The **Start** value must be greater than -90 degrees, or the equivalent in radians.
Stop	The point where the rotation of theta ends. In classic definition the **Stop** value must be greater than the **Start** value and less than or equal to 180 degrees, or the equivalent in radians. In HFSS, you can input 360 degrees. If the **Stop** value is equal to the **Start** value, HFSS assumes that only one angle should be used and the **Step Size** value will be ignored.
Step Size	The number of degrees or radians (spherical grid points) between the sweep of theta. For example, to divide a sweep from -60 degrees to 60 degrees into 10-degree increments, you would enter **10deg**. Entering zero for the number of steps causes the sweep to consist of one point, the **Start** value. If the **Step Size** value is zero, then HFSS assumes that only one angle should be used.

Related Topics

Setting up a Far-Field Infinite Sphere

Setting up a Near-Field Sphere

Maximum Near-Field Data

The parameters listed in the **Max Field Data** window remain the same regardless of the geometry over which they were calculated. However, the coordinates displayed change depending on the geometry.

On a sphere, the coordinates — phi and theta — of the maximum value are listed under **Phi** and **Theta**. The values are given in volts per meter. Along a line, the coordinates — x, y, and z — of the maximum values are listed under **X**, **Y**, and **Z**. The values are given in volts per meter, and the coordinates are given in meters.

The following parameters are listed:

Total	The maximum of the total E-field.
X	The maximum E-field in the x-direction.
Y	The maximum E-field in the y-direction.
Z	The maximum E-field in the z-direction.
Phi	The maximum E-field in the ϕ-direction.
Theta	The maximum E-field in the θ-direction.
LHCP	The maximum left-hand circularly polarized component, which is equal to $\left\|\frac{1}{\sqrt{2}}(E_\theta - jE_\phi)\right\|$.

RHCP	The maximum right-hand circularly polarized component, which is equal to $\left\|\frac{1}{\sqrt{2}}(E_\theta + jE_\phi)\right\|$.
Ludwig 3/X dominant	The maximum of the dominant component, *Vmain*, for an x-polarized aperture using Ludwig's third definition of cross polarization. This is equal to $\|E_\theta cos\phi - E_\phi sin\phi\|$.
Ludwig 3/Y dominant	The maximum of the dominant component, *Vmain*, for a y-polarized aperture using Ludwig's third definition of cross polarization. This is equal to $\|E_\theta sin\phi + E_\phi cos\phi\|$.

Maximum Far-Field Data

When HFSS calculates antenna parameters, the following maximum field data is calculated:

Total	The maximum of the total rE-field.
X	The maximum rE-field in the x-direction.
Y	The maximum rE-field in the y-direction.
Z	The maximum rE-field in the z-direction.
Phi	The maximum rE-field in the ϕ-direction.
Theta	The maximum rE-field in the θ-direction.
LHCP	The maximum left-hand circularly polarized component, which is equal to $\left\|\frac{1}{\sqrt{2}}(E_\theta - jE_\phi)\right\|$.
RHCP	The maximum right-hand circularly polarized component, which is equal to $\left\|\frac{1}{\sqrt{2}}(E_\theta + jE_\phi)\right\|$.
Ludwig 3/X dominant	The maximum of the dominant component, *Vmain*, for an x-polarized aperture using Ludwig's third definition of cross polarization. This is equal to $\|E_\theta cos\phi - E_\phi sin\phi\|$.
Ludwig 3/Y dominant	The maximum of the dominant component, *Vmain*, for a y-polarized aperture using Ludwig's third definition of cross polarization. This is equal to $\|E_\theta sin\phi + E_\phi cos\phi\|$.

When calculating the maximum far field values, the distance *r* is factored out of the E-field. Therefore, the units for the maximum field data values are given in volts.

Array Factors

HFSS enables you to compute antenna array radiation patterns and antenna parameters for designs that have analyzed a single array element. You can define array geometry and excitation. HFSS models the array radiation pattern by applying an "array factor" to the single element's pattern.

Two array geometry types are supported. The "regular uniform array" geometry defines a finite 2D array of uniformly spaced, equal-amplitude elements. This is a natural specification after analyzing a single-unit cell of an infinite array. The regular array type may be scanned to a user-specified direction. Scan direction can be specified in terms of spherical coordinate angles in the radiation coordinate system. The regular array geometry type also allows scan specification in terms of differential phase shifts between elements.

The "custom array" geometry allows for greater flexibility. It defines an arbitrary array of identical elements distributed in 3D space with individual user-specified complex weights.

Cautionary Note for Array Factor Use

The field factorization (eq. 1) and consequent use of an array factor are useful tools for analyzing the radiated fields of antenna arrays; however, the analysis can yield incorrect results if used improperly. An HFSS single array element solution does not generally take into account the effects of the element's hypothetical neighbors.

For closely spaced array elements, these proximity effects (mutual coupling) may be significant. Consequently the patterns of the array elements vary with their position in the array and may depart significantly from the isolated element pattern. In such cases, the primary assumption in the use of the array factor is violated and the results will be inaccurate.

Note in particular that the array power expressions (eq. 13) and (eq. 14) neglect mutual coupling between elements of the finite array. Unless mutual coupling effects are negligible or have been implicitly included in the single element solution, the normalizations (eq. 13) and (eq. 14) gain and directivity are incorrect.

Related Topics

Defining Antenna Arrays

Theory of the Array Factor Calculation

Regular Uniform Arrays

Scan Specification for Regular Uniform Array

Custom Array

Power Normalization

Theory of the Array Factor Calculation

The composite far-field pattern, $\boldsymbol{E}_{array}(\phi, \theta)$ from an array of N identical radiating sources, each with far-field pattern $\boldsymbol{E}_{element}(\phi, \theta)$, may be factored into the form

$$\boldsymbol{E}_{array}(\phi, \theta) = AF(\phi, \theta)\boldsymbol{E}_{element}(\phi, \theta) \quad (1)$$

where the "array factor" AF (ϕ, θ) is defined as

$$AF(\phi, \theta) = \sum_{n=1}^{N} W_n e^{jk\mathbf{r}_n \cdot \hat{\mathbf{r}}} \quad (2)$$

and where

- (ϕ, θ) are the field-point spherical angles.
- W_n is the complex weight assigned to element *n*.
- j is $\sqrt{-1}$.
- k is 2π/λ., whereλ is the wave length in the global material environment.
- $\mathbf{r}_n$ is the position vector of element *n*, <*xn*, *yn*, *zn*>.
- $\hat{\mathbf{r}}$ is the pattern angle unit vector, <sinθcosϕ, sinθsinϕ, cosθ>.

The complex weights W_n in (eq. 2) may be written in terms of a (real) voltage amplitude A_n and (real) phase ψ_n as:

$$W_n = A_n e^{j\psi_n} \quad (3)$$

To scan a regular array in the direction (ϕ_0, θ_0), the element phases ψ_n are set to

$$\psi_n = -k\mathbf{r}_n \cdot \hat{\mathbf{r}}_o \quad (4)$$

where

$$\hat{\mathbf{r}}_0 = <\sin\theta_0\cos\phi_0, \sin\theta_0\sin\phi_0, \cos\theta_0> \quad (5)$$

is the scan-angle unit vector.

Regular Uniform Arrays

Let us define a uniform array as an array with unity amplitude weights for all elements, i.e., $A_n = 1$ for all *n*. For the case in which a uniform array is scanned to direction $\hat{\mathbf{r}}_0$, the array factor (eq. 2) becomes

$$AF(\phi, \theta) = \sum_{n=1}^{N} e^{jk\mathbf{r}_n \cdot (\hat{\mathbf{r}} - \hat{\mathbf{r}}_0)} \quad (6)$$

For a "regular" uniform array with element spacing defined by lattice vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, the element position vectors $\mathbf{r}_n$ may be written in the doubly-indexed form

$$\mathbf{r}_{mn} = (m-1)\boldsymbol{u} + (n-1)\boldsymbol{v} \quad (7)$$

with *m* = 1, 2, ..., *Nu* and *n* = 1, 2, ..., *Nv*. The total number of elements in the array is given by *N* = *Nu**Nv*.

The array factor (eq. 6) for the *Nu* x *Nv* array becomes

$$AF(\phi, \theta) = \sum_{m=1}^{N_u} \sum_{n=1}^{N_v} e^{jk\mathbf{r}_{mn} \cdot (\hat{\mathbf{r}} - \hat{\mathbf{r}}_0)} \quad (8)$$

Scan Specification for Regular Uniform Arrays

The scanning phase (eq. 4) is written in terms of the scan direction $\hat{\boldsymbol{r}}_0$. Alternatively, for a regular uniform array, the scanning phase may be written in terms of the differential phase shift between elements.

To develop this alternate scanning phase description, (eq. 7) is used to rewrite the expression (eq. 4) in doubly-indexed form as follows:

$$\psi_{mn} = -k\boldsymbol{r}_{mn} \cdot \hat{\boldsymbol{r}}_0 = -k(m-1)\boldsymbol{u} \cdot \hat{\boldsymbol{r}}_0 - k(n-1)\boldsymbol{v} \cdot \hat{\boldsymbol{r}}_0 \quad (9)$$

(Let us define ψ_u as the differential phase between adjacent elements in the **u** direction. Similarly, let us define ψ_v as the differential phase between adjacent elements in the **v** direction. Then

$$\psi_u \equiv \psi_{m+1,\,n} - \psi_{m,\,n} = -k\boldsymbol{u} \cdot \hat{\boldsymbol{r}}_0 \quad (10)$$

and

$$\psi_v \equiv \psi_{m,\,n+1} - \psi_{m,\,n} = -k\boldsymbol{v} \cdot \hat{\boldsymbol{r}}_0 \quad (11)$$

The scanning phase (eq. 4) may now be rewritten in terms of ψ_u and ψ_v as

$$\psi_{mn} = (m-1)\psi_u + (n-1)\psi_v \quad (12)$$

Thus in the case of a regular uniform array, the angle pair (ψ_u, ψ_v) may act as a substitute scan definition for the more general (ϕ_0, θ_0).

Custom Arrays

Once you have imported the array factor information from a text file, HFSS uses (eq. 8) to compute the array factor. When a custom array is defined, no scan direction is set and the array factor phase weights are those specified on an element-by-element basis in the geometry file.

The text file must have the following format. Coordinate values are assumed to be model units, and phase and amplitude are in SI units (radians and volts respectively):

```
N
x_1 y_1 z_1 A_1 P_1
x_2 y_2 z_2 A_2 P_2
...
...
x_N y_N z_N A_N P_N
```

where

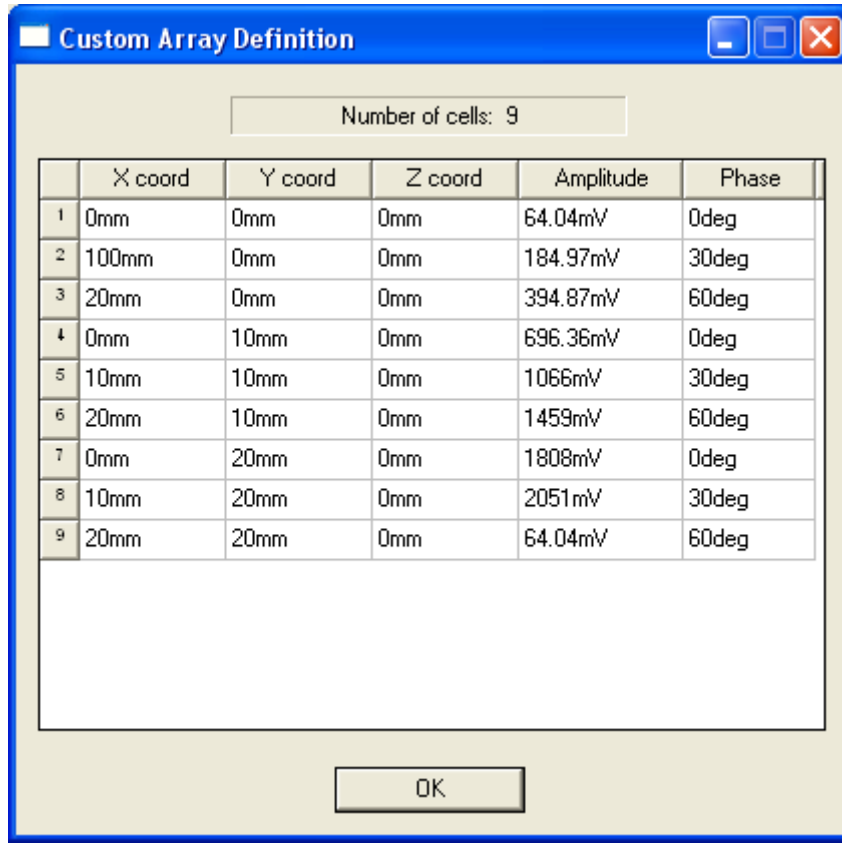
- x_1 is the x-coordinate position of the first element.
- y_1 is the y-coordinate position of the first element.
- z_1 is the y-coordinate position of the first element.
- A_1 is the amplitude weight of the first element. Amplitude references voltage.

- P_1 is the phase weight for the first element. Phase references radians.

Following is an example of a square 3 x 3 custom array geometry defined in a text file. The array elements are uniformly weighted and separated from one another in the x- and y-directions by 0.6729 user units.

```
! first line is a count of cells
9

! then [x,y,z] position of each cell
! and amplitude and phase
! default units for position are model units
! default units for amplitude is Volts
! default units for phase is radians

0       0       0       64.04mV     0
10cm    0       0      184.97mv     30deg
20      0       0      394.87mv     60deg
0       10      0      696.36mv     0
10      10      0      1.066        30deg
20      10      0      1.459        60deg
0       20      0      1.808        0
10      20      0      2.051        30deg
20      20      0      64.04mv      60deg
```

The information will appear as follows in the **Custom Array Definition** window:



	X coord	Y coord	Z coord	Amplitude	Phase
1	0mm	0mm	0mm	64.04mV	0deg
2	100mm	0mm	0mm	184.97mV	30deg
3	20mm	0mm	0mm	394.87mV	60deg
4	0mm	10mm	0mm	696.36mV	0deg
5	10mm	10mm	0mm	1066mV	30deg
6	20mm	10mm	0mm	1459mV	60deg
7	0mm	20mm	0mm	1808mV	0deg
8	10mm	20mm	0mm	2051mV	30deg
9	20mm	20mm	0mm	64.04mV	60deg

Power Normalizations

When the array factor feature is in use, the power normalization used to compute realized gain, gain, and directivity are modified as follows.

Let $P^{inc}_{element}$, $P^{acc}_{element}$, and $P^{rad}_{element}$ denote the incident, accepted, and radiated power of the single array element. When the array factor feature is invoked for an array of N elements, the array incident power, array accepted power, and the array radiated power will be computed simply as the sums of the corresponding element powers. Symbolically

$$P^{inc}_{array} = N \bullet P^{inc}_{element} \quad (13)$$

$$P^{acc}_{array} = N \bullet P^{acc}_{element} \quad (14)$$

$$P^{rad}_{array} = N \bullet P^{rad}_{element} \quad (15)$$

Caution: The above power array formulas are applicable to uniform excitation of large planar arrays analyzed using a unit cell with linked boundaries. For array elements that have been analyzed in isolation, only the incident power formula is rigorous, and the accepted and radiated power normalizations are generally not accurate.

Antenna Parameters

Generally, when dealing with radiated fields, you are also interested in the antenna properties of the radiated bodies. HFSS calculates the following antenna properties:

- Maximum intensity (Max U)
- Peak directivity
- Peak gain
- Peak realized gain
- Radiated power
- Accepted power
- Incident power
- Radiation efficiency
- Decay Factor

Warning A radiation or PML boundary must have been defined in the design for HFSS to calculate radiated fields.

Related Topics

Computing Antenna Parameters

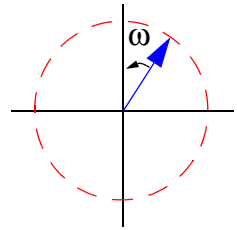
Polarization of the Electric Field

At each aspect angle in the far field of a radiating source, the electric and magnetic field vectors lie in a fixed plane. Over time, the instantaneous electric field vector traces out a figure or shape in this plane. This figure defines the polarization state of the field.

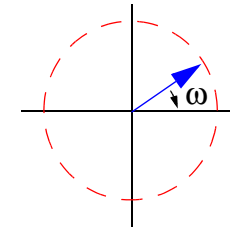
In general, this figure is an ellipse and is called the *polarization ellipse*. The wave is said to be *elliptically polarized* when the instantaneous electric field traces out an ellipse. As a special case, the polarization ellipse may be a circle, in which case the wave is *circularly polarized*. Elliptical and circular polarization have two different states, *left* and *right*, distinguished by the sense of rotation

of the electric field vector. Some of these figures, or states, are shown below. In each case the direction of propagation is off the screen.

Circular

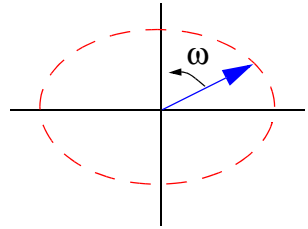


Right-hand circular polarization

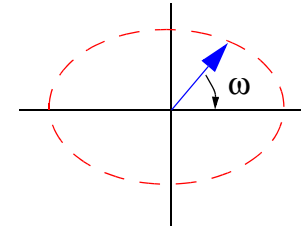


Left-hand circular polarization

Elliptical



Right-hand elliptical polarization



Left-hand elliptical polarization

where ω is the rotation radian frequency.

A second special case occurs when the polarization ellipse degenerates to a straight line. In this case the wave is *linearly polarized*.

To completely describe the polarization state of a radiated field, two independent components are required. HFSS supports three types of descriptions:

- Spherical polar
- Ludwig-3
- Circular

Spherical Polar

The most fundamental description of the polarization state of a radiated field is spherical polar, which is the electric field phasor resolved in the directions of unit theta and phi vectors of the reference coordinate system. In this description, the field may be written as $E = (E_\theta, E_\phi)$.

The polarization ratio for a predominantly ϕ-polarized antenna is equal to

$$\left|\frac{E_\phi}{E_\theta}\right| \quad (1)$$

The polarization ratio for a predominantly θ-polarized antenna is equal to

$$\left|\frac{E_\theta}{E_\phi}\right| \quad (2)$$

Ludwig-3 Polarization

Arthur C. Ludwig wrote a classic paper [Ref. 1]on the definition of cross polarization. In particular, his third definition is often used since it describes the field components that are typically measured on a far-field antenna test range. Using his definition, the radiated field may be written as $E = E_n, E_y$ where

$$E_x = E_\theta \cos\phi - E_\phi \sin\phi \quad (1)$$

$$E_y = E_\theta \sin\phi + E_\phi \cos\phi \quad (2)$$

and phi is the usual azimuthal angle in the reference spherical coordinate system.

[1] Arthur C. Ludwig, The Definition of Cross Polarization, IEEE Transactions on Antennas and Propagation, vol. AP-21 num. 1, pp. 116 -119, Jan. 1973.

Circular Polarization

For antennas designed to receive or transmit circularly polarized fields, a meaningful description is in terms of pure left and right circular states. In this description, the field may be written as $E = E_R, E_L$ where

$$E_R = \frac{1}{\sqrt{2}}(E_\theta + jE_\phi) \quad (1)$$

$$E_L = \frac{1}{\sqrt{2}}(E_\theta - jE_\phi) \quad (2)$$

Axial Ratio

Axial ratio is defined as the ratio of the major to the minor axis of the polarization ellipse.

1. E_x and E_y are orthogonal complex-valued field components. If either is zero, HFSS treats the field as linearly polarized. However, if neither E_x and E_y is zero:
2. Compute circular components E_- and E_+ from:

$$E_- = E_x - jE_y \quad (1)$$

$$E_+ = E_x + jE_y \quad (2)$$

3. If $E_+ = 0$ or $E_- = 0$ HFSS understands the field as perfectly circular, the axial ratio is 1. Otherwise, for the elliptical polarization case, HFSS determines the tilt angle τ from:

$$\text{phase}(E_-/E_+) = 2\tau \quad (3)$$

4. Rotate the original data to coincide with the axes of the polarization ellipse.

$$E'_x = E_x \cos\tau - E_y \sin\tau \quad (4)$$

$$E'_y = E_x \sin\tau - E_y \cos\tau \quad (5)$$

5. The Axial Ratio *AR* is given by:

$$AR = \left|\frac{E'_y}{E'_x}\right| \quad (6)$$

Because the above definition does not discriminate between major and minor ellipse axes, to enforce the convention that $AR \leq 1$, it is necessary to check this condition and if necessary invert the value obtained.

Polarization Ratio

The IEEE defines the (complex) polarization ratio as, "For a given field vector at a point in space, the (magnitude of the) ratio of the complex amplitudes of two specified orthogonally polarized field vectors into which the given field vector has been resolved." [Ref. 2] HFSS computes the following six polarization ratios at each selected aspect angle:

$$Circular/LHCP = \left|\frac{E_L}{E_R}\right| \quad (1)$$

$$Circular/RHCP = \left|\frac{E_R}{E_L}\right| \quad (2)$$

$$Spherical/Phi = \left|\frac{E_\phi}{E_\theta}\right| \quad (3)$$

$$Spherical/Theta = \left|\frac{E_\theta}{E_\phi}\right| \quad (4)$$

$$Ludwig\ 3/X = \left|\frac{E_x}{E_y}\right| \quad (5)$$

$$Ludwig\ 3/Y = \left|\frac{E_y}{E_x}\right| \quad (6)$$

[2] IEEE Standard Definitions of Terms for Antennas, IEEE Transactions on Antennas and Propagation, vol. AP-31 num. 6, Nov. 1983.

Max U

The radiation intensity, *U*, is the power radiated from an antenna per unit solid angle. HFSS calculates the radiation intensity in the direction in which it has the maximum value. The maximum intensity of the radiation is measured in watts per steradian and is calculated by

$$U(\theta,\phi) = \frac{1}{2}\frac{|\boldsymbol{E}|^2}{\eta_0}r^2 \quad (1)$$

where

- $U(\theta,\phi)$ is the radiation intensity in watts per steradian.
- $|E|$ is the magnitude of the E-field.

- η_0 is the intrinsic impedance of free space — 376.7 ohms.
- r is the distance from the antenna, in meters.

Related Topics

Computing Antenna Parameters

Peak Directivity

Directivity is defined as the ratio of an antenna's radiation intensity in a given direction to the radiation intensity averaged over all directions. Peak directivity, in turn, is the maximum directivity over all the user-specified directions of the far-field infinite sphere.

Directivity is a dimensionless quantity represented by

$$directivity = \frac{4\pi U}{P_{rad}} \quad (1)$$

where

- U is the radiation intensity in watts per steradian in the direction specified.
- P_{rad} is the radiated power in watts.

Note The peak directivity displayed in the **Antenna Parameters** window is the directivity in the direction of maximum radiation intensity, U_{max}.

- For a lossless antenna, the directivity will be equal to the gain. However, if the antenna has inherent losses, the directivity is related to the gain by the radiation efficiency of the antenna.

Related Topics

Setting up a Far-Field Infinite Sphere

Computing Antenna Parameters

Peak Gain

Gain is four pi times the ratio of an antenna's radiation intensity in a given direction to the total power accepted by the antenna. Peak gain, in turn, is the maximum gain over all the user-specified directions of the far-field infinite sphere.

The following equation is used to calculate gain in HFSS:

$$gain = 4\pi\frac{U}{P_{acc}} \quad (1)$$

where

- U is the radiation intensity in watts per steradian in the direction specified.
- P_{acc} is the accepted power in watts entering the antenna.

Gain can be confused with directivity, since they are equivalent for lossless antennas. Gain is related to directivity by the radiation efficiency of the antenna. If the radiation efficiency is 100%, they are equal.

Note Because the gain is calculated from the input signal at the port, a port must be defined for this quantity to be displayed.

Related Topics

Setting up a Far-Field Infinite Sphere

Computing Antenna Parameters

Peak Realized Gain

Realized gain is four pi times the ratio of an antenna's radiation intensity in a given direction to the total power incident upon the antenna port(s). Peak realized gain, in turn, is the maximum realized gain over all the user-specified directions of the far-field infinite sphere.

The following equation is used to calculate realized gain in HFSS:

$$\text{realized gain} = 4\pi\frac{U}{P_{incident}} \quad (1)$$

where

- U is the radiation intensity in watts per steradian in the direction specified.
- $P_{incident}$ is the incident power in watts.

Note Because the gain is calculated from the input signal at the port, a port must be defined for this quantity to be displayed.

Related Topics

Setting up a Far-Field Infinite Sphere

Computing Antenna Parameters

Radiated Power

Radiated power is the amount of time-averaged power (in watts) exiting a radiating antenna structure through a radiation boundary.

For a general radiating structure in HFSS, radiated power is computed as

$$P_{rad} = \Re\int_{s} \boldsymbol{E} \times \boldsymbol{H}^{*} \cdot ds \quad (1)$$

where

- P_{rad} is the radiated power in watts.
- $\Re$ is the real part of a complex number.
- s represents the radiation boundary surfaces.
- E is the electric field.
- H^{*} is the conjugate of the magnetic field.
- ds is the local radiation-boundary unit normal directed out of the 3D model.

Note The accuracy of the computed radiated power depends on the accuracy of $\boldsymbol{E}$ and $\boldsymbol{H}$. In some cases it is possible that the computed radiated power may deviate slightly from the actual radiated power.

The accuracy of the computed radiated power depends on the accuracy of E and H on the absorbing boundary. In some cases it is possible that the computed radiated power may deviate slightly from the actual radiated power. To increase the accuracy of the radiated power, seed the mesh on the absorbing boundary.

Related Topics

Computing Antenna Parameters

Accepted Power

The accepted power is the time-averaged power (in watts) entering a radiating antenna structure through one or more ports.

Fundamentally in HFSS, accepted power is computed as

$$P_{acc} = Re\int_{A} \boldsymbol{E} \times \boldsymbol{H}^{*} \cdot dS \qquad (1)$$

where

- P_{acc} is the accepted power in watts.
- Re is the real part of a complex number.
- A is the union of all port boundaries in the model, including Wave and Lumped Ports, but excluding Floquet ports.
- E is the electric field.
- H^{*} is the conjugate of the magnetic field.
- dS is the local port-boundary unit normal directed into the 3D HFSS model.

Example: Consider the simple case of a single port antenna modeled in a modal project. Then the general expression for accepted power reduces to

$$P_{acc} = |a|^{2}(1 - |s_{11}|^{2}) \qquad (1)$$

where

- $|a|$ is the complex modal excitation specified in the **Edit Sources** window.
- s_{11} is the single-entry generalized scattering matrix.

Note Because the accepted power is calculated from the input signal at the port, a port must be defined for this quantity to be displayed.

Thus when $|s_{11}|$ approaches 1 (total reflection), P_{acc} approaches zero. When $|s_{11}|$ approaches 0 (perfect match), $P_{acc} = |a|^{2}$, the total power carried by the incident mode.

In the multiport case the physical interpretation of accepted power can sometimes be confusing. By the fundamental definition, energy that leaves the antenna structure through a port reduces accepted power. This, in a multiport antenna, energy cross coupled from an excited port to a different port reduces accepted power. Therefore in such cases, even if an excited port is well-matched, P_{acc} may be small.

In the Edit Sources window of a terminal project, you specify total voltage V instead of model incident power. Using the underlying model S-matrix, HFSS converts the terminal excitation to total fields at the ports, and then applies the general definition.

Related Topics

Computing Antenna Parameters

Incident Power

Incident power for an antenna structure is defined in terms of accepted power under special conditions. In HFSS the incident power flow on a transmission structure is defined as the accepted power when that structure is terminated in a conjugate-matched load. Thus we write a general definition:

$$P_{inc} = P_{acc}\big|_{\text{conjugate match}} \quad (1)$$

For a transmission structure that may be modeled with a real characteristic impedance, this definition is equivalent to the zero-reflective case where the transmission structure is terminated in a matched load. For example, in the simple example discussed in this section on accepted power, setting $S_{11} = 0$ produced the incident power as:

$$P_{inc} = P_{acc}\big|_{S_{11}=0} = |a|^2 \quad (2)$$

However, there is an important difference in the case of Lumped ports with complex impedance, and Wave ports renormalized to complex impedance. In these cases, the conjugate-match and zero-reflection definitions of incident power give different results, but only the conjugate-match definition used by HFSS enforces the important property that $P_{acc} \leq P_{inc}$.

In the Edit Sources window of a terminal project, you specify total voltage V instead of the modal incident power. Using the underlying modal S-matrix, HFSS converts the terminal excitation to total fields at the ports, and then applies the general definition. As a result of this procedure, the incident power in a terminal project depends on both the user-specified stimulation as well as the model itself.

Related Topics

Computing Antenna Parameters

Radiation Efficiency

The radiation efficiency is the ratio of the radiated power to the accepted power given by

$$e = \frac{P_{rad}}{P_{acc}} \quad (1)$$

where

- P_{rad} is the radiated power in watts.
- P_{acc} is the accepted power in watts.

Note Because the radiation efficiency is calculated from the accepted power, a port must be defined for radiation efficiency to be displayed.

Related Topics

Computing Antenna Parameters

Decay Factor

When Global Environment Material is lossy, strictly speaking, the far field is zero, since the Green's function is:

$$|g| = \frac{1}{r}e^{-kr} = \frac{1}{r}e^{-(\alpha + j\beta)r} \quad (1)$$

where

$$k = j\omega\mu(\sigma + j\omega\varepsilon) = \alpha + j\beta \quad (2)$$

Against that fact, HFSS calculates the far field using the imaginary part of vector ***k***.

You can then modify the magnitude of the field pattern by using the real part (α) of the propagation constant, which is displayed as "Decay Factor).

ANSYS Near Field Description File Format Specification

The purpose of the ANSYS Near Field Description (AND) File is to facilitate the auto-generation of HFSS design for any Near Field Data the user may wish to use for a HFSS simulation. These Near Field Data files could be in a variety of different formats and files, so the AND Format was designed to be as flexible as possible to accommodate them.

Overview

- Recognized by the ".and" extension
- 'NearFieldHeader' Block - Provides format, field type, and geometry information.
- 'NearFieldData' Block - Links the user's near field data files to particular solve frequencies.

Blocks are defined with opening ($begin 'block_name') and closing ($end 'block_name') block tags that contain either key/value pairs, or function/argument calls described below.

```
$begin 'BlockName'
   key=value
   func(arg1, arg2)
$end 'BlockName'
```

NearFieldHeader Block

Every AND File requires a NearFieldHeader block shown below. This header is used to provide information to HFSS about the Near Field Data that may not be present in the Data Files them-

selves. Depending on the Near Field Data Format, some of this information may be redundant and merely used for verification purposes.

Key	Value Format	Description
type	*'format type*'	[Required] Format Type of the of the User's Near Field Data files.
fields	*'field type*'	[Optional] Field Type of the User's Near Field Data. Defaults to "EH" when not present.
geometry	*'primitive type*'	[Required] Primitive describing Near Field Source Geometry.
center	'10mm, 10mm, 10mm'	[Based on Geometry] Center of geometric primitive.
size	'10mm, 10mm, 10mm'	[Based on Geometry] xSize, ySize, zSize of primitive.
	'20mm, 20mm	Only two dimensions (XY, YZ or XZ) required for Plane Geometry.
axis	'0, 0, 1'	[Based on Geometry] Primary axis of primitive.
normal	'0, 1, 0'	[Based on Geometry] Normal axis of primitive.
radius	'20mm'	[Based on Geometry] Radius
height	'20mm'	[Based on Geometry] Height
fsweep	'1GHz, 2GHz, … 5.6 GHz'	[Required] The frequency band of the analysis.
fcoords	*'fcoords type'*	[Optional] Coordinate System used to describe data points

Valid Dimension Units:

Standard SI and Imperial Length Units are supported in Geometry Dimensions including:

'nm', 'mm', 'meter', 'cm', 'ft', 'in', 'mil', 'uin'

Standard SI Frequency Units are supported in 'fsweep' including:

'Hz, 'kHz, 'MHz, 'GHz, 'THz

Supported '*format type*':

- "nfd" - HFSS External Near Field Data File generated by exporting Fields.

Supported '*field type*':

- "EH" - Both E and H incident fields are provided. (Default)
- "E" - Only E Fields are provided.
- "H" - Only H Fields are provided.
- "JM" - Both J and M Tangential Currents are provided.
- "J" - Only J Current is provided.

- "M" - Only M Current is provided.

Supported '*primitive type*':

- ""sphere" - Spherical Geometry requiring center, radius.
- ""box" - Cube Geometry requiring center, size. Must be axis aligned.
- ""cylinder" - Cylinder Geometry requiring center, axis, radius, height. Must be axis aligned.
- ""plane" - Pseudo-Plane Geometry requiring center, normal, size. Must be axis aligned. Plane data is really treated as a Rectangle so a two dimensional size must be specified.

Supported '*fcoords type*':

- "cartesian" - Data Points are described using cartesian coordinates.
- "spherical" - Data Points are formatted using spherical coordinates.
- "cylindrical" - Data Points are formatted using cylindrical coordinates.

The Coordinate System Type is optional and will default to the corresponding system for that primitive type

("spherical" for Spheres, "cylindrical" for Cylinders, "cartesian" for Box/Plane).

Example NearFieldHeader Blocks

Box

```
$begin 'NearFieldHeader'
   type='nfd'
   fcoords='cartesian'
   fields='JM'
   geometry='box'
   center='0mm,0mm,0mm'
   size='20mm,20mm,20mm'
   fsweep='1GHz, 3GHz'
$end 'NearFieldHeader'
```

Sphere

```
$begin 'NearFieldHeader'
   type='nfd'
   fcoords='spherical'
   fields='EH'
   geometry='sphere'
   center='0mm,0mm,0mm'
   radius='150mm'
   fsweep='1GHz, 3GHz'
```

```
$end 'NearFieldHeader'
```

Plane

```
$begin 'NearFieldHeader'
   type='nfd'
   fields='E'
   geometry='plane'
   center='0mm,0mm,0mm'
   size='20mm,20mm'
   normal='0,1,0'
   fsweep='1GHz, 3GHz'
$end 'NearFieldHeader'
```

Cylinder

```
$begin 'NearFieldHeader'
   type='nfd'
   geometry='cylinder'
   center='0mm,0mm,0mm'
   axis='0,0,1'
   radius='20mm'
   height='258mm'
   fsweep='1GHz, 3GHz'
$end 'NearFieldHeader'
```

NearFieldData Block

The NearFieldData Block is used to specify which external data files to use for which frequencies. External Near Field Data can be spread across multiple files and multiple frequencies. This block provides a means of organizing these data files.

To bind a single Near Field Data file to a particular frequency, use:

```
FreqData("1GHz", "path/to/data.nfd")
```

To split Near Field Data files (one contains E Fields, one contains H Fields), use:

```
FreqData("1GHz", "path/to/efields.nfd", "path/to/hfields.nfd")
```

To bind a default Near Field Data file to all frequencies not explicitly defined, use:

```
FreqData("0", "path/to/data.nfd")
```

Example NearFieldData Blocks

```
$begin 'NearFieldData'
   FreqData("0", "exportfields.nfd")
```

```
$end 'NearFieldData'
$begin 'NearFieldData'
   FreqData("1GHz", "exportfields.nfd")
   FreqData("2GHz", "exportfields2.nfd")
   FreqData("3GHz", "exportfields3.nfd")
   FreqData("4GHz", "exportfields4.nfd")
   FreqData("5GHz", "exportfields5.nfd")
$end 'NearFieldData'

$begin 'NearFieldData'
   FreqData("0", "efields0.nfd", "hfields0.nfd")
   FreqData("2GHz", "efields1.nfd", "hfields1.nfd")
   FreqData("4GHz", "efields2.nfd", "hfields2.nfd")
$end 'NearFieldData'
```

Example .and Files

```
$begin 'NearFieldHeader'
   type='nfd'
   geometry='box'
   center='0mm,0mm,0mm'
   size='20mm,20mm,20mm'
   fsweep='1GHz, 3GHz'
   $end 'NearFieldHeader'
   $begin 'NearFieldData'
   FreqData("0", "exportfields.nfd")
$end 'NearFieldData'

$begin 'NearFieldHeader'
   type='nfd'
   fcoords='cartesian'
   geometry='cylinder'
   center='0mm,0mm,0mm'
   axis='0,0,1'
   radius='20mm'
   height='258mm'
```

```
   fsweep='1GHz, 2GHz, 3GHz, 4GHz, 5GHz'
$end 'NearFieldHeader'
$begin 'NearFieldData'
   FreqData("1GHz", "exportfields.nfd")
   FreqData("2GHz", "exportfields2.nfd")
   FreqData("3GHz", "exportfields3.nfd")
   FreqData("4GHz", "exportfields4.nfd")
   FreqData("5GHz", "exportfields5.nfd")
$end 'NearFieldData'

$begin 'NearFieldHeader'
   type='nfd'
   fcoords='cartesian'
   geometry='plane'
   center='0mm,0mm,0mm'
   normal='0,1,0'
   size="100mm,100mm"
   fsweep='1GHz, 2GHz, 3GHz, 4GHz'
   $end 'NearFieldHeader'
   $begin 'NearFieldData'
   FreqData("0", "efields0.nfd", "hfields0.nfd")
   FreqData("2GHz", "efields1.nfd", "hfields1.nfd")
   FreqData("4GHz", "efields2.nfd", "hfields2.nfd")
$end 'NearFieldData'
```

Calculating Finite Thickness Impedance

The **Assign DC Thickness** option on the **HFSS** menu is enabled if at least one object contains a good conducting isotropic material (such as copper), and the **Solve Inside property** is not selected. If the object meets these conditions, you can assign a DC thickness. If the thickness of the layer is finite, the skin impedance is calculated as:

$$Z = R + jX \tag{1}$$

$$R = \frac{1}{\sigma\delta}\frac{sh(2v) + \sin(2v)}{ch(2v) - \cos(2v)} \tag{2}$$

$$X = \frac{1}{\sigma\delta}\frac{sh(2v) - \sin(2v)}{ch(2v) - \cos(2v)} \tag{3}$$

where

$$\delta = \sqrt{\frac{2}{\omega\mu\sigma}} \tag{4}$$

$$v = \frac{h}{\delta} \tag{5}$$

where h is the layer thickness.

Similar skin impedance is assigned to surfaces of 3D objects of good conductors, which are of NoSolveInside and Thickness for DC Resistance is set.

Modes to Terminals Conversion

This section describes the conversion of modal S-parameters, computed using Ansoft HFSS's full-wave field solver, into terminal or voltage-based pseudo-S-parameters used in circuit theory.

The S-matrix solutions in HFSS are expressed in terms of incident and reflected complex amplitudes of waveguide modes. This description is mathematically and physically rigorous, but for the case of multi-conductor lines where multiples modes are propagating simultaneously it is convenient to convert the modes to physical quantities directly related to conductors such as currents. Examples of such situations include coupled transmission lines on printed circuit boards, multi-conductor cables, and many common types of electrical connectors. For these types of structures, which generally support multiple, quasi-transverse electromagnetic (TEM) modes of propagation, it is difficult to excite and measure a single mode. In any practical, or laboratory-like, measurement situation:

- measurements (e.g., of voltage) contain contributions from several modes
- the applied stimuli excite several modes simultaneously

Terminal support was added to HFSS to enable the simulation of terminal currents and voltages directly, without requiring engineers to determine the needed linear combinations of waveguide modes.

Background

Recall from electromagnetic field theory that the solutions for the transverse components of the electric and magnetic fields in a waveguide with a uniform cross-section can be written as

$$\vec{E}_t = \sum_n a_n \hat{e}_n(x,y)e^{-\gamma z} + \sum_n b_n \hat{e}_n(x,y)e^{\gamma z} \quad (1)$$

$$\vec{H}_t = \sum_n a_n \hat{h}_n(x,y)e^{-\gamma z} - \sum_n b_n \hat{h}_n(x,y)e^{\gamma z} \quad (2)$$

Here the a_n and b_n are the intensities of the forward and backward traveling modal waves $\hat{e}_n$ and $\hat{h}_n$ are the transverse electric and magnetic field patterns. Note that the a_n and b_n are considered to be dimensionless quantities here, while $\hat{e}_n$ and $\hat{h}_n$ have the normal units for electric and magnetic fields. For this discussion's purpose, assume that each port is defined at z = 0; this permits the removal of exponential terms in the above equations.

Terminal Voltages and Currents

It is possible to define a set of voltages for a port by establishing a number of different integration paths across the port. If a port supports *N* quasi-TEM modes of propagation, one can set up *N* different open contours of integration, C_1, C_2, ... C_N, and define *N* different voltages, v_1, v_2, ... v_N, according to the integral formulas

Alternatively, you can define a set of currents $\{i_k\}$using a set of closed contours of integration $\{D_k\}$:

$$v_k = -\int_{C_k} \vec{E}_t \cdot \vec{dl}. \quad (3)$$

$$i_k = \oint_{D_k} \vec{H}_t \cdot \vec{dl}. \quad (4)$$

These closed contour paths are set when you define the terminals on a port. Ansoft HFSS does not require you to set up the voltage integration paths separately. The software automatically infers the voltage paths from the chosen current contours using a power conservation relationship.

Each definition produces a set of relationships between the modal intensities and the voltages or currents. Since the total transverse fields can be expanded as a linear combination of modes, we can rewrite the previous equations as

$$v_k = -\sum_n a_n \int_{C_k} \vec{e}_n(x,y) \cdot \vec{dl} - \sum_n b_n \int_{C_k} \vec{e}_n(x,y) \cdot \vec{dl}. \quad (5)$$

$$i_k = \sum_n a_n \int_{D_k} \vec{h}_n(x,y) \cdot \vec{dl} - \sum_n b_n \int_{D_k} \vec{h}_n(x,y) \cdot \vec{dl} \quad (6)$$

To simplify, introduce the following terms:

$$t_{kn} = -\int_{C_k} \vec{e}_n(x,y) \cdot \vec{dl} \quad (7)$$

$$u_{kn} = \int_{D_k} \vec{h}_n(x,y) \cdot \vec{dl}. \quad (8)$$

Now v_k and i_k become

$$v_k = \sum_n a_n t_{kn} + \sum_n b_n t_{kn} \quad (9)$$

$$i_k = \sum_n a_n u_{kn} - \sum_n b_n u_{kn} \quad (10)$$

Since there are N possible voltage definitions and N quasi-TEM modes, there are NxN matrices that relate the terminal voltages and currents to the following modal intensities:

$$[i_k] = [u_{kn}]([a_n] - [b_n]) \quad (11)$$

$$[v_k] = [t_{kn}]([a_n] + [b_n]) \quad (12)$$

Now introduce the matrices $T = [t_{kn}]$ and $U = [u_{kn}]$ to represent these transformations. Note that the units of T are volts, while the units of U are amperes.

In summary, the **3D Post Processor** could be used to load different modal field patterns and manually perform the integrals defined previously over each terminal contour line. This would be a tedious and error-prone process, which HFSS's modes-to-terminals conversion feature eliminates.

Related Topics

Terminal Based Models for Circuit Analysis

Terminal Characteristic Impedance Matrix

Differential Pairs

Calculating the W-Elements

W-elements are a distributed model for transmission lines used in certain HSPICE-compatible circuit simulators. The W-element models in HFSS are computed from the 2D port solutions.

$$Z = R + j\omega L = \gamma * Z_0 \quad (1)$$

$$Y = G + j\omega C = \gamma * \text{inverse}(Z_0) \quad (2)$$

Extensions to the multi conductor case are possible but are beyond the scope of this discussion.

There are two flavors of W-element export in HFSS: Tabular and RLGC. (You see them in the Equivalent Circuit Export Options panel.) The more basic "RLGC" format fits the computed RLGC data to the model equations repeated below. For RLGC format RG parameters are frequency dependent such that HSPICE computes them as

$$R(f) = Ro + Rs * \text{sqrt}(f) \quad (3)$$

$$G(f) = Go + Gd * f \quad (4)$$

where Ro is the low frequency resistance and Go is the low frequency conductance. Rs and Gd represent the high-frequency asymptotes due to skin effect and loss tangent, respectively. L and C are assumed to be fixed over the frequency band.

Thus the "RLGC" format uses four matrices (Ro, Rs, Go, Gd) to model frequency dependence for R(f) and G(f), and it neglects the frequency dependence of L and C.

In contrast the "Tabular" format is not fitting the data to this simplified model: it is printing out the actual RLGC values at each frequency as computed by HFSS from the port solution.

Under some conditions, during W-Element export, HFSS may issue a warning that "One or more of the diagonal terms in the W-element <designated> matrix for some frequency are negative for a <specified port>."

This signifies a non-physical result. A negative diagonal entry in the RLGC matrices indicates that the model is non-passive at that frequency (i.e. can produce energy, which is probably not realistic for a metal interconnect.)

This can be caused by a number of things:

- Interpolating sweeps.
- Non-causal dielectric material models, such as the traditional constant loss tangent, constant permittivity dielectric model. The Djordjevic or Debye models in HFSS would be the preferred ones to use here.
- Allowing radiation boundaries on the ports.
- Solving to a very low frequency (one so low that the field solution from the full-wave solver contains significant numerical errors.)

Terminal-Based Models for Circuit Analysis

It is possible to use the transformations developed above to compute terminal-based admittance (Y), impedance (Z) and pseudo-S-matrices (S_p). First consider the admittance case. A relationship of the form $i = Yv$ is needed that gives the vector of terminal currents $i = [i_k]$ as a function of the vector of terminal voltages $v = [v_k]$. It is known that

$$v = T(a + b) \tag{1}$$

and

$$i = U(a - b) \tag{2}$$

It is also known that the incident and scattered modal amplitudes are related by $b = Sa$, where S is the modal S-matrix computed by HFSS. Therefore:

$$v = T(a + Sa) = T(I + S)a \tag{3}$$

$$i = U(a - Sa) = U(I - S)a \tag{4}$$

Here I denotes an identity matrix of the same size as S. i can then be solved for in terms of v by eliminating the incident wave variable a.

$$a = (I + S)^{-1} T^{-1} v \tag{5}$$

$$i = U(I - S)(I + S)^{-1} T^{-1} v \tag{6}$$

The terminal-based admittance matrix Y is identified from the above expression as:

$$Y = U(I - S)(I + S)^{-1} T^{-1} \tag{7}$$

A similar relationship can be developed for the terminal-based impedance matrix:

$$Z = T(I + S)(I - S)^{-1} U^{-1} \tag{8}$$

It is also possible to convert the terminal-based admittance and impedance matrices into a terminal-based "pseudo-S-matrix" S_p. To do this, a reference impedance matrix must be specified. Then standard formulas are used to convert the terminal impedance matrix Z into the terminal S-matrix

$$S_p = Z_{\text{ref}}^{1/2}(Z + Z_{\text{ref}})^{-1}(Z - Z_{\text{ref}})Z_{\text{ref}}^{-1/2} \quad (9)$$

Z_{ref} can be either a user defined diagonal reference impedance matrix or the terminal characteristic impedance matrix, Zo, which is the matched case for terminal-based models.

The terminal S-matrix S_p relates the intensities of the incident and reflected pseudo-waves at the terminals.

$$\beta = S_p\alpha \quad (10)$$

These pseudo-waves are defined by:

$$\alpha = \frac{1}{2}Z_{ref}^{-1/2}(v + Z_{ref}i) \quad (11)$$

$$\beta = \frac{1}{2}Z_{ref}^{-1/2}(v - Z_{ref}i) \quad (12)$$

Note that the units of α and β are watts$^{1/2}$. The terminal voltages and currents can also be written in terms of the pseudo-waves.

$$v = Z^{1/2}(\alpha + \beta) \quad (13)$$

$$i = Z^{-1/2}(\alpha - \beta) \quad (14)$$

Unlike true waveguide modes, the pseudo-waves α and β have no associated propagation constant. The pseudo-waves represent linear combinations of several modes, which may all have differing propagation constants. HFSS is still capable of performing de-embedding on the terminal-based S-matrix, but this is accomplished by first de-embedding the modal S-matrix and then performing the transformation back to a terminal-based S-matrix.

For the normalization of terminal voltages in the Fields Post Processor, see **Scaling a Source's Magnitude and Phase**.

Related Topics

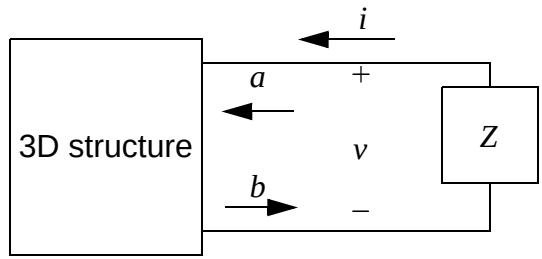
Modes to Terminals Conversion

Terminal Characteristic Impedance Matrix

Differential Pairs

Terminal Characteristic Impedance Matrix

Consider the situation illustrated below. A 3D structure with one multi-mode waveguide port is loaded by an N-port matrix impedance Z. The structure contains internal sources, which generate outgoing waves that exit through the waveguide port and strike the impedance Z. If Z does not match the impedance of the waveguide in some sense, a reflection will occur from this load and will return to the 3D structure, where it is interpreted as an incident wave.



An optimal choice of Z will prevent any reflections from the load returning as an incident wave a. Note that the circuit equations at the load are $v = -Zi$ (the minus sign is due to the sense of the current i). By replacing the voltages and currents with their modal expansions, the voltage becomes

$$v = T(a+b) = -ZU(a-b) \quad (1)$$

Rearranging this to isolate a and b, it is determined that:

$$(T+ZU)a + (T-ZU)b = 0 \quad (2)$$

Now notice that if we select $T = ZU$; the incident wave a vanishes. Corresponding to this condition is an "optimal" choice Z_0 for the impedance Z:

$$Z_0 = TU^{-1} \quad (3)$$

Z_0 is the terminal characteristic impedance matrix for the multi-mode waveguide port. This value of impedance will completely absorb any linear combination of modal waves leaving the port. As such, it should be of interest to circuit designers wishing to control reflections. In the important, special case of a lossless waveguide, it can be shown that $Z_0 = TT^T$ is a real-valued, symmetric impedance matrix. It is then easy to synthesize a network of resistors with the specified matrix impedance.

Related Topics

Modes to Terminals Conversion

Terminal Based Models for Circuit Analysis

Differential Pairs

Overview of the Technical Approach for Derivatives in HFSS

HFSS includes the ability to compute derivatives of S-parameters and related matrix quantities. The variables of differentiation are project variables or design properties. By far the most useful variable type is geometric, though these also require the most computational effort.

The general approach by which derivatives are computed in HFSS is not really new and the general subject of linear systems has long contained the high-level plan in use. However, the application to full-wave field solvers is relatively recent, and of course the details of the HFSS implementation is where original work in this regard was done.

It is important to realize that in HFSS, it is the discrete (approximate) solution which is differentiated. In a nutshell, the differentiation process can be described as follows. The global matrix equation that HFSS sets up and solves can be written as Ax = b where A is a very large matrix, x is the unknown vector containing fields and S-parameters, and b represents the excitations. Given a model parameter "g" (e.g. the diameter of an iris in a filter) each of the quantities in the matrix equation is in turn a function of g.

Formal differentiation of the global equation with respect to g and simple rearrangement yields Ax'= b' - A'x, where the primes indicate derivatives. The quantity sought is x' and the matrix multiplying this quantity is A, just as in the original global matrix equation. The right-hand side contains b', A', and x. The latter quantity is the solution to the global system and is assumed already known. Thus if b' and A' can be computed, x' may be obtained by solving a matrix equation of the same form as the original Ax=b.

Now, consider the quantities A' and b'. Restriction is currently made to the important case where the excitation is from ports only, in which case b'= 0, leaving the equation Ax'= -A'x. To generate the right-hand side of this matrix equation, the key remaining quantity to be supplied by the field solver is therefore A'. In HFSS this is done analytically for the most part. Thus a matrix assembly algorithm has been crafted that fills A' in a manner similar to how A is filled. So "analytical", rather than "numerical" differentiation is a principal aspect of the HFSS method.

Also, it can be shown that in most cases the computation of x' can avoid even an additional matrix solution (the system is "self adjoint") and S' is obtained by a purely algebraic post-process operation once A' is assembled and x is known.

Finally, it should be noted that if there are multiple variables of differentiation, each one is simply considered individually. Differentiation of response functions is not currently supported.

In regards to fundamental limitations, mostly the issue is computation of A' when the variable of differentiation affects ports. Considerable original work was successfully performed to include such variables, but not all cases are doable yet, in particular certain ports containing degenerate modes.

Some technical details on the implementation of derivatives in HFSS may be found in: "Sensitivity Analysis of S-parameters Including Port Variations Using the Transfinite Element Method", L Vardapetyan, J. Manges, Z. Cendes, 2008 IEEE International Microwave Symposium, Atlanta Georgia, June 2008.

Best Practices for Accurate and Causal Broad Band Frequency Sweeps

This section presents practices applied in the Finite Element Method in order to extract accurate and causal broad band frequency sweeps when simulating Printed Circuit Boards (PCBs) and interconnects. The importance and treatment of metal traces of finite thickness, accurate DC point, causal material models, causal surface roughness models and deembedding parasitic effects will be discussed and demonstrated by simulating real life problems.

Introduction to Causality Issues for Simulations

Signal integrity (SI) engineers are challenged to efficiently design and verify high-speed PCBs and interconnects by simulation. When working in the multi-GHz frequency range, simulation typically requires full-wave electromagnetic analysis. The Finite Element method (FEM) is the most common way to rigorously and accurately simulate 3D structures in a broad frequency range and extract the S parameters, which will then be used in a transient SPICE simulation as black boxes to predict the signals in the overall system. A critical consideration for these is the causality behavior of the resulting S-parameter models. Causality means that an output signal cannot start to change before the input signal changes. A necessary and sufficient condition for the causality is that the frequency domain transfer functions (S parameters) have to satisfy the Kramers - Kronig relations [1]. This technical note presents the best practices for getting accurate and causal solutions and not how to enforce causality of existing broad band S parameters. The Kramers - Kronig relationship will just be applied as a causality checker.

Physical Parameters Compared to Simulation

There might be two sources of the discrepancies between the physical parameters and the results of the simulation: one is the improper mathematical model used for the FEM calculations and the other is the numerical solution itself. Both discrepancies can result in causality violation. The paper focuses on the proper mathematical models. It is supposed that the adaptive mesh refinement has converged for the finite element solution, so the mesh is fine enough to approximate the exact solution well.

Modeling Issue: Solve Inside and Surface Impedance Boundaries

One of the most important modeling issues is how the solution should be treated in metals when the frequency range is broad. It is well known that the skin depth varies between infinity and very small value. A large skin depth typically necessitates solving the field inside the metal objects. This solve inside option is impractical for very small skin depths (high frequency), because a large mesh

is needed to model the very strong decay of the field. A widely used method is to apply the infinite half space skin impedance as a Surface Impedance Boundary Condition (SIBC), replacing the solution inside the metal objects. It is accurate for high frequencies, MHz to GHz for typical interconnect and PCB geometries, but very inaccurate/unphysical for low frequencies when the finite thickness of the object approaches the skin depth δ, because the real part of the impedance goes to zero with decreasing the frequency instead of going to an appropriate finite DC value. An obvious option would be to combine two methods; apply solve inside at low frequencies and use SIBC at high frequencies. However problems would arise in the transition region, strongly violating causality by introducing jumps or non-physical smoothing. The paper presents a method, which uses the analytical impedances of the conductors with finite thickness in the whole frequency region. The method doesn't violate the causality, since the analytical formulas satisfy the Maxwell's equations and the DC point can also be approximated with arbitrary accuracy.

Modeling Material Properties

A common dielectric material model is to assume constant properties across the frequency range. However this is not a physical model of the loss mechanism in dielectrics and violates the causality of the results. Fitting a Debye or Djordjevic - Sarkar material model [3] to the input data will provide a causal model definition.

Modeling Surface Roughness

Another issue is how to take the loss mechanism of surface roughness into account. Two methods for obtaining causal surface roughness models will be presented and discussed in the paper.

When extracting S parameters of the system, ports are inserted into the simulation model. Ports won't introduce any error when they model perfect transmission line terminations, which is possible in FEM codes, unlike Method of Moment (MoM) codes, where so called port calibration is always necessary. In geometries where a transmission line port is not possible, the FEM method allows for non-transmission line terminations (e.g. lumped gap ports). Such a situation most often occurs when there is not even room for a complete transmission port in the design. The user is still interested in extracting S parameters as if the model was terminated by perfect transmission line wave ports. A method will be presented to deembed the parasitic effects introduced by using lumped gap ports.

If all the above mentioned techniques are used in a FEM extraction and the adaptive meshing procedures have converged within a reasonably small tolerance, the S parameters will be accurate and causal. Real-life problems will be presented to demonstrate this using ANSYS HFSS, where all the techniques are implemented.

Next

Surface Impedance Boundary Condition (SIBC) for Metal Traces of Finite Thickness

Related Topics

Best Practices for Accurate and Causal Broad Band Frequency Sweeps

References for Accurate and Causal Broad Band Frequency Sweeps

Surface Impedance Boundary Condition (SIBC) for Metal Traces of Finite

Thickness

As previously noted, it is not recommended to solve inside of conductive objects in the whole frequency range. The task is to apply equivalent SIBC on the surface which result in the right circuit impedance in the whole frequency range. The derivations will be detailed for internal objects only.

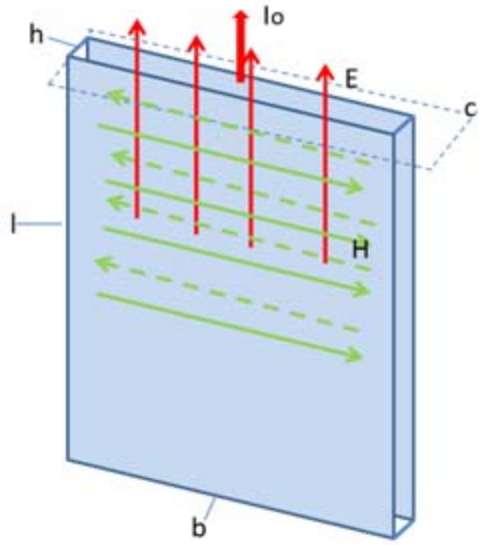


Figure 1: A conductor of finite thickness

Using the notations of Figure. 1 and applying Ampere's law, we get:

$$\oint_c Hdc = I_o \qquad 2Hb = I_o \tag{1}$$

Integrating the Poynting vector over both sides of the object:

$$P = 2EHlb = I_o^2 Z_{circuit} \tag{2}$$

Then we get:

$$Z_{field} = \frac{E}{H} = 2Z_{circuit}\frac{b}{l} \tag{3}$$

Using the circuit impedance analytical formula from [5], the real and imaginary parts of the field impedance for the SIBC are:

$$R_{field} = \frac{1}{\sigma\delta}\frac{sh(v) + \sin(v)}{ch(v) - \cos(v)} \tag{4}$$

$$X_{field} = \frac{1}{\sigma\delta}\frac{sh(v) - \sin(v)}{ch(v) - \cos(v)} \quad (5)$$

Here $v=h/\delta$ and δ is the skin depth. The formulas satisfy the asymptotic conditions, namely at low frequencies and

$$R_{circuit\_DC} = \frac{1}{\sigma h} \text{ at low frequencies and } R_{field\_high\_freq} = X_{field\_high\_freq} = \frac{1}{\sigma\delta} \quad (6)$$

at high frequencies. For outer surfaces, where the SIBC is applied to the outer boundary of the computational domain,

$$Z_{field} = \frac{E}{H} = Z_{circuit}\frac{b}{1} \quad (7)$$

$$R_{field} = \frac{1}{\sigma\delta}\frac{sh(2v) + \sin(2v)}{ch(2v) - \cos(2v)} \quad (8)$$

$$X_{field} = \frac{1}{\sigma\delta}\frac{sh(2v) - \sin(2v)}{ch(2v) - \cos(2v)} \quad (9)$$

Using SIBC, the solution, including the extrapolated DC point, will be more accurate because the DC behavior is modeled accurately. Also the causality is not violated because the analytical formulas satisfy the Maxwell's equations. Experimental data shows that FEM with low frequency treatment can cope with structures when the ratio of the average mesh size and the wave length is 10-10. Assuming a PCB with 1um average mesh size. It allows us to solve down to 300 Hz.

Next

Using Causal Material Models

Related Topics

Best Practices for Accurate and Causal Broad Band Frequency Sweeps

References for Accurate and Causal Broad Band Frequency Sweeps

Using Causal Material Models

Using causal dielectric material models is one of the most important considerations to extract causal S parameter models. Causal material model means that the real and imaginary parts of the permittivity should satisfy the causality relationship (causal magnetic materials are not discussed here).

The most general material model is the multipole Debye model [3]:

$$\varepsilon_r(\omega) = \varepsilon'_\infty + \sum_{i=1}^{N} \frac{\Delta\varepsilon_i}{1 + j\frac{\omega}{\omega_1}} - j\frac{\sigma}{\omega\varepsilon_o} \quad (10)$$

It is applicable when we have information about the permittivity and the loss tangent of the material in the frequency range, such as when measured material data are available. Single pole Debye model is also a good approximation when just polarization losses are present and the polarization frequency is known:

$$\varepsilon_r(\omega) = \varepsilon_{r\_optical} + \frac{\varepsilon_{r\_static} - \varepsilon_{r\_optical}}{1 + j\frac{\omega}{\omega_{pol}}} \quad (11)$$

In other cases, especially when the permittivity and loss tangent are known at only one or two frequencies, the broad band Djordjevic - Sarkar (D-S) material model can be used:

$$\varepsilon_r(\omega) = \varepsilon_{r\_optical} + \frac{\varepsilon_{r\_static} - \varepsilon_{r\_optical}}{m_2 - m_1} \frac{\ln\frac{\omega_2 + j\omega}{\omega_1 + j\omega}}{\ln 10} - j\frac{\sigma}{\omega\varepsilon_o} \quad 12$$

where

$$\omega_1 = 10^{m_1} \qquad \omega_2 = 10^{m_2}$$

are the limits of the frequency range. The D-S model can also be viewed as a multipole Debye model with an infinite continuum of poles.

Using causal material models leads to frequency dependent complex materials, which is standard in most finite element codes.

Next

Surface Roughness Modeling

Related Topics

Best Practices for Accurate and Causal Broad Band Frequency Sweeps

References for Accurate and Causal Broad Band Frequency Sweeps

Surface Roughness Modeling

Surface roughness plays an important role in accurately modeling radio frequency components (RFC), e.g. resonators, and signal integrity projects, like microstrips and printed circuit boards (PCBs). The increase of the surface resistance due to surface roughness results in lower quality factor (Q) in resonators and higher insertion losses for microstrips and PCBs. The scale of the roughness is different in the different application areas so different surface roughness models are used. The common theme in the different models is that an effective frequency dependent conductivity is defined depending on the type and magnitude of the roughness, which shows the increase of the surface resistance compared to the smooth surface resistance.

For RFC projects, Groiss [6] or Hammerstand [4] model may be employed.

Impedance at a rough surface is higher than the impedance at a smooth surface by a factor $fact_R$ as shown in the general equation below.

$$Z_{rough} = Z_{smooth} fact_R; \quad (13)$$

where Z_{rough} is the impedance at the rough surface and Z_{smooth} is the impedance at the smooth surface.

For the Groiss model the factor is given by the equation:

$$fact_R = 1 + e^{-\left(\frac{\delta}{2s}\right)^{1.6}} \quad (14)$$

where δ is the skin depth and s is the surface roughness.

It can be seen that the maximal ratio is two, so this model is just applicable to RFC problems, where the surfaces are highly polished to avoid lower Q. A surface roughness model of higher losses is necessary to model the fabrication effects in PCBs where the roughness of the deposited metal is considerably higher.

The recently developed Huray surface roughness model ([2], [4]) copes with the problem by applying a factor of higher saturation value as shown below:

$$fact_R = 1 + 1.5 S_r \frac{1}{1 + \frac{\delta}{a} + \frac{\delta^2}{2a^2}} \quad (15)$$

where S_r is the ratio of the surfaces of all "spheres" found in a cut and the cut surface. a is the radius of the spheres and δ is the skin depth. All SIBCs depend on the conductivity explicitly and implicitly as well.

Note: For more information about Huray model, see the section Surface Roughness Modeling - Huray Model.

The figure below shows the insertion losses. Green, and red curves show the measured and Huray model respectively. .

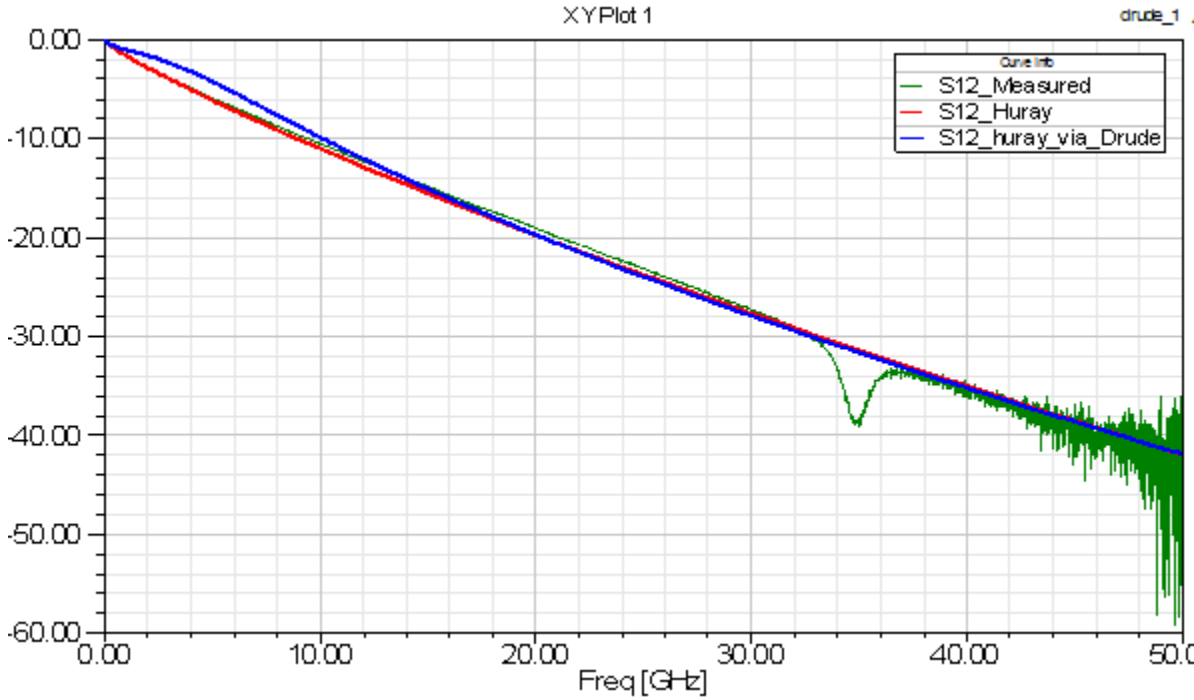


Figure 3: Insertion loss of a microstrip line. Measured (green), Huray model (red). Resonance is due to fiberglass weave [9].

Next

Deembedding Parasitic Lumped Port Effects

Related Topics

Best Practices for Accurate and Causal Broad Band Frequency Sweeps

References for Accurate and Causal Broad Band Frequency Sweeps

Deembedding Parasitic Lumped Port Effects

Fig. 4 shows a strip line, which is fed by internal lumped ports instead of perfect wave ports. The FEM simulation was performed using these excitations and resulted in the entire scattering matrix, S^*. The goal is to calculate scattering matrix S (scattering matrix of the DUT), as if the structure were fed by perfect virtual ports. It can be done if S matrices of the transition or error boxes are known ($S^{(1)}$and $S^{(2)}$ in Fig.2). When **b* =S*a*** is known and **b=Sa** is sought, we get:

$$\mathbf{S} = [\langle \mathbf{C}_{bb} \rangle + \mathbf{S}^* \langle \mathbf{C}_{ab} \rangle]^{-1} [\mathbf{S}^* \langle \mathbf{C}_{aa} \rangle - \langle \mathbf{C}_{ba} \rangle] \quad (16)$$

where

$\langle \mathbf{C}_{aa} \rangle$, $\langle \mathbf{C}_{ab} \rangle$, $\langle \mathbf{C}_{ba} \rangle$, and $\langle \mathbf{C}_{bb} \rangle$

are diagonal matrices:

$$\mathbf{C}^{i}_{aa} = \frac{1}{S_{21}^{(i)}}; \quad \mathbf{C}^{i}_{ab} = \frac{S_{22}^{(i)}}{S_{21}^{(i)}}; \quad \mathbf{C}^{i}_{ba} = \frac{S_{11}^{(i)}}{S_{21}^{(i)}}; \quad \mathbf{C}^{i}_{bb} = S_{12}^{(i)} - \frac{S_{11}^{(i)} S_{22}^{(i)}}{S_{21}^{(i)}} \qquad (17)$$

; ; (18)

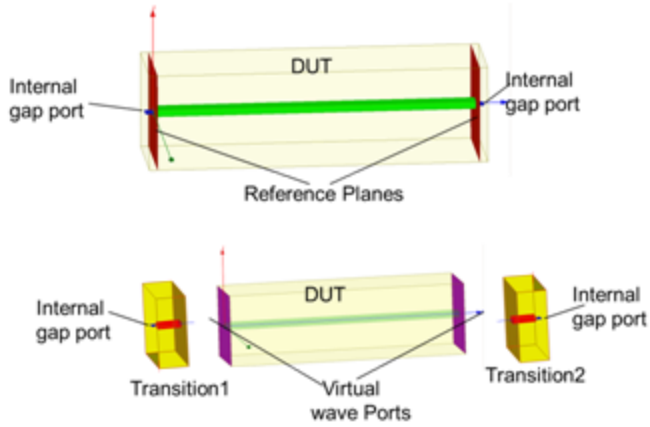


Figure 4: Strip line (DUT) fed by two internal lumped ports

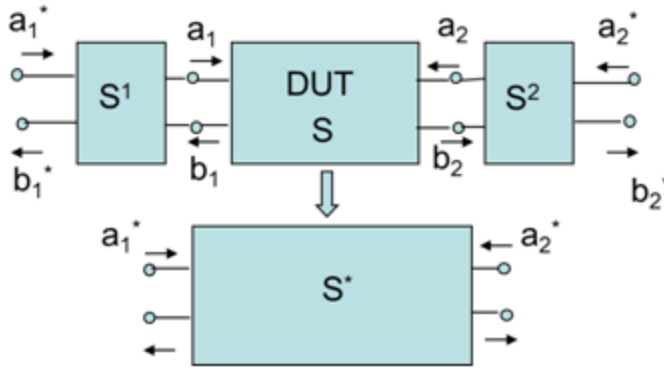


Figure 5: Black boxes of a strip line fed by two internal lumped ports

Usually, the device under test (DUT) is very large and complex. It is not possible to create perfect wave ports inside the complex geometry. However, it is not necessary to do that, when using the proposed method. The transition boxes can be independently created and solved as very small, 3D projects with two ports. In that case, the method is exact, within computational errors. In simple cases, *S* matrices of the transition boxes can be estimated, e.g. by a series impedance. The simplest case is when a lumped gap port is represented by a "parasitic" inductance. Applying experimental approximations for the calculation of the partial inductance of a lumped gap port ([7], [8]):

$$L = 0.2length\left(\ln\frac{2length}{w} + \frac{0.223w}{length} + 0.5\right) \quad (18)$$

where length and w are the length and width of the gap port in millimeters, respectively. L will be in nH.

Fig. 6 shows the inductance of a pair of cylindrical wires. The two feeding lumped gap ports introduce parasitic inductances, which can be deembedded by using the above method, which analytically accounts for the DC and AC inductances. The maximum error is reduced by roughly a factor of 3; 4.3% without deembedding and 1.6% with deembedding the parasitic inductance.

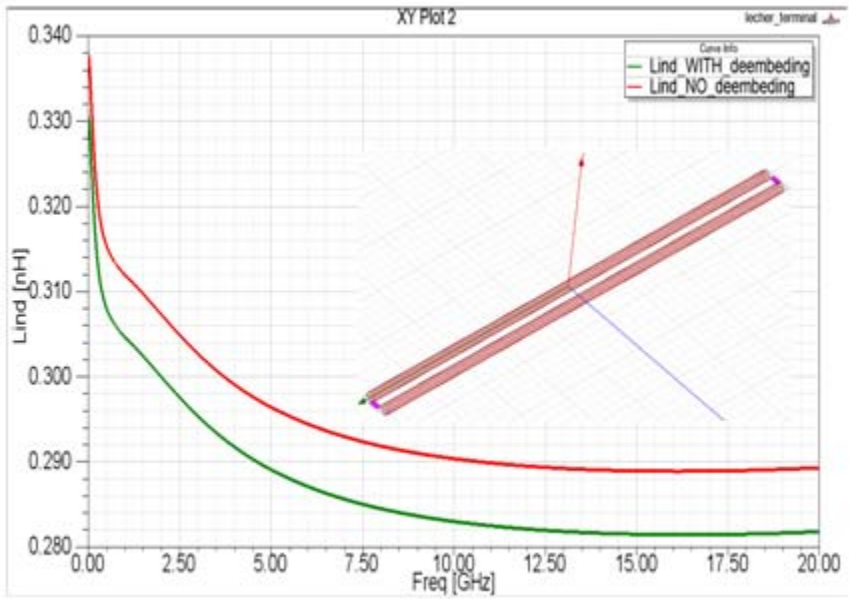


Figure 6: Loop inductance of a pair of cylindrical wires with and without deembedding the parasitic gap port inductance

Conclusions

New numerical constructs are developed for the causal modeling of transient signals in PCBs and interconnects. By introducing the correct treatment of metal traces of finite thickness, accurate DC point solution, causal material models, causal surface roughness models and deembedding parasitic effects of lumped ports, result in accurate and causal broad band frequency sweeps of S parameters. Using these techniques in a finite element code generates accurate and causal simulations provided the mesh is converged to reasonably small tolerance. The techniques described here are implemented in ANSYS HFSS. Real-life problems were solved to demonstrate the efficacy of the methods.

Related Topics

Best Practices for Accurate and Causal Broad Band Frequency Sweeps

References for Accurate and Causal Broad Band Frequency Sweeps

References for Accurate and Causal Broad Band Frequency Sweeps

[1] B. Young and A. Bhandal, "Causality Checking and Enhancement of 3D Electromagnetic Simulation Data," Electrical Performance of Electronic Packaging and Systems (EPEPS), 2010 IEEE, 19th Conference, Conference Proceedings, pp. 81-84.

[2] J.E. Bracken, "A Causal Huray Model for Surface Roughness," submitted to DesignCon 2012.

[3] A. R. Djordjevi, R. M. Bilji, V. D. Likar-Smiljani?, and T. K. Sarkar,"Wideband frequency-domain characterization of FR-4 and time-domain causality," IEEE Tran. Electromagnetic Compatibility., vol. 43, pp, 662-667, Nov. 2001.

[4] S. H. Hall and H. L. Heck, High-Speed Digital Designs, John Willey & Sons, Inc.,Hoboken, New Jersey, 2009.

[5] I. Vago and M Gyimesi, Electromagnetic Fields, Academiai Kiado, Budapest, Hungary, 1998.

[6] S. Groiss, "Numerical Analysis of Lossy Cavity Resonators," (In German) PhD Theses, Technical University of Graz, Austria, June 1996.

[7] F. W. Grover, Inductance Calculations, Dover Publications, Inc., Mineola, New York, 1946, 1973.

[8] C. R. Paul, Inductance, Loop and Partial, John Willey & Sons, Inc.,Hoboke, New Jersey, 2010.

[9] P. Pathmanathan, C.M. Jones, S.G. Pytel, D.L. Edgar, P.G. Huray, "Power Loss due to Periodic Structures in High-Speed Packages and Printed Circuit Boards", European Microelectronics Packaging Conference - IMAPS 2011, Brighton, England, September 12 - 15, 2011

Related Topics

Best Practices for Accurate and Causal Broad Band Frequency Sweeps

Modified Hammerstad Model

This section describes the modified Hammerstad model and how it is expressed as a function of frequency. Hammerstad model is strictly mathematical. It is possible to map the Hammerstad surface roughness data to a physics-based Huray model. This section shows how the Hammerstad model is mapped to the physics-based Huray model.

The modified Hammerstad model was discussed in the paper *"Which one is better? Comparing options to describe frequency dependent losses", by E. Bogatin, D. DeGroot, P. G. Huray and Y. Shlepnev*, in the DesignCon 2013 proceedings. It is described by two parameters, the RMS surface roughness Δ and the scale factor *SF*. In the original Hammerstad-Jensen model, the value of $SF = 2$.

The model equation is as follows:

$$H_{HJ}(f) = 1 + \left(\frac{2}{\pi}\right)(SF - 1)arctan\left[1.4\left(\frac{\Delta}{\delta(f)}\right)^2\right] \quad (1)$$

Here $H_{HJ}(f)$ is the ratio of the losses in a rough surface to the losses in a smooth surface of equal area.

Skin Depth $\delta(f)$ is given by

$$\delta(f) = \frac{1}{\sqrt{\pi f \mu \sigma}}$$

Therefore the model equation reduces to

$$H_{HJ}(f) = 1 + \left(\frac{2}{\pi}\right)(SF - 1)arctan[1.4\pi\Delta^2\mu\delta f] \quad (2)$$

Define the model's "corner frequency" f_{HJ} as

$$f_{HJ} = \frac{1}{1.4\pi\Delta^2\mu\sigma} \quad (3)$$

Therefore the modified Hammerstad model can be written as

$$H_{HJ}(f) = 1 + \left(\frac{2}{\pi}\right)(SF - 1)arctan\left[\frac{f}{f_{HJ}}\right] \quad (4)$$

Note that at low frequencies the value of $H_{HJ}(f)$ approaches 1. At high frequencies the value approaches *SF*.

Related Topics

Huray Model

Procedure for Obtaining Broadband Matching between Huray and Hammerstad Models

Huray Model

The Hall Huray model (Huray model for short) in its simplest, single-snowball form, is characterized by three parameters:

- snowball radius a
- number of snowballs N in a unit cell
- area A of that unit cell

The model equation is as follows:

$$H_{HH}(f) = 1 + \left(\frac{3}{2}\right)\left(\frac{4\pi Na^2}{A}\right)\left(\frac{1}{1 + \frac{\delta(f)}{a} + \frac{1}{2}\left(\frac{\delta(f)}{a}\right)^2}\right) \quad 1$$

This equation can be simplified by defining the Hall-Huray surface ratio SR as: $SR = \frac{4\pi Na^2}{A}$.
Using the definition of the Hall-Huray surface ratio in equation 1, it simplifies to the following:

$$H_{HH}(f) = 1 + \left(\frac{3}{2}\right)(SR)\left(\frac{1}{1 + \frac{\delta(f)}{a} + \frac{1}{2}\left(\frac{\delta(f)}{a}\right)^2}\right) \quad 2$$

At low frequencies, the value of $H_{HH}(f)$ approaches 1, while at high frequencies it approaches the value of $1 + \frac{3}{2}(SR)$.

But skin depth $\delta(f)$ is given by $\delta(f) = \frac{1}{\sqrt{\pi f \mu \sigma}}$

Therefore the quantity $\frac{\delta(f)}{a} = \frac{1}{a\sqrt{\pi f \mu \sigma}} = \sqrt{\frac{f_{HH}}{f}}$

where $f_{HH} = \frac{1}{\pi a^2 \mu \sigma}$.

Therefore the Huray model can be explicitly written as a function of frequency as follows:

$$H_{HH}(f) = 1 + \left(\frac{3}{2}\right)(SR)\left(\frac{\frac{f}{f_{HH}}}{\frac{f}{f_{HH}} + \sqrt{\frac{f}{f_{HH}}} + \frac{1}{2}}\right) \quad (3)$$

Related Topics

Modified Hammerstad Model

Procedure for Obtaining Broadband Matching between Huray and Hammerstad Models

Procedure for Obtaining Broadband Matching between Huray and Hammerstad Models

This section contains the procedure that obtains broadband matching between Huray and Hammerstad models.

Note: Before reading this section, see the sections Huray Model and Modified Hammerstad Model for the definitions of Scale Factor SF and to see how each of these models is expressed as a function of frequency.

At high frequencies, the Hall-Huray surface roughness ratio approaches the value of $1+\frac{3}{2}(SR)$.

The simplest way to create a Hammerstad model that is, in some sense, close to a Huray model is to ensure that the maximum values of the models are the same.

For the maximum values of the models to be same, the scale factor should be $SF = 1+\frac{3}{2}SR$.

Fit the Huray model to the Hammerstad model parameters so that $SR = \frac{2}{3}(SF-1)$.

Then, try to line up the models at their 50% crossing points. For the Hammerstad model, that occurs when

$$1+\left(\frac{2}{\pi}\right)(SF-1)arc\tan\left(\frac{f}{f_{HJ}}\right) = \frac{1+SF}{2} \quad (1)$$

It is easy to show that this occurs when $f = f_{HJ}$.

For the Huray model, the 50% point is reached when

$$\frac{\frac{f}{f_{HH}}}{\frac{f}{f_{HH}}+\sqrt{\frac{f}{f_{HH}}}+\frac{1}{2}} = \frac{1}{2} \quad (2)$$

By solving for f, the equation reduces to the following:

$$f = f_{HH}\left(1+\frac{\sqrt{3}}{2}\right) \quad (3)$$

Therefore for the 50% points to agree, the following condition must be realized.

$$f_{HJ} = f_{HH}\left(1+\frac{\sqrt{3}}{2}\right) \cong 1.866 f_{HH} \quad (4)$$

Using equation 4 and relating back to the model parameters, the following equation is obtained.

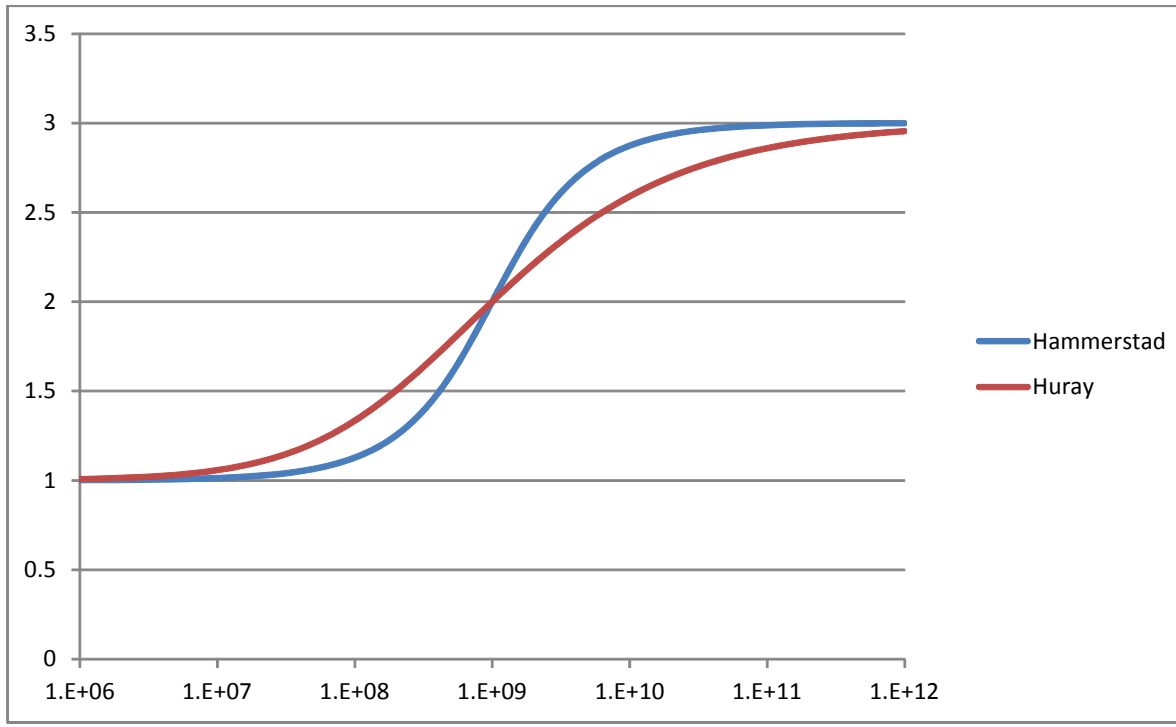
$$\frac{1}{1.4\pi\Delta^2\mu\delta} = \frac{1.866}{\pi a^2\mu\sigma} \quad (5)$$

Solving equation 5, a = 1.616Δ.

Compare the two models. For the Hammerstad model, set SF=3 and Δ=1.766m and let the material be copper. This conveniently yields f_{HJ}=1 GHz.

Using the above relations, the corresponding Huray model has SR=4/3 and a=2.854μm. Plots of $H_{HJ}(f)$ and $H_{HH}(f)$ are shown in the following figure.

Figure 1 Comparison of Hammerstad and Huray models



The figure demonstrates the expected match between the two curves at the 50% point (1 GHz), as well as at the low and high frequency extremes. It is also clear that the Hammerstad model transitions between low and high frequency behaviors more rapidly than the Huray model. This is inherent to the mathematical functions used in the model. Since the Huray model is based on the physics of the underlying loss mechanism, while the Hammerstad model is simply a curve fit, it is reasonable to expect the Huray model to be more realistic.

The above model-matching procedure potentially may be critiqued as follows:

- The matching is exact at some (three) points and is decent across the entire frequency range.
- The computed snowball radius is actually greater than the RMS surface roughness value. This does not seem realistic.

An alternative procedure would be to try matching the two models at low frequencies. The rationale for doing this is that the loss measurements used to derive a Hammerstad model are likely to be taken mainly at frequencies well below the 50% point. Therefore the higher frequency values might not be given as much weightage in the matching process. This approach is described in the following section.

Procedure for Obtaining Better Low Frequency Matching

Observe from the above plots in Figure 1 that the low frequency match could be improved by shifting the Huray model curve to the right. Such a shift can be accomplished by reducing the value of the snowball radius; this will increase the Huray corner frequency f_{HH}. See the equations in the above section **Procedure for Obtaining Broadband Matching between Huray and Hammerstad Models.**

However, an increase in f_{HH} would compromise the high frequency matching. The deviation in the high frequency match (at least over a limited frequency range) can be compensated for by increasing the maximum value of the Huray model, which can be accomplished by raising the value of SR.

No unique method exists to do this since there are many different frequency ranges and error metrics that could be used. Results obtained from some trial and error experiments reveal that using the broadband parameter values (see **Procedure for Obtaining Broadband Matching between Huray and Hammerstad Models**) $SR = \frac{2}{3}(SF - 1)$ is a good starting point.

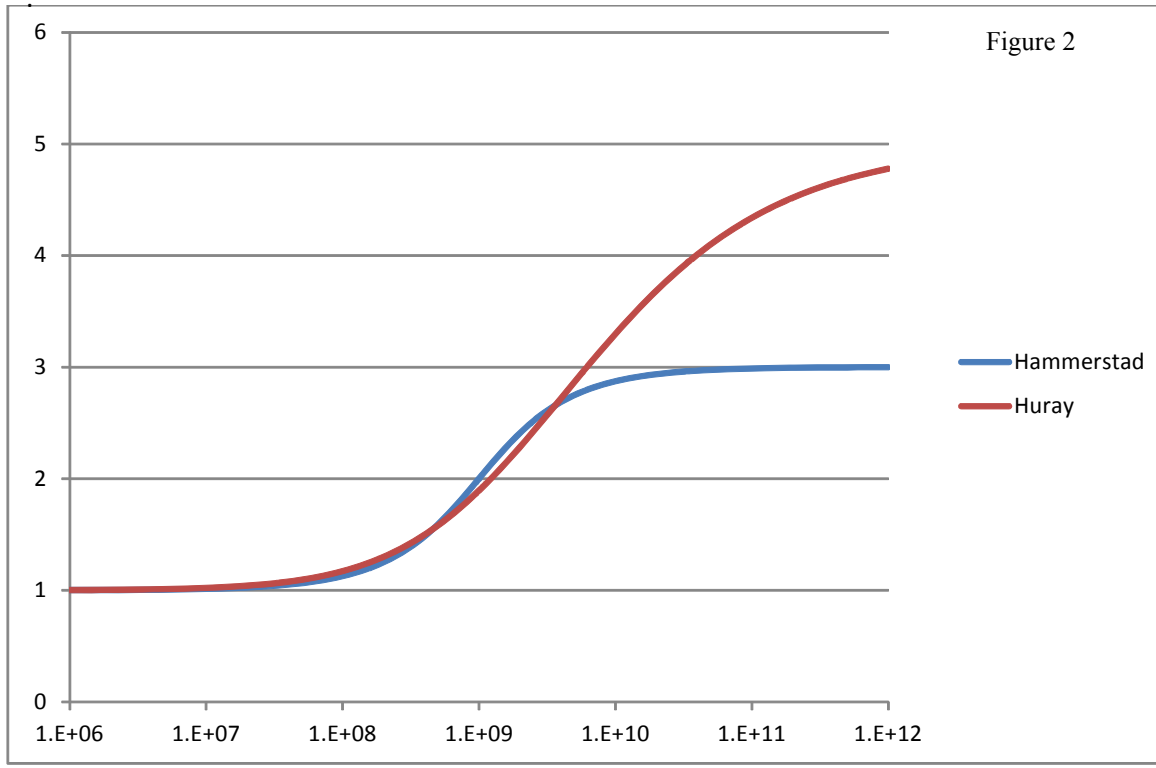
A match qualitatively similar to that shown in the following figure, that appears in the paper *"Which one is better? Comparing options to describe frequency dependent losses", by E. Bogatin, D. DeGroot, P. G. Huray and Y. Shlepnev* in the DesignCon 2013 proceedings, was obtained by choosing:

$a = 0.65\Delta$

$SR = \frac{4}{3}(SF - 1)$

Returning to the example in the section **Procedure for Obtaining Broadband Matching between Huray and Hammerstad Models**, set a = 1.148 μ m and SR = 2.667. The resulting Huray model is compared to the Hammerstad model in the following figure.

Comparison of Hammerstad and Huray model with low-frequency matching emphasized at the expense of high frequency



The match between the two is very good up to about 4 GHz, well beyond the 50% crossing point of the Hammerstad model at 1 GHz. Beyond this frequency, the fitted Huray model predicts significantly higher losses (66% greater) than the Hammerstad model. This is the price paid for the improved low frequency match. Of course it was assumed that the Hammerstad model may not be accurate in this higher frequency range anyway.

Note: It is interesting to note that the DesignCon paper does not provide a plot similar to Figure 1, but it does provide one similar to Figure 2. This may give the mistaken impression that a very high quality broadband match between the two models can be obtained.

To summarize, the Hammerstad model is fundamentally different from the Huray model. Since Hammerstad is strictly a mathematical model and the Huray model is a physics-based model, the conversion between the two is not exact. An approximate conversion is indeed possible, as shown, and the choice of conversion process depends on whether a broadband or a low-pass match is desired. Two conversion schemes have been described here, but a continuum of different choices is available. Expect that any usable conversion scheme would have Huray model parameters that satisfy the following bounds:

$$0.65\Delta \le a \le 2\Delta \quad (1)$$

$$\frac{2}{3}(SF-1) \le SR \le \frac{3}{2}(SF-1) \quad (2)$$

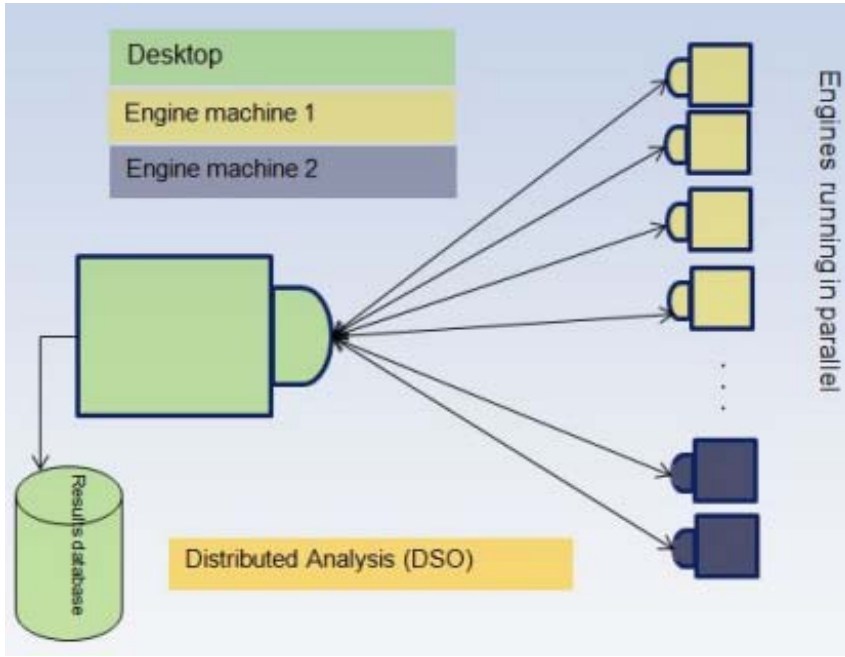
Related Topics

Huray Model

Modified Hammerstad Model

Large Scale DSO Theory

The parametric analysis command in Desktop computes simulation results as a function of model parameters, such as geometry dimensions, material properties and excitations. The parametric analysis is either performed on a local machine, where each variation is analyzed serially by a single engine, or distributed across machines through a DSO license. Desktop's DSO analysis runs multiple engines in parallel, thus generating results in a shorter time. In the Regular DSO algorithm, the parametric analysis job (Desktop) runs on master node, which in turn launches one or more distributed-parallel engines on each machine allocated to the job. Desktop distributes parametric variations among these engines running in parallel. As variations are solved, the progress/ messages and variation results are sent back to Desktop, where they are persisted into the common results database on master node, as illustrated below:



Regular DSO Bottleneck

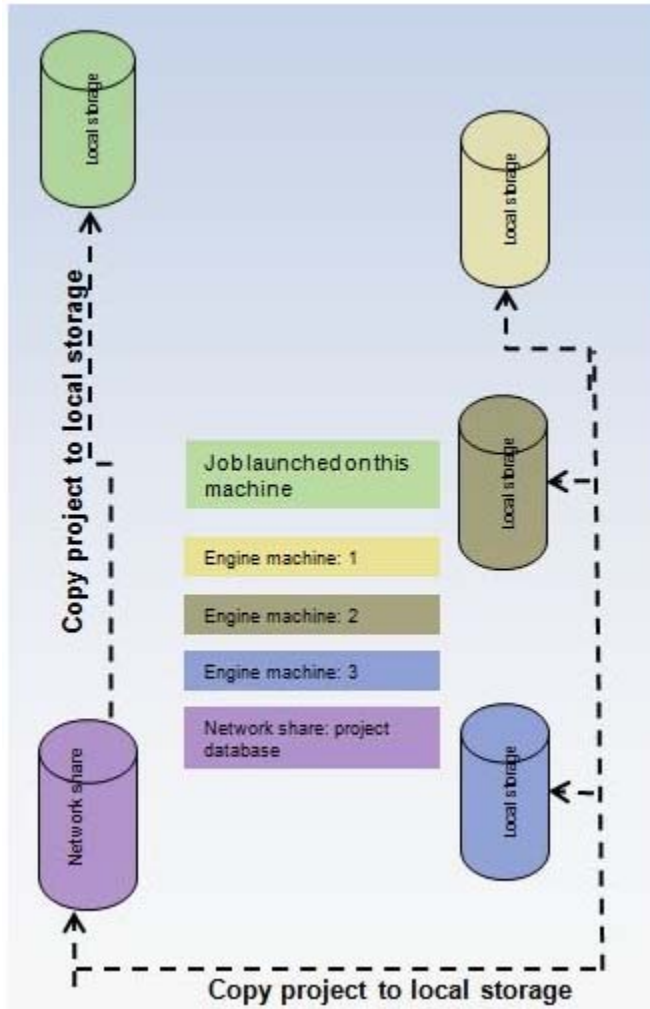
As per above illustration, Regular DSO's speedup is limited by the resources of the centralized 'Desktop' bottleneck. It's been observed that DSO becomes unreliable at a certain point, as the number of engines and number of variations is increased. The term 'large scale parallel' can be used to define this tipping point. For a given model, a 'large-scale parallel' job denotes scenarios, in terms of the number of distributed-parallel engines and the number of parametric variations, where the Regular DSO runs into centralized bottlenecks that result in one or both of: progressively smaller speedups, unreliability.

With the advent of economical availability and timely provisioning of compute resources, product designers have access to large compute clusters to run their simulations. And they are throwing larger and larger number of compute resources at simulation jobs, in order to obtain results faster. The parametric DSO needs to meet this challenge and target linear speedup for 'large-scale parallel' jobs. The Large Scale DSO feature is targeted towards 100% reliability and linear speedup of large-scale parallel DSO jobs.

Key Algorithms/Concepts for Large Scale DSO

- **Embarrassingly Parallel algorithm**: Large Scale DSO exploits the embarrassingly parallel nature of parametric runs. The solve of each variation is made fully independent of another variation, by replacing Regular DSO's centralized database with per-engine distributed databases
- **Distributed databases**: For each distributed-parallel engine, the database of the input model is cloned to local storage of engine's compute node (as illustrated in the following picture). Parallel analysis is performed on these cloned models and the analysis results also go the corresponding location on local storage. As each engine has it's own database, there is no 'shared database' contention between any two engines. There is negligible network traffic as analysis/

computations are contained in a single machine and use just the 'local resources'.



- **Hierarchical activation of engines**: In the Large Scale DSO algorithm, engine activation is done hierarchically, where the overhead of launching of engines is also shared across the nodes. Such hierarchical activation improves reliability of the activation phase of large-scale parallel jobs. A new 'desktopjob' program encapsulates hierarchical activation, as illustrated in the below pictures. In this approach, the 'root' desktopjob (Root DJ) activates a 'level-one' child desktopjob (DJ(L1)) for each unique node allocated to the DSO job. Each level-one desktopjob activates one or more 'leaf' desktopjobs (DJ(L2)) equal to the number of distributed engines per node. A leaf desktopjob in turn runs Desktop in batch mode, to perform local-machine parametric analysis to solve the variations assigned to this engine.
- **Decentralized Load balancing:** A parametric table is divided into regions of equal number of variations, with the number of regions equal to the number of engines. In the below illustra-

tion, the analysis of 30 parametric variations is distributed among 5 engines. Each engine solves it's assigned region as a 'local machine' parametric analysis. Large Scale DSO job is considered done once all engines are done with their assigned variations.

- **Distributed results postprocessing**: As engine is done with the solve of a variation, it extracts results for the solved variation before progressing to the analysis of next variation. The extracted results are saved to the local storage. When the engine is done with analysis of all variations, the extracted results are transferred from local storage to the results folder of the input project.

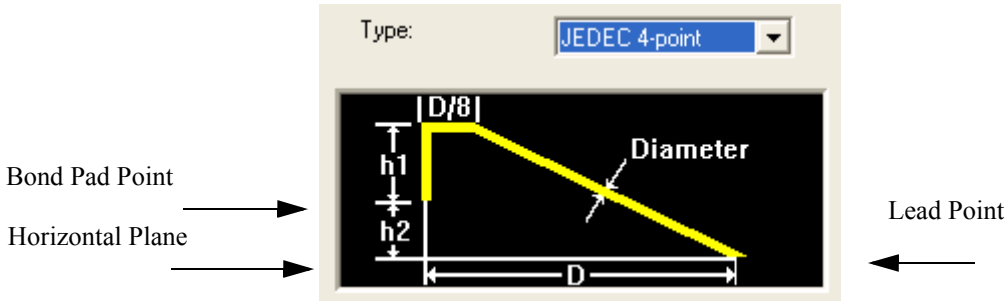
Geometric Objects

Following are supplemental technical details about working with geometric objects and complicated models:

- Bondwires
- Healing and Meshing
- Detecting and Addressing Model Problems to Improve Meshing
- Handling Complicated Models

Bondwires

A bondwire is a thin metal wire that connects a metal signal trace with a chip. You can choose to draw a standard **JEDEC 4-point** bondwire, as shown below:



where

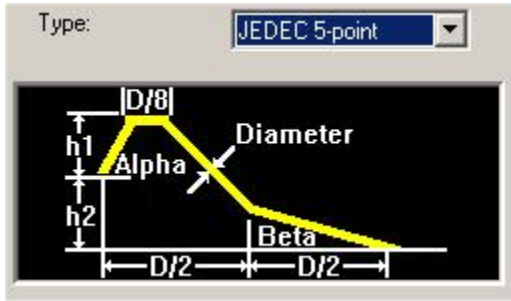
h1 = the height between the bond pad point and the top of the loop.

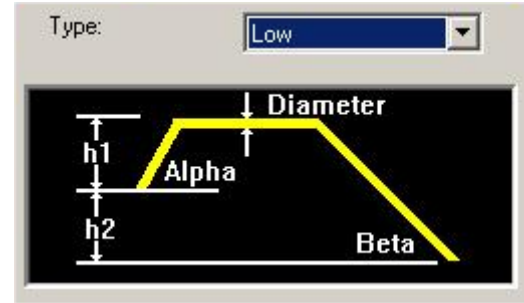
h2 = the height between the lead point and the bond pad point.

diameter = thickness of the wire.

D = the distance between the start and end points, and D/8 is the distance divided by 8.

Or you can choose to draw a **JEDEC 5-point** or **Low** bondwire, as shown below:





where

α = the angle between the horizontal plane and the wire at the bond pad point.

β = the angle between the horizontal plane and the wire at the lead point.

D = the distance between the start and end points, D/2 is the distance divided by 2, and D/8 is the distance divided by 8.

When drawing the bondwire, you first select the bond pad point, a point in 3D space that defines the bond pad position in a horizontal plane. Then you select the lead point, which indicates the distance the wire covers in the horizontal plane. HFSS will use the distance between the bond pad and lead points to calculate the height between the bond pad and the lead point, or *h2*, a value that you can modify in the **Bondwires** dialog box.

For the JEDEC types, notice that the horizontal distance on the wire is calculated as the total length divided by 8. (This horizontal distance is shown in the JEDEC 4 and 5-point figure as "D/8.") For the Low type, the horizontal distance is affected by the Alpha and Beta values as well as the length.

Related Topics

Drawing Bondwires

Healing and Meshing

Potential problems with 3D Models

This section lists problems that can prevent a 3D model from being meshed successfully. Subsequent sections will describe how these problems can be detected and addressed.

ACIS errors

The underlying solid modeling technology used by ANSYS EM's 3D products Maxwell, HFSS and Q3D is provided by the ACIS geometric modeler. You can create models directly in the drawing environment of these products using primitives, such as boxes, cylinders, etc. and operations on primitives, such as Boolean operations. In addition, you can import models produced by other CAD tools in a variety of formats such as STEP, IGES, etc. In ANSYS EM's 3D products, all models have to be stored internally in ACIS' native format, known as sat format. When you import models into ANSYS EM products, translators are invoked that convert the models to sat format. Often, models that were created in other CAD tools were created initially for other purposes than electromagnetic analysis, such as for mechanical design or just for display purposes. They may have imperfections that make them illegal to ACIS. Further, there can be compatibility issues between different versions and even flavors of modeling tools. All this can lead to errors in imported 3D models.

If you use ANSYS EM products to create geometry models, and thereby avoid model import and translation, you are unlikely to encounter such problems.

Note Owing to the ACIS tolerance acis MinD() of 1.0e-06, the default range of dimensions is [1.0e-06 1.0e04]. We recommend that you keep the dimensions in the [1.0e-03 1.0e04] range, which covers 7 orders of magnitude. This helps ensure 1 mm accuracy for 10km sized models and 1 micron accuracy for 10 m sized models. Determine the unit in which the smallest critical features of the model such as gaps, trace thicknesses, trace widths etc will be 0.001. Select that unit for the model so that it falls within the range.

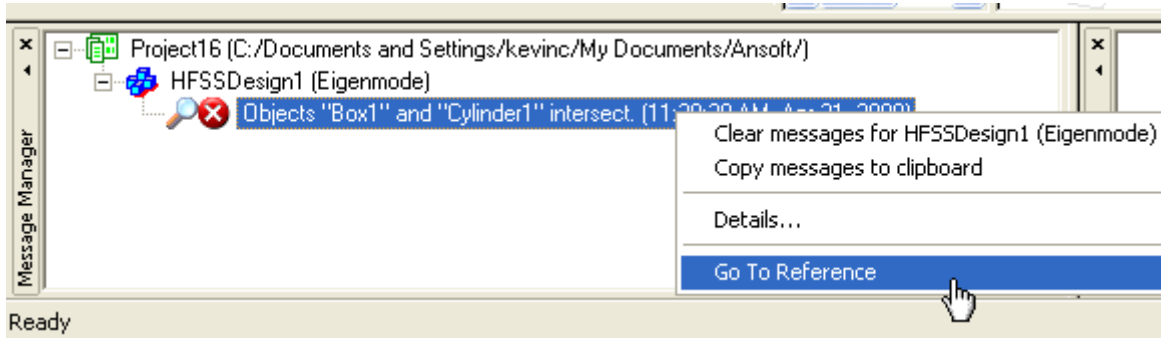
Mixed dimensionality

Even if a model is imported and translated without errors, there is a restriction to be aware of. ACIS can handle mixed-dimensionality models. One of the goals of ANSYS EM's use of the ACIS modeling system is to create a valid volumetric mesh for simulation. Mixed-dimensionality models will not yield a valid volumetric mesh. Therefore, the ANSYS EM tools will not mesh objects with mixed dimensionality, so-called non-manifold objects. For instance, imagine a 3D object representing a curved metal plate with a small but finite thickness. If it reaches zero thickness somewhere while having non-zero thickness elsewhere, it has mixed dimensionality, 2D as well as 3D. You will get an error message saying that the object is non manifold. Of course, 2D and 3D objects can co-exist in a model, but any one object cannot be both 2D and 3D.

Intersecting objects

Another restriction is that ANSYS EM 3D tools don't allow partial intersections (also known as partial overlaps) between 3D objects. Each element of the mesh has to belong unambiguously to one object. There is no problem if one object is enclosed completely inside a bigger object, but partial intersections lead to ambiguities. As long as there are partial object intersections, the mesh generator will not attempt to create a mesh. Instead, you will get an error message notifying you which objects are intersecting.

After you run a validation check, you can right-click on an intersection error message in the Message window, and select Go to reference from the shortcut menu. This selects the intersecting objects.



You must remove the intersections before you can proceed. You can do this by changing the shapes of objects slightly, or by subtracting one object from the other.

Caveat: if as a result of a subtraction the model has pairs of true surfaces that are coincident, that is, smooth curved surfaces that fit exactly one inside the other, you make it harder for the mesh generator to create a mesh. This is because ACIS will create segmentations on each of these surfaces, and these segmentations are not guaranteed to fit. Setting a small value for Surface Deviation under ***Mesh Operations >Assign>Surface Approximation*** *increases your chance of success in such a case, but it is better to avoid such situations if you can.*

Small features and misalignment

When there are no ACIS errors in the model, no non-manifold objects and no partial object intersections, the mesh generator can be invoked to create a valid mesh for the electromagnetic analysis. Even if the geometry is valid, mesh generation can still fail. Possible causes are the presence of very short edges, very small faces, long and thin sliver faces, and slight misalignments between faces that are supposed to be coincident.

Related Topics

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Technical Notes: Handling Complicated Models

Detecting and Addressing Model Problems to Improve Meshing

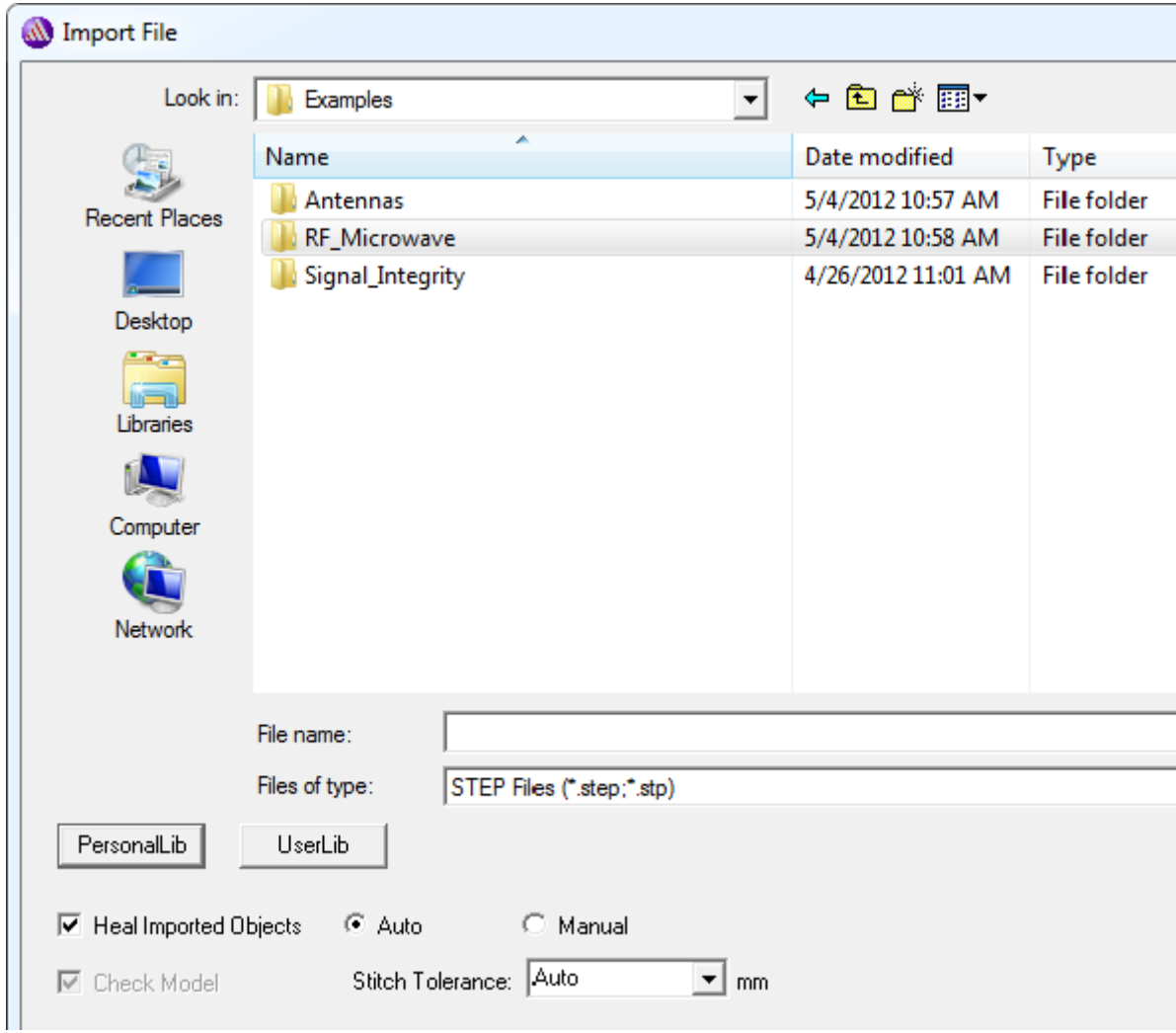
The following sections describe a systematic procedure to detect and address model problems that can interfere with the meshing process.

- Technical Notes: Healing During Geometry Import
- Technical Notes: Healing After Geometry Import
- Technical Notes: Removing Object Intersections
- Technical Notes: Removing Small Features
- Technical Notes: Aligning Objects
- Technical Notes: Troubleshooting if Meshing Still Fails

One: Healing during geometry import

In case you don't draw your entire geometry in the ANSYS EM environment but wish to import (part of) it, in the **Import File** window you select which geometry file to import. Some formats permit healing during import. These are: ACIS SAB files (*.sab), ACIS SAT file (*.sat), Ansoft 3D Modeler files (*.sm3), CATIA V4/V5 files (*.model, *.CATpart, *.CATProduct), IGES files (*.iges, *.igs), Parasolid files (*.x_t, *.x_b), ProE files (*.prt, *.asm), STEP files (*.step,*. stp), and Unigraphics files (*.prt). Selecting these formats enables a checkbox at the bottom of this window, "**Heal Imported Objects**." For these supported formats (except for 3D Modeler files), two modes exist, "auto" and "manual". Auto Healing will try to address ACIS errors and non-manifold errors, the first two classes of potential problems listed earlier. It will also fix surface normals in the body and updating orientation of body, to avoid having a body with negative volume.

Manual healing adds tolerant stitching, geometry simplification (such as small-feature removal) and tight en gaps settings to this. You can remove small features at this stage if you wish. However, the usual approach is to apply auto-healing at this stage and leave small-feature removal until later.



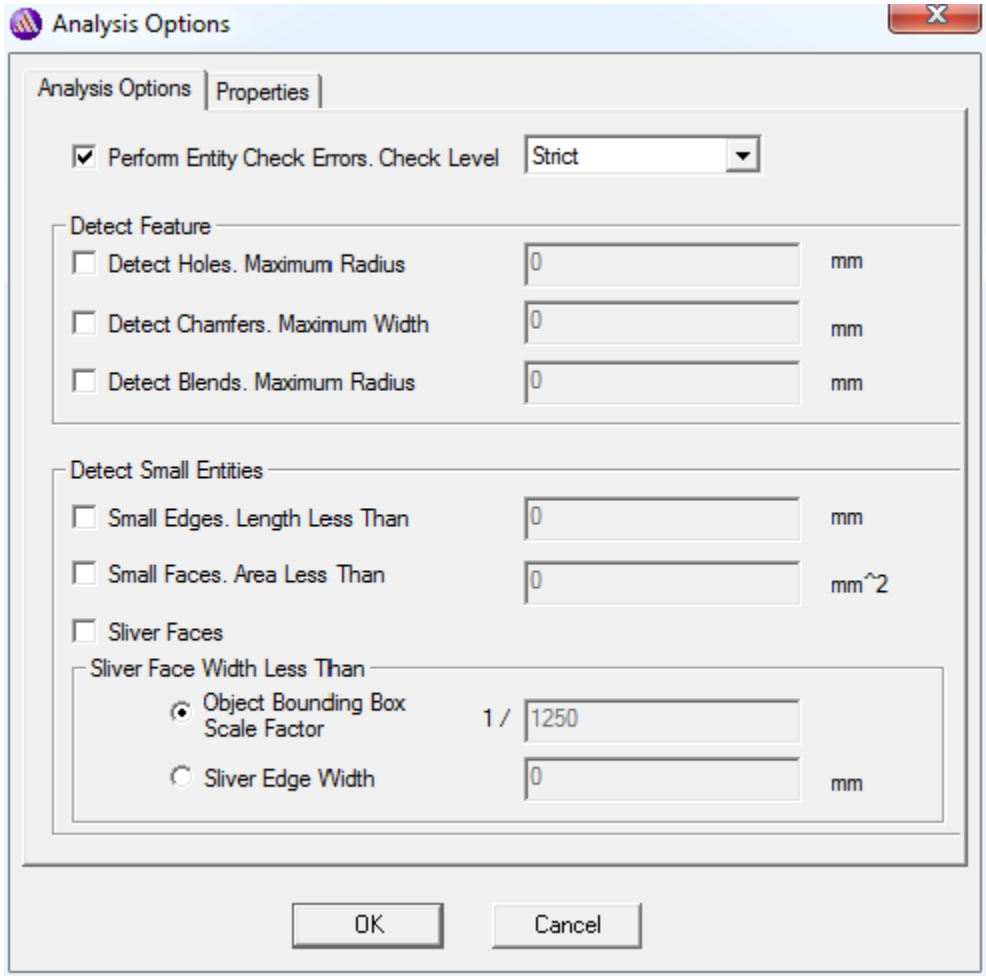
Two: Healing after geometry import

Healing can only be performed on objects that have no drawing history other than "import". If necessary, object history will be automatically deleted through **Modeler>Purge History**. If that causes a warning that another object will be deleted, you may need to purge the history of that other object first, or purge the histories of several objects simultaneously. If you need to preserve the object history, you should save a separate copy before purging the history.

At any time after import, you can perform a Validation Check: **HFSS>Validation Check**. This will enable you to focus on bodies and body pairs that need attention before a mesh can be created. Note that you can set the Modeler Validation settings for Warning Only, Basic, and Strong.

1. Select the objects that have ACIS errors, such as failing api_check_entity(), and the objects that have non-manifold features, i.e. mixed dimensionality.
2. Invoke **Modeler->Model Analysis->Analyze Objects**.

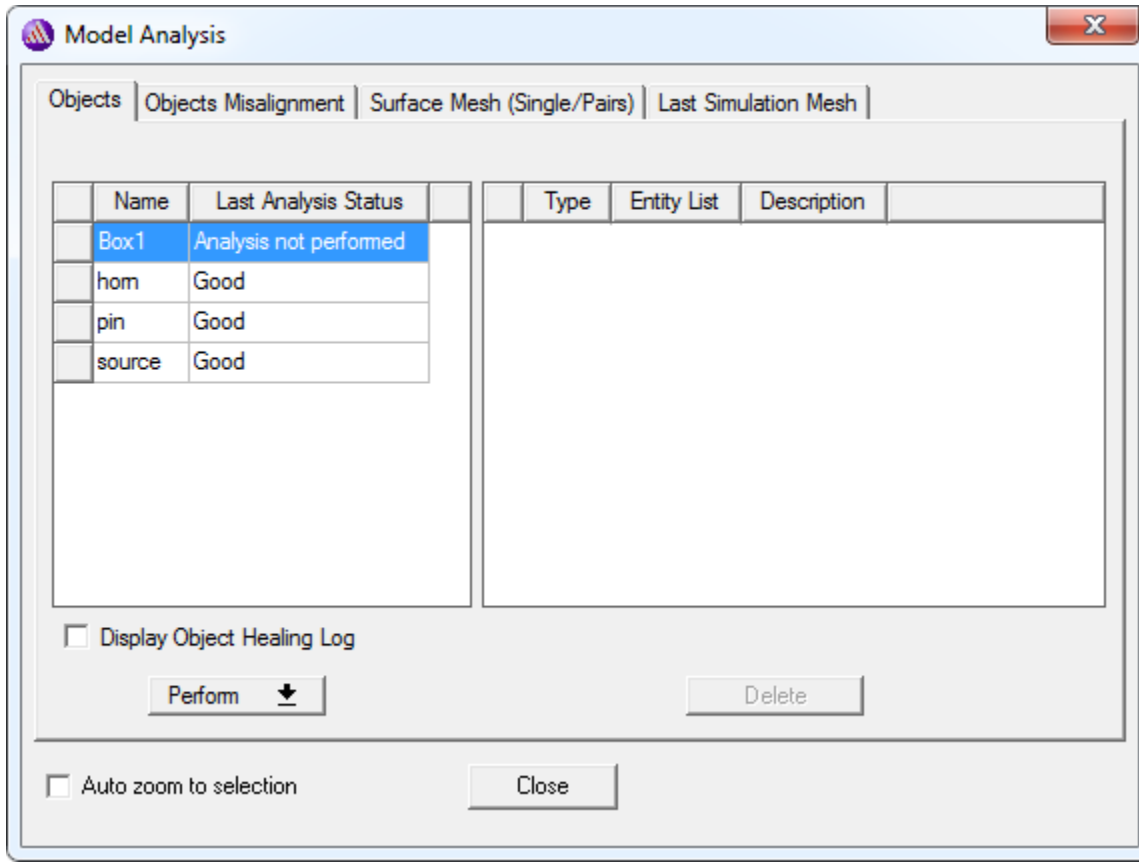
   This will bring up an Analysis Options dialog where you set the strictness of the entity check, detection and thresholds for holes, chamfers and blends, thresholds for small feature detection.



3. When you have made selections, click **OK** to start the analysis.

   On completion, the **Model Analysis** dialog is displayed. All bodies in the model are shown in the objects grid along with their status. Bodies can have the following status:

1. Good
2. Null Body
3. Analysis not performed
4. Invalid entities found
5. Small-entity errors



Invalid-entity errors are ACIS errors and non-manifold errors. Small-entity errors are small faces, sliver faces and small edges that are optionally detected based on user-defined parameters.

Note Invalid-entity errors must be fixed before a mesh can be generated.

To fix invalid entity errors:

1. Choose the bodies that have "Invalid Entities Found."
2. In the same **Model Analysis** window, choose **Perform>Heal Objects**, with or without an optional setting for small-feature removal.

   In most cases, the bodies will be healed and the errors fixed.

3. If errors still persist, choose "offending" faces and edges and click on **Delete**.

   This will replace the selected face/edge entity by a tolerant edge/vertex respectively.

In order to avoid unintended changes, it is good practice to do the following:

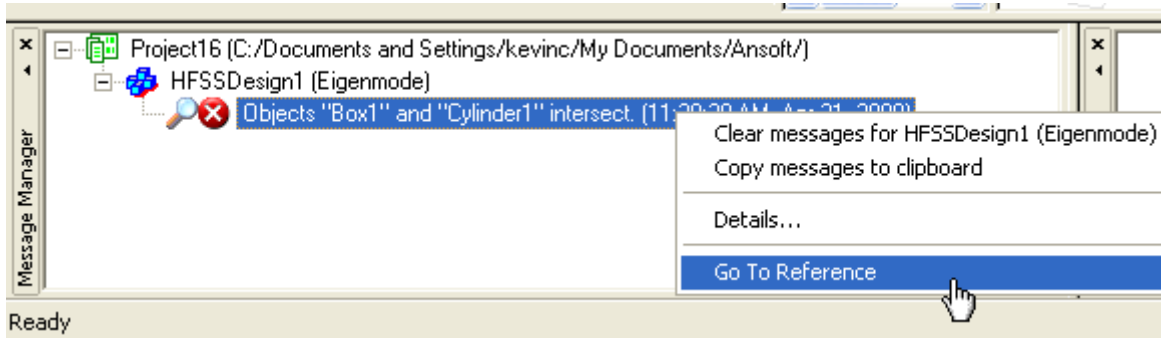
1. At the bottom of the **Model Analysis** window, check the box "Auto Zoom to Selection."
2. Select one face or edge at a time
3. Decide for each face and edge whether you want to delete it.

Note Healing causes changes to the geometry and topology of the body being healed. Validation check has to be re-run after healing is done to identify body pairs that intersect. It is possible that after healing, bodies that were disjoint before now overlap.

In some cases the replacement of the face/edge by tolerant edge/vertex will fail. If the object remains invalid, you know at this point what parts of the object are invalid. You will need to change that part of the object manually, either in ANSYS EM's drawing environment or in the original CAD tool, to make it pass. Often, the invalid entities are in small details that can be changed without noticeably affecting the results of the electromagnetic analysis. For example, it may be possible to create a small object, well placed in the "offending" region, and to unite it with or subtract it from the problematic object, such that the "offending" details don't exist anymore.

Three: Removing Object Intersections

If there are any intersecting objects, a Validation Check will list them. You must eliminate object intersections before a mesh can be created. After you run a validation check, you can right-click on an intersection error message in the Message window, and select **Go to reference** from the shortcut menu. This selects the intersecting objects.



In complicated models, before making changes, it is good practice to inspect the overlap visually. A way to do this is to:

1. Duplicate both objects and place the copies outside the model.
2. Perform Boolean Intersect on the copies.

   This will show you what causes the intersection and will help you decide how to remove it.
3. Then, delete the copies.

The easiest way to eliminate object overlap is to subtract one object from the other, in the order that leaves the desired material in the region of overlap. If the overlap is very small and you can choose the order of subtraction, choose one that does not create coincident true surfaces, if possible.

Caveat: if as a result of a subtraction the model has pairs of true surfaces that are coincident, that is, smooth curved surfaces that fit exactly one inside the other, you will make it harder for the mesh generator to create a mesh. This is because ACIS will create segmentations on each of these surfaces, and these segmentations are not guaranteed to fit. Setting a small value for **Surface Deviation** under **Mesh Operations>Assign>Surface Approximation** increases your chance of success in such a case, but it is better to avoid such situations if you can.

A way to eliminate object intersections without subtraction is to split one object in parts, in such a way that some parts are completely enclosed in the other object, and some parts are completely outside the other object. Even for complicated objects, this is possible through a sequence of Boolean operations on the objects and copies of the objects.

At this point, the geometry has no ACIS errors, no non-manifold objects and no partial object intersections. A mesh can be created for the electromagnetic analysis.

The procedure for fixing face improper intersections is similar. When it occurs, the error report lists a pair of faces intersecting.

1. You can detach those faces (using **Modeler>Surface>Detach Faces**), and others if necessary.
2. Select the faces and invoke either the **Modeler>Model Healing>Stitch Sheets** command (see Stitch Sheets) or **Modeler>Boolean>Unite** (see Uniting Objects).
3. Once the improper intersections have been corrected, you can stitch the set of faces back to the original model.

Four: Removing Small Features

Even though, in principle, the geometry may be ready for a mesh to be created, it is possible that small features in the geometry lead to a mesh that is unnecessarily large and contains long and thin tetrahedra that make the simulation converge slower. Small features may even cause the mesh generation to fail. By small, we mean details on an object that are thousands of times smaller than the main features of the object, and that, in most cases, are unintended consequences of the drawing history in another CAD tool. Therefore, it is advantageous to remove small features.

You may have noticed that you could have invoked small-feature removal at several earlier stages. There is no objection to doing it earlier. See Object Overlap Settings for Complicated Models. The reason why it is presented here as stage four is that the previous stages were necessary while this one is optional.

To start the small-feature removal:

1. Click HFSS>Modeler>Analysis Options and specify the Entity Check level, make selections for holes, chamfers, and blends, their thresholds, as well as for edges, faces and slivers.
2. Select objects and invoke object analysis through **Modeler>Model Analysis>Analyze Objects**.

   Alternatively, without objects selected, use **Modeler>Model Analysis>Show Analysis Dialog>Objects** and select objects from the list. In the **Model Analysis window**, invoke **Per-**

form>Analyze Objects.

The software will report according to the **Analysis Options** settings, such as holes smaller than a given radius, the smallest edge length and the smallest face area.

3. Upon clicking OK, the analysis is performed.

   As a result of the analysis, the software presents a list of all holes, chamfers, blends, faces and edges that don't meet the thresholds set by you in the Analysis Options.

4. Check the box "Auto-Zoom to Selection" at the bottom of the **Model Analysis** window and click on items in the list.

   Inspect them visually and decide whether they can be deleted. It is good practice to delete them one by one rather than deleting many at once in order to prevent unintended changes.

   Sometimes, an edge or face cannot be deleted, and you get a message notifying you. In that case, either ignore it, or revisit it after deleting some other details first, or revisit it later manually in the 3D drawing environment.

   At this point, the geometry has no ACIS errors, no non-manifold objects and no partial object intersections. Furthermore, there are fewer small features that were unintended or unimportant for the electromagnetic analysis, so the quality of the model has improved.

Five: Aligning Objects

Objects that touch each other in imported geometries don't always have well-aligned faces. Often, this is a consequence of the limited level of precision in the imported file. Misaligned faces can cause tiny object intersections or tiny gaps between objects, which in turn can lead to an inefficient mesh or even a failure to create the mesh.

To repair such occurrences in an automated way, you can select groups of objects and invoke **Modeler>Model Analysis>Analyze Interobject Misalignment**. This will yield face pairs from different bodies that are slightly misaligned with respect to each other.

In the window that shows this list, check the box "**Auto-Zoom to Selection**" and select face pairs from the list. When you decide that faces should be aligned, click Align Faces. In some cases, face alignment will fail if the topology of the body would change by a large amount after alignment. In that case, you can decide to ignore it, as it may not be a problem, or revisit it later manually in the Modeler environment.

Note In complicated models, the **Interobject Misalignment** analysis can take a long time if you select all objects before launching the analysis. If you don't know which pairs of objects to analyze, just let the mesh generator try to make a mesh. If the mesh fails, a list will be presented to you of misalignments that the mesh generator finds suspicious but didn't want to adjust without permission. Not every misalignment in the list is always a problem: this is a list of features that might need your attention.

Note As face misalignments between touching objects can cause small object intersections, this alignment capability can already serve a useful role in stage three.

Six: Troubleshooting if meshing still fails

If mesh generation fails, information about the reasons for the failure is presented under **Modeler>Model Analysis>Show Analysis Dialog>Last Simulation Mesh**.

Again, check the box "Auto Zoom to Selection" and click on the errors in the list. This can give you hints about which parts of the model are causing difficulties.

For instance, there may be self-intersecting bodies or faces. Such errors can have a variety of causes, such as a face that is supposed to be planar, but of which the vertices don't quite lie in the same plane. When you zoom and search you are likely to see what causes the problem.

Also, there may be face misalignments. Once you know they exist, you can inspect them and decide whether to align them under the Objects Misalignment tab.

One of the tabs of the **Model Analysis** window is the **Surface Mesh** tab. Under that tab, you can try to create surface meshes for objects and pairs of objects. Since a surface mesh on selected objects is easier to create than a volume mesh for the whole model, this can help you to identify quickly which objects are causing difficulties and why.

Also, in order to determine which objects are causing difficulties, you can exclude objects temporarily from the model. If the mesh succeeds without them, this helps to identify the reason for failure. To exclude an object temporarily, select it and uncheck "Model" in its properties window. Then try to create the mesh again.

Once you know which objects make the mesh fail, you can try to make small changes to them that don't affect the electrical properties noticeably but help the mesh maker succeed. For example:

- Zoom in on details and consider removing details;
- Find coincident true surfaces and move one of the faces over a very short distance so the pair of faces is not coincident anymore;
- Split very complicated objects into multiple less-complicated objects;
- Delete a complicated 3D ground object and create a 2D ground through a boundary condition on the appropriate faces of a dielectric;
- Replace imported objects by objects drawn in ANSYS EM's 3D modeling environment. For instance, some CAD tools produce cylinders that consist of two half cylinders that have a seam where they join. The fit is not always perfect.

Finally, for coincident true surfaces, set a very small value for **Surface Deviation** under **Mesh Operations>Assign>Surface Approximation**. ACIS will give them more segments, but you can compensate for that with **Model Resolution**. In a parametric sweep, you can experiment with settings for **Surface Deviation** and **Model Resolution**.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Validating Projects

Technical Notes: Handling Complicated Models

Handling Complicated Models

Complicated models, often imported from a CAD tool or layout tool, may slow down the interface, use a lot of RAM during file I/O and other operations, contain imperfections and object overlaps. After analysis, post processing of such models may be time consuming. The modeler has several options and features that address these problems.

- Interface Options for Complicated Models
- Geometry Imperfections and Complicated Models
- Object Overlap Settings for Complicated Models
- Post Processing Settings for Complicated Models

Note Remember that Healing always purges the history. If you heal an object for which you need to recover the history, use the Undo command. If you want to heal an object for which you want to keep the history, you should make a copy of the object for that purpose.

Interface Options for Complicated Models

To improve the speed of the interface when dealing with complicated geometries, do the following:

Under **Tools>Options>Modeler Options**, on the **Display** tab...

- Set "Default View Render" to "Wire Frame". Wire-frame rendering is faster than shaded rendering.
- Turn off "Display UV Isolines". For models with curved faces, this will simplify the wire-frame display, so the rendering will be faster.
- Turn off "Visualize History of Objects". This will remove visualization of objects that are part of the model history. For large models, this is faster and uses less memory.

Under **Tools>Options>HFSS Options**, on the **General** tab...

- Turn off "Visualize Boundaries on Geometry". When a boundary or excitation is selected, it will not be shown with a "pattern". This will prevent large delays when selecting.

Under **View>Visualization Settings**

- Use larger deviations to view curved objects in less detail.

Under **Modeler>Import**

- Un-check "Check Model" and "Heal Imported Objects." This helps for complicated models: Validation and healing take considerable time for such models. Use this option to defer checking to a later stage (especially in cases where you know that you want to mesh the model as is).

Under **Tools>Options>General Options**, on the **Project Options** tab

- Turn off "Do Autosave" or set the autosave interval to a larger value, e.g. 50. Auto-save can be time consuming.

Geometry Imperfections and Complicated Models

Modeler>Import

Many formats can be handled. It is recommended to import a version of the geometry that is as close as possible to its source, rather than geometries that have been translated before from one format to another, or that have been imported into another computational tool and later exported from it.

Modeler>Validation Settings

Geometry imperfections are listed as ACIS errors when executing **HFSS>Validation Check** and when starting an analysis. It is recommended to attempt to heal objects with such errors. However, HFSS enables you to bypass the errors (not the check itself) by choosing a setting under **Modeler>Validation Settings**. "Warning Only" enables you to ignore all errors. "Basic" enables you to bypass all but the most severe errors. The HFSS mesh generator has been enhanced to handle many geometry errors.

Object Overlap Settings for Complicated Models

HFSS>Set Material Override

Complicated geometries often have small object overlaps. This option allows some intersections to be resolved automatically in the mesh. If metal intersects dielectric, the metal will override the dielectric in the region of the overlap. If objects with the same material intersect, the smaller object will override the larger. All other intersections will still be treated as errors.

Post Processing for Complicated Models

Under **Tools>Options>General Options**, on the **Miscellaneous Options** tab...

- Turn off "Dynamically update postprocessing data during edits". This will disable expensive updating of existing reports and plots.
- Turn off "Update reports on file open". This will disable expensive updating of reports and plots when opening a project.

Under **HFSS>Fields>Modify Plot Attributes**, on the Plots tab…

- Set Plot Quality to Coarse, and save as default. This will make field plots much faster. The fields will not be as smoothly approximated within each tetrahedron, but this should not be noticeable on very large meshes.

Boundaries

Boundary conditions specify the field behavior on the surfaces of the problem region and object interfaces. This area of the technical notes includes information about the following boundary types:

- Perfect E
- Impedance
- Radiation
- PML
- Finite Conductivity
- Symmetry
- Master and Slave
- Lumped RLC
- Anisotropic Impedance
- Layered Impedance
- Infinite Ground Planes

and the following subjects:

- Frequency-Dependent Boundaries
- Default Boundary Assignments

Perfect E Boundaries

In HFSS, perfect E boundaries represent perfectly conducting surfaces in a structure.

By default, all HFSS model surfaces exposed to the background are assumed to have perfect E boundaries; HFSS assumes that the entire structure is surrounded by perfectly conducting walls. The electric field is assumed to be normal to these surfaces. The final field solution must match the case in which the tangential component of the electric field goes to zero at perfect E boundaries.

The surfaces of all model objects that have been assigned perfectly conducting materials are automatically assigned perfect E boundaries.

Note Currents from a Symmetric E boundary will not be included in the ZPI calculation when there is only 1 conductor on the port. Otherwise, the currents will be included in ZPI calculations.

Impedance Boundaries

In HFSS, impedance boundaries represent surfaces of known impedance. The behavior of the field at the surface and the losses generated by the currents flowing on the surface are computed using analytical formulas; HFSS does not actually simulate any fields inside the resistor.

Similar to finite conductivity boundaries, the following condition applies at impedance boundaries:

$$\boldsymbol{E}_{tan} = Z_s(\hat{n} \times \boldsymbol{H}_{tan}) \quad (1)$$

where

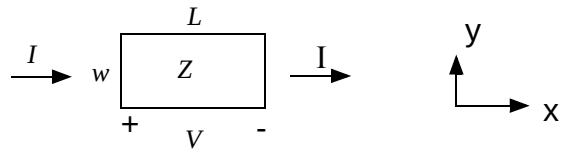
- $\hat{n}$ is the is the unit vector that is normal to the surface.
- E_{tan} is the component of the E-field that is tangential to the surface.
- H_{tan} is the component of the H-field that is tangential to the surface.
- Z_s is the surface impedance of the boundary, $Rs + jXs$, where
  - Rs is the resistance in ohms/square.
  - Xs is the reactance in ohms/square.

For example, assume that a structure contains two dielectrics separated by a thin-film resistor. This resistor could be represented by an impedance boundary at the surface between the two objects.

Units of Impedance Boundaries

Impedance on the surface of objects, Z_s, has units of ohms per square. The units *ohms per square* indicate that the impedance, Z_s, is equal to the equivalent circuit impedance, Z, measured between the edges of a square sheet of the material.

For example, a rectangle of length *L* and width *w* has a uniform current, *I*, applied to it. It has a voltage drop, *V*, across it and an equivalent circuit impedance of *Z* ohms.



If the current density, *J*, is uniform over the rectangle then the equation (1)

$$\hat{n} \times \vec{E} = Z_s \hat{n} \times \vec{J} \quad (1)$$

becomes equation (2)

$$E = Z_s J \quad (2)$$

where

- $E = |\vec{E}|$ on the rectangle, and
- $J = |\vec{J}|$ on the rectangle.

The circuit quantities and fields are related as follows:

$$V = \int_{x=0}^{L} \vec{E} \cdot d\vec{L} = EL \quad (3)$$

$$I = \int_{y=0}^{w} \vec{J} \cdot \hat{x} dy = Jw \quad (4)$$

$$Z = \frac{V}{I} = \frac{EL}{Jw} \quad (5)$$

Substituting equation (1) into equation (5) results in the following equation:

$$Z = Z_s \frac{L}{w} \quad (6)$$

Thus, when $L = w$, the equivalent circuit impedance is equal to the impedance on one square. Hence the units *ohms per square*.

If in this example $L = 2w$, the impedance would be equal to one-half of the circuit equivalent impedance for the rectangle, or the circuit equivalent impedance of one "square" of the rectangle is equal to the impedance of that square. Therefore, when entering the surface impedance for an object, you must enter the impedance per square.

Radiation Boundaries

When solving radiating (i.e. antenna) and scattering (i.e. Radar Cross Section) structures in an unbounded, infinite domain, HFSS truncates the problem into a bounded domain and prescribes the appropriate truncation condition. This is generally known as the "radiation boundary condition". Theoretically, the radiation boundary condition should be a "transparent" condition. In other words it should not produce any unphysical reflection as a result of the artificial truncation. HFSS offers three types of radiation boundaries: first-order absorbing boundary condition (ABC), perfectly matched layers (PML), and boundary integral equations (IE).

ABC and PML Boundaries

Both the ABC and PML boundaries attempt to minimize reflections by absorbing all outgoing waves at the truncation boundary. Because of this, they can only be prescribed at convex surfaces. This is because for concave surfaces, outgoing waves will re-enter the problem domain and should therefore not be completely eliminated. While PMLs absorb any kind of waves including guided waves, ABC imitates radiation to homogeneous background space.

ABCs only absorbs normal or near normal incident waves. Thus in order to produce accurate results, it must be placed sufficiently far away from structures. The typical recommendation is at least a quarter wavelength from the radiating source, although in some cases the radiation boundary may be located closer than one-quarter wavelength, such as portions of the radiation boundary where little radiated energy is expected. ABC is a local condition and thus preserving the sparse nature of the FEM formulation.

PMLs absorb all outgoing waves by adding artificial material layers that are designed such that all of the incident waves impinging upon them are completely transmitted with minimal reflections. Thus PMLs can be placed closer than ABCs. Furthermore, the PML absorbs a much wider range of waves in terms of frequency and direction whereas ABC absorbs only normally incident waves

accurately. However, PMLs in general makes it more difficult for the iterative solver to reach convergence compared to ABCs. PMLs also preserves the sparse nature of the FEM formulation.

Note Radiating BCs and PMLs are local boundary conditions and inherit the Global Environment Material from the computational domain. They work as if the material of the computational domain were continued to infinity. FEBI uses the user defined Global Environment material which may or may not be the same as the material of the computational domain. If they are the same, PMLs and FEBI give the same results. If the materials are different, there is a jump in the materials which can be modeled by FEBI, but not by PMLs.

Boundary Integral Equations

The boundary integral equations (IE) is an exact transparent condition. The Sommerfeld radiation condition at infinity as required by physics is enforced exactly through the employment of appropriate Green's functions via an integral equation method. This hybridization is commonly known as the finite element-boundary integral (FEBI) method. Unlike ABC and PML, the IE boundary can be of arbitrary shape, both concave and convex thus in some cases allowing the size of the finite element solution domain to be significantly reduced. It can be placed very close to the structures and produce accurate results and from a performance point of view a minimum distance of 0.125 wavelengths is recommended. Since analytic Green's functions are known only for a few specific situations, the IE boundary is not allowed to touch any other boundary except an infinite ground plane. In addition the IE boundary does not work with symmetry planes. The IE boundary is not compatible with curvilinear finite element and thus should not be placed on a true surface if it is desired to use curvilinear elements in the finite element domain. The IE condition is a non-local condition, producing a partial sparse and partial dense system matrix. For this reason when simulating electrically large structures, the IE condition typically requires more RAM and CPU than either ABC or PML assuming all are placed on the same geometry. However by taking advantage of conformal radiation volumes to reduce the overall finite element solution domain utilization of the IE boundary will result in more efficient simulation for electrically large open boundary problems.

Related Topics

Assigning Radiation Boundaries

Technical Notes: PML Boundaries

PML Boundaries

Perfectly matched layers (PMLs) are fictitious materials that fully absorb the electromagnetic fields impinging upon them. These materials are complex anisotropic.

There are two types of PML applications: free space termination and guided wave termination. With free space termination, PMLs are associated with a surface that radiates into free space equally in every direction. PMLs are more appropriate than radiation boundaries in this case because PMLs enable radiation surfaces to be located closer to radiating objects, reducing the problem domain. Any homogeneous isotropic material, including lossy materials like ocean water, can surround the design.

With reflection-free termination of guided waves, the structure continues uniformly to infinity. Its termination surface radiates in the direction in which the wave is guided. Reflection-free PMLs are appropriate for simulating phased array antennas because the antenna radiates in a certain direction.

Related Topics

Assigning PML Boundaries

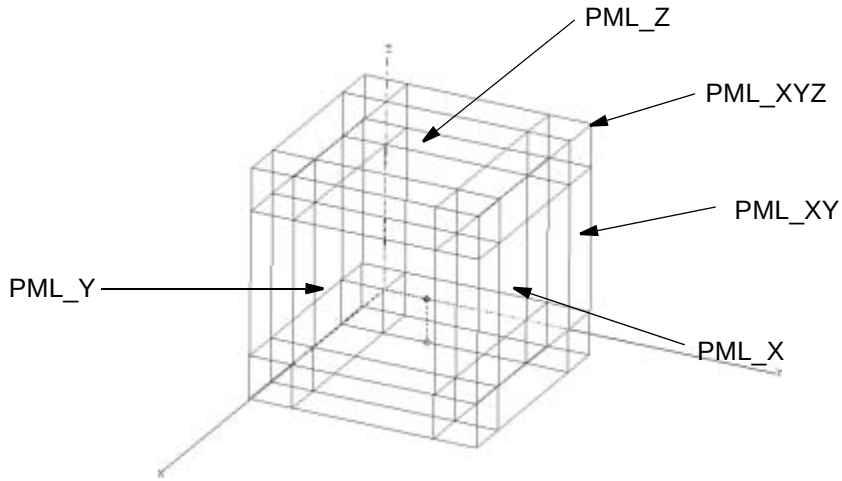
Material Tensors Applied at PML Boundaries

Tensor Entries

Boundaries at PML Surfaces

Material Tensors Applied at PML Boundaries

PMLs materials are complex anisotropic. An example is shown below.



To ensure that there will not be any reflection at the PML/air interface, the bi-axial diagonal material tensors for x-, y- and z-directed PMLs (PML_X, PML_Y, and PML_Z) are as follows.

For PML_X :

$$\frac{[\varepsilon]}{\varepsilon_0} = \begin{bmatrix} \frac{1}{C} & C & C \end{bmatrix} \qquad \frac{[\mu]}{\mu_0} = \begin{bmatrix} \frac{1}{C} & C & C \end{bmatrix} \tag{1}$$

For PML_Y:

$$\frac{[\varepsilon]}{\varepsilon_0} = \begin{bmatrix} C & \frac{1}{C} & C \end{bmatrix} \qquad \frac{[\mu]}{\mu_0} = \begin{bmatrix} C & \frac{1}{C} & C \end{bmatrix} \tag{2}$$

For PML_Z:

$$\frac{[\varepsilon]}{\varepsilon_0} = \begin{bmatrix} C & C & \frac{1}{C} \end{bmatrix} \qquad \frac{[\mu]}{\mu_0} = \begin{bmatrix} C & C & \frac{1}{C} \end{bmatrix} \tag{3}$$

where $C = a - jb$.

The tensors designated as PML_X characterize an x-directed PML corresponding to a PML wall in the yz plane. Similarly, PML_Y and PML_Z are designated tensors for y- and z-directed PMLs.

PMLs of different directions must be joined in order to construct a box with PML walls. To ensure complete coverage where the edges and corners of two PMLs meet, create edge and corner PML objects. The tensors of an edge object joining PML_X and PML_Y are as follows for PML_XY:

$$\frac{[\varepsilon]}{\varepsilon_0} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & C^2 \end{bmatrix} \qquad \frac{[\mu]}{\mu_0} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & C^2 \end{bmatrix} \tag{4}$$

A similar tensor construction rule is valid for joining x- and z-directed and y- and z-directed PMLs. The tensor for a corner object is a follows for PML_XYZ:

$$\frac{[\varepsilon]}{\varepsilon_0} = \begin{bmatrix} C & & \\ & C & \\ & & C \end{bmatrix} \qquad \frac{[\mu]}{\mu_0} = \begin{bmatrix} C & & \\ & C & \\ & & C \end{bmatrix} \tag{5}$$

Note This example in shows the material in GLOBAL coordinate system, while the Material Manager shows the materials in LOCAL coordinate system assigned to the object, which are then rotated to the global coordinate system by the solver.

Related Topics

Tensor Entries

Boundaries at PML Surfaces

PML Boundaries

Tensor Entries

Entering the matrices of the anisotropic materials doesn't require a special procedure. The usual anisotropic material definitions can be used for any PML structure. However, keep in mind that the efficiency of the PMLs depends on the material values assigned to them.

Setting the complex parameter C ensures that the electromagnetic field decays strongly in the PMLs. Back reflections from the bounding PECs are then kept below a prescribed bound. To accomplish this, the following inequalities have to be satisfied:

$$e \geq \frac{-\ln \rho}{2 D_{min} H} = e_{min} \tag{1}$$

$$e \leq \frac{-\ln d}{2 D_{max} h} = e_{max} \tag{2}$$

where

$$D_{min} = \alpha_{min} + \beta_{min} = \frac{1}{r_{max}} + \frac{\omega_{min}}{c} \quad (3)$$

$$D_{max} = \alpha_{max} + \beta_{max} = \frac{1}{r_{min}} + \frac{\omega_{max}}{c} \quad (4)$$

- $e = a = b$
- a and b are the real and imaginary parts of C.
- H is the thickness of the PML object.
- ω_{max} and ω_{min} are the minimum and maximum angular frequencies.
- r_{max} and r_{min} are the minimum and maximum distance of a radiating object to the PML surface.
- ρ is the bound for back reflection.
- d is the maximum decay characterizing the element. (d is approximately $3* 10^{-3}$.)
- h is the thickness of one finite element.
- c is the velocity of light in vacuum.

Related Topics

Material Tensors Applied at PML Boundaries

Boundaries at PML Surfaces

PML Boundaries

Boundaries at PML Surfaces

After embedding a structure in PMLs, the next step is to specify boundaries on the outer surface of the box. The simplest way is to bound the box either with perfect electric conductors (PECs) or perfect magnetic conductors (PMCs.) In general, use PECs because they reduce the problem size.

Related Topics

Tensor Entries

PML Boundaries

Boundaries at PML Surfaces

Material Tensors Applied at PML Boundaries

Finite Conductivity Boundaries

In HFSS, finite conductivity boundaries represent imperfect conductors. At such boundaries, the following condition holds:

$$\boldsymbol{E}_{tan} = Z(\hat{n} \times \boldsymbol{H}_{tan}) \quad (1)$$

where

- E_{tan} is the component of the E-field that is tangential to the surface.
- H_{tan} is the component of the H-field that is tangential to the surface.
- Z_s is the surface impedance of the boundary. When the thickness is much larger than the skin depth $(1 + j)/(\delta\sigma)$ can be used. When not, Z_s can be calculated from the transmission line model used at a layer impedance boundary condition.
  - δ is the skin depth, $\sqrt{2/(\omega\sigma\mu)}$, of the conductor being modeled.
  - ω is the frequency of the excitation wave.
  - σ is the conductivity of the conductor.
  - μ is the permeability of the conductor.

The fact that the E-field has a tangential component at the surface of imperfect conductors simulates the case in which the surface is lossy.

The surfaces of any objects defined to be non-perfect conductors are automatically set to finite conductivity boundaries. Note that HFSS does not attempt to compute the field inside these objects; the finite conductivity boundary approximates the behavior of the field at the surfaces of the objects.

The finite conductivity boundary condition is valid only if the conductor being modeled is a good conductor, that is, if the conductor's thickness is much larger than the skin depth in the given frequency range. If the conductor's thickness is in the range or larger than the skin depth in the given frequency range, HFSS takes the thickness into account, if it has been defined.

Related Topics

Surface Roughness Model: Huray

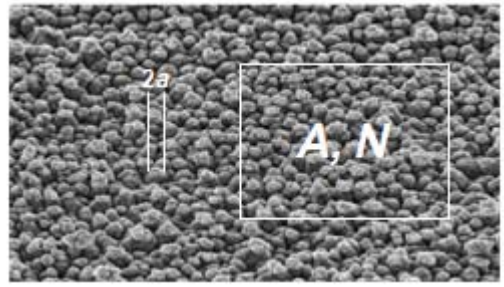
Surface Roughness Model: Huray

Because surface roughness can increase conductor power losses more than two times, HFSS includes an advanced surface roughness model based on the visible features of copper conductors used in circuit fabrication. For instance, the magnified surface of copper appears as an irregular collection of nodules. The Huray model of the physical effects calls for three parameters:

A_f, the area of a unit cell

N, the number of nodules per cell (modeled as spheres).

a, the radius of a typical nodule.



These relate to the two user specified parameters in the Finite Conductivity Boundary dialog as follows:

Nodule radius, a

Hall-Huray Surface Ratio = 4 * PI *a^2*N/A_f

The equations

$$Z_{rough} = Z_{smooth}\left(\frac{P_{rough}}{P_{smooth}}\right) = Z_{smooth}factor \quad (1)$$

$$factor = \frac{P_{rough}}{P_{smooth}} = 1 + \frac{3}{2}\frac{N4\pi a^2}{A_f}\frac{1}{1 + \frac{\delta}{a} + \frac{\delta^2}{2a^2}} \quad (2)$$

where

$$\delta = \sqrt{\frac{2}{\omega\mu\sigma}} \quad (3)$$

Introducing

$$sr = \frac{N4\pi a^2}{A_f} \quad (4)$$

The actual calculation implemented is:

$$factor = \frac{P_{rough}}{P_{smooth}} = 1 + \frac{3}{2}sr\frac{1}{1 + \frac{\delta}{a} + \frac{\delta^2}{2a^2}} \quad (5)$$

P_{rough}	power loss at the rough surface
P_{smooth}	power loss at the smooth surface
a	radius of the uniform spheres
N	number of uniform spheres
A_f	surface of the hexagonal geometry
delta	skin depth
f	frequency [1/s]

The dialog accepts two parameters: a and sr. The simplified model uses a=0.5 um, sr= 2.9

References:

1. P.G. Huray, S.G. Pytel, S.H. Hall, F. Oluwafemi, R.I. Mellitz, D. Hua, and P. Ye, "Fundamentals of a 3-D "Snowball" Model for Surface Roughness Power Losses", 11th Annual IEEE SPI Proceedings, May 13 - 16, 2007.
2. S.H. Hall, S.G. Pytel, P.G. Huray, D. Hua, A. Moonshiram, G. Brist, and E. Sijercic, "Multi-

GHz, Causal Transmission Line Modeling Methodology with a Hemispherical Surface Roughness Approach", IEEE Transactions on Microwave Theory and Techniques, December 2007 pp 2614 - 2624.

3. S.G. Pytel, P.G. Huray, A. Moonshiram, S.H. Hall, R.I. Mellitz, G. Brist, F. Oluwafemi, H.M. Meyer, L. Walker, and M. Garland, "Analysis of Copper Treatments and the Effects on Signal Propagation", 58th Annual IEEE ECTC, May 26 - 30, 2008, pp 1144 - 1149.
4. S.G. Pytel, "Multi-gigabit data signaling rates for PWBs including dielectric losses and effects of surface roughness", PhD. Dissertation, University of South Carolina, 2007.
5. P.G. Huray, O. Oluwafemi, J. Loyer, E. Bogatin, and X. Ye; "Impact of Copper Surface Texture on Loss: A Model That Works", DesignCon 2010, February 1 - 4, 2010.

Symmetry Boundaries

In HFSS, symmetry boundaries represent perfect E or perfect H planes of symmetry. Symmetry boundaries enable you to model only part of a structure, which reduces the size or complexity of your design, thereby shortening the solution time.

When you are defining a symmetry plane, keep the following requirements in mind:

- A plane of symmetry must be exposed to the background.
- A plane of symmetry must not cut through an object drawn in the **3D Modeler** window.
- A plane of symmetry must be defined on a planar surface.
- Only three orthogonal symmetry planes can be defined in a problem.

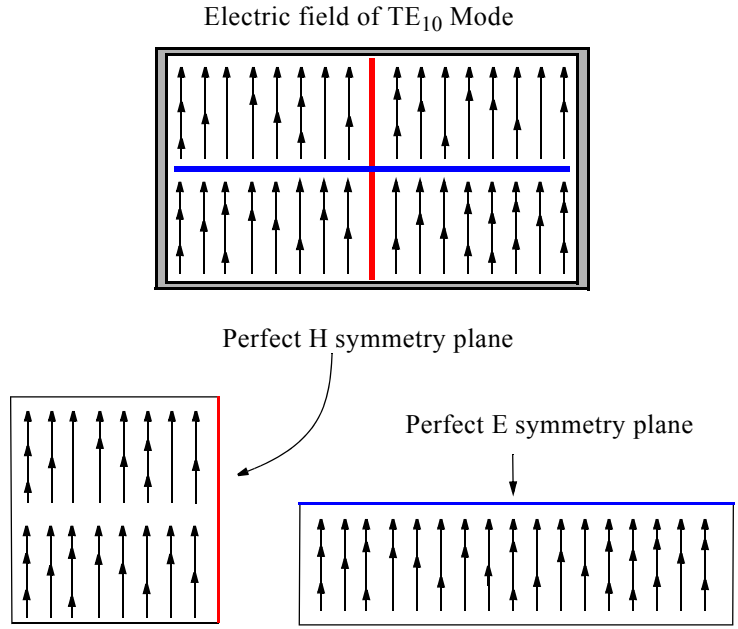
Perfect E Vs. Perfect H Symmetry Boundaries

In general, use the following guidelines to decide which type of symmetry boundary to use, a perfect E or a perfect H:

- If the symmetry is such that the E-field is normal to the symmetry plane, use a perfect E symmetry plane.
- If the symmetry is such that the E-field is tangential to the symmetry plane, use a perfect H symmetry plane.

The simple rectangular waveguide shown below illustrates the differences between the two types of boundaries. The E-field of the dominant mode signal (TE_{10}) is shown. The waveguide has two planes of symmetry, one vertically through the center and one horizontally.

The horizontal plane of symmetry is a perfect E surface. The E-field is normal and the H-field is tangential to that surface. The vertical plane of symmetry is a perfect H surface. The E-field is tangential and H-field is normal to that surface.



For common problems, you can usually decide which symmetry boundary to use by reviewing the geometry. For example, if the structure is a microstrip, the flux lines of the E-field run between the ground plane and the conductive strip; therefore, the E-field is tangential to any vertical symmetry plane that slices a microstrip in half.

Symmetry and Port Impedance

If a symmetry plane has been defined, the computed port impedances will not match the port impedance of the full structure unless an impedance multiplier is specified.

Note Port impedance is only calculated when a port has been defined. If you are solving a problem without ports, you do not need to specify an impedance multiplier.

Symmetry and Multiple Modes

If you are solving for multiple modes, keep in mind that the orientation of the E- and H-fields may differ from mode to mode. A perfect H symmetry boundary for the dominant mode may be a perfect E symmetry for another mode.

Master and Slave Boundaries

Master and slave boundaries enable you to model planes of periodicity where the E-field on one surface matches the E-field on another to within a phase difference. They force the E-field at each

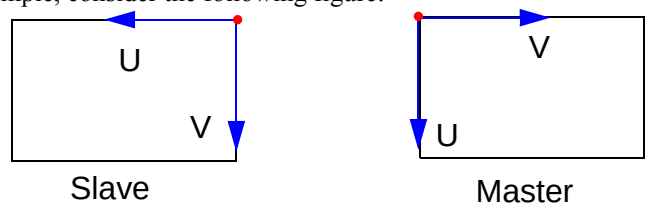
point on the slave boundary match the E-field to within a phase difference at each corresponding point on the master boundary. They are useful for simulating devices such as infinite arrays.

Unlike symmetry boundaries, **E** does not have to be tangential or normal to these boundaries. The only condition is that the fields on the two boundaries must have the same magnitude and direction (or the same magnitude and opposite directions).

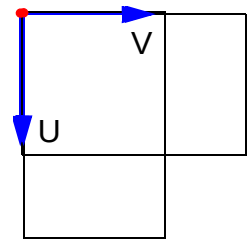
When creating matching boundaries, keep the following points in mind:

- Master and slave boundaries can only be assigned to planar surfaces. These may be the faces of 2D or 3D objects.
- The geometry of the surface on one boundary must match the geometry on the surface of the other boundary. For example, if the master is a rectangular surface, the slave must be a rectangular surface of the same size.
- If the mesh on the master boundary does not match the mesh on the slave boundary exactly, the solution will fail. Normally HFSS automatically forces the mesh to match on each boundary; however, in some cases, the mesh cannot be forced to match. To prevent the solution from failing, create a virtual object on the slave boundary that exactly matches any extra object on the master boundary, or create a virtual object on the master boundary that exactly matches any extra object on the slave boundary.
- To make a surface a master or slave boundary, you must specify a coordinate system that defines the plane on which the selected surface exists. When HFSS attempts to match the two boundaries, the two coordinate systems must also match each other. If they do not, HFSS will transpose the slave boundary to match the master boundary. When doing this, the surface to which the slave boundary is assigned is also transposed. If, after doing this, the two surfaces do not occupy the same position relative to their combined defined coordinate system, an error message appears.

  For example, consider the following figure:



  To match the coordinate system of the master boundary, the coordinate system on the slave boundary must rotate 90 degrees counterclockwise; however, when this is done, you get the following:



The two surfaces do not correspond and thus the mesh will not match, causing an error message.

- The angle between the axes defined by the u point and v point must be identical for the master and slave boundary.

Calculating the E-Field on the Slave Boundary

The E-field on the slave boundary is forced to match the E-field on the master boundary. The magnitude of the E-field on both boundaries is the same; however, the fields may be out of phase with each other.

The function relating the electric field on the slave boundary, E_S, to the electric field on the master boundary, E_M, depends on the type of problem you are solving. For example, consider an infinite array simulation for a rectangular array. If the array excited to radiate in the direction (θ, ϕ) in spherical coordinates. The fields above the array experience a phase delay of

$$\Psi = k(\hat{r}_o \bullet \boldsymbol{v}) \quad (1)$$

where

- $\hat{r}_0$ is the unit vector in the direction of scan.
- $\boldsymbol{v}$ is the vector from the slave boundary to the master boundary.

To simulate this in the finite element solution, HFSS incorporates phase shifts in the relation between the matching boundaries. That is, the electric field values on the master boundary will be related to the electric field values on the corresponding points on the slave boundary. This equation would be the following:

$$\boldsymbol{E}_S = e^{j\Psi}\boldsymbol{E}_M \quad (2)$$

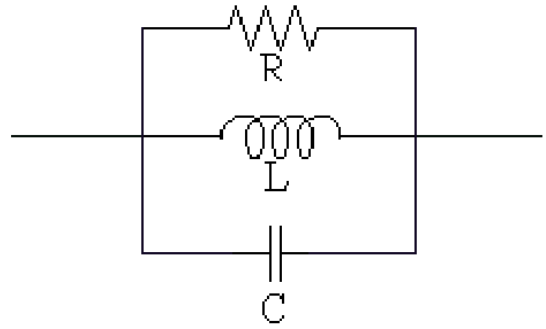
HFSS gives you the option of entering the scan angles, ϕ and θ, when relating E_S to E_M. The phase delay is calculated from the scan angles. However, if you know the phase delay, you may enter that directly.

Anisotropic Impedance Boundaries

Planar screens or grids of large extent with periodic geometry can be replaced by a Anisotropic impedance boundary. The boundary applies a homogeneous characteristic impedance to the surface in an effort to create an equivalent electrical representation of the geometric grid pattern.

Lumped RLC Boundaries

To model any combination of lumped resistor, inductor, and/or capacitor in parallel on a surface, create a lumped RLC boundary. A lumped RLC boundary represents R, L, and C in parallel:



Similar to impedance boundaries, the following condition holds at lumped RLC boundaries:

$$\boldsymbol{E}_{tan} = Z_s(\hat{n} \times \boldsymbol{H}_{tan}) \quad (1)$$

where

- $\hat{n}$ is the is the unit vector that is normal to the surface.
- E_{tan} is the component of the E-field that is tangential to the surface.
- H_{tan} is the component of the H-field that is tangential to the surface.
- Z_s is the surface impedance of the boundary, $Rs + jXs$, where
  - Rs is the resistance in ohms/square.
  - Xs is the reactance in ohms/square.

Unlike impedance boundaries, you are not required to supply the impedance per square, but you must supply the actual values for R, L, and C. HFSS then determines the impedance per square of the lumped RLC boundary at any frequency.

A Fast frequency sweep is supported for this boundary condition.

Layered Impedance Boundaries

A layered impedance boundary is used to model multiple layers in a structure as one impedance surface. The effect is the same as an impedance boundary condition, except that HFSS calculates the reactance and resistance values for the surface based on data you enter for the layered structure. Surface roughness is also taken into account.

The reactance and resistance values are calculated differently for internal and external layered impedance boundaries. For external layered impedance boundaries, HFSS calculates the impedance for the side of the surface in contact with the computational domain and assigns this value to the boundary. For internal layered impedance boundaries, HFSS calculates the average impedance value for the two sides of the surface in contact with the computational domain and assigns this value to the boundary.

The layered impedance boundary is supported for single-frequency solutions and Discrete and Interpolating frequency sweeps.

Related Topics

Impedance Calculation for Layered Impedance Boundary

Surface Roughness Calculation for Impedance Boundary

Impedance Calculation for Layered Impedance Boundary

The impedance of the layered structure is calculated by recursively calling the impedance calculation formulation known from transmission line theory:

$$Z_{inputk} = Z_{wk}\frac{Z_{inputk+1}ch(\gamma_k d_k) + Z_{wk}sh(\gamma_k d_k)}{Z_{inputk+1}sh(\gamma_k d_k) + Z_{wk}ch(\gamma_k d_k)} \quad (1)$$

where

- Z_{inputk} is the input impedance for the k[th] layer.

$$Z_{wk} = \sqrt{\frac{\mu_0\mu_{rk}}{\varepsilon_0\mu\varepsilon_{rk}}} \quad (2)$$

- *ch* is the hyperbolic cosine function.
- *sh* is the hyperbolic sine function.
- $\gamma_k = k_0\sqrt{-\varepsilon_{rk}\mu_{rk}}$

  where
  - γ is the propagation coefficient.
  - k_0 is the free space wave number, $\omega\sqrt{\mu_0\varepsilon_0}$, where ω is the angular frequency, $2\pi f$.
  - ε_{rk} is the relative complex permittivity of the K[th] layer.
  - μ_{rk} is the relative complex permeability of the K[th] layer.

  where
  - $\varepsilon_{rk} = epsr_k - j\left(\frac{sigma}{\omega\varepsilon_0} + epsr \cdot \tan de\right)$
  - $\mu_{rk} = mur_k - j(mur_k \cdot \tan dm_k)$
  - d_k is the thickness of the K[th] layer.

Surface Roughness Calculation for Impedance Boundary

The surface roughness is measured as the RMS deviation of the conductor surface from a plane. Surface roughness increases conduction losses. HFSS calculates surface roughness by modifying the conductivity as follows:

$$\sigma_c = \frac{\sigma}{K_w^2} \quad (1)$$

where

σ is the material's conductivity.

$$K_w = 1 + \exp\left(-\left(\frac{s}{2h}\right)^{1.6}\right) \quad (2)$$

where, further:

- h is the surface roughness.
- s is the skin depth.

Infinite Ground Planes

To simulate the effects of an infinite ground plane, select the **Infinite ground plane** check box when setting up a perfect E, finite conductivity, or impedance boundary condition. The selection only affects the calculation of near- and far-field radiation during post processing. The 3D Post Processor models the boundary as a finite portion of an infinite plane.

Conceptually, a boundary condition designated as an infinite ground plane divides the problem region into the half above it, where the entire model resides, and the half below it, where the radiated fields are set to zero. Antenna parameters involving radiated power will be consistent with these properties.

Lossy ground planes may be approximated by selecting the **Infinite ground plane** check box when assigning a finite conductivity or impedance boundary. The effects of these boundaries are incorporated into the field solution in the usual manner, but the radiated fields in the 3D Post Processor are computed as if the lossy ground planes were perfectly conducting.

When defining an infinite ground plane, keep the following requirements in mind:

- An infinite ground plane in a model must be exposed to the background.
- An infinite ground plane must be defined on a planar surface.
- The total number of infinite ground planes and symmetry planes cannot exceed three.
- All infinite ground planes and symmetry planes must be mutually orthogonal.
- For impedance, layered impedance, or finite conductivity boundaries only one infinite ground plane can exist in a design. For perfect E boundary conditions, multiple antenna ground planes are supported

Frequency-Dependent Boundaries and Excitations

In general, boundary and excitation parameters cannot depend on intrinsic functions. An exception is when a parameter depends on the variable *Freq*, which represents the solution frequency. The following boundary parameters can be assigned an expression that includes *Freq*:

- Impedance boundary - the Resistance and Reactance parameters.
- Finite conductivity boundary - the Conductivity parameter. If a material is specified, the material can be frequency dependent.
- Slave boundary - the Phase parameter.
- Lumped RLC boundary - Resistance, Inductance, and Capacitance parameters.
- Layered impedance boundary - materials assigned on layers can be frequency dependent.
- Anisotropic Impedance boundary - Resistance and Reactance parameters.,

Note Dependence on *Freq* is supported for single-frequency solutions and for Discrete and Interpolating frequency sweeps. If a Fast sweep is requested, the solution will be valid for the center frequency, but may not be valid at other frequencies.

Implicit Boundary Assignments

If a boundary has not been assigned to a model surface, one of the following default boundaries will be assigned to the surface:

smetal	A single perfect E boundary is assigned to all objects that do not have **Solve Inside** selected in the **Properties** window and that are perfect conductors.
i_<*object name*>	A finite conductivity boundary is assigned to each object that does not have **Solve Inside** selected in the **Properties** window and that is not a perfect conductor. <*object name*> is the name of the object on which the boundary is assigned.
outer	A default boundary applied on the outermost surfaces of the model.

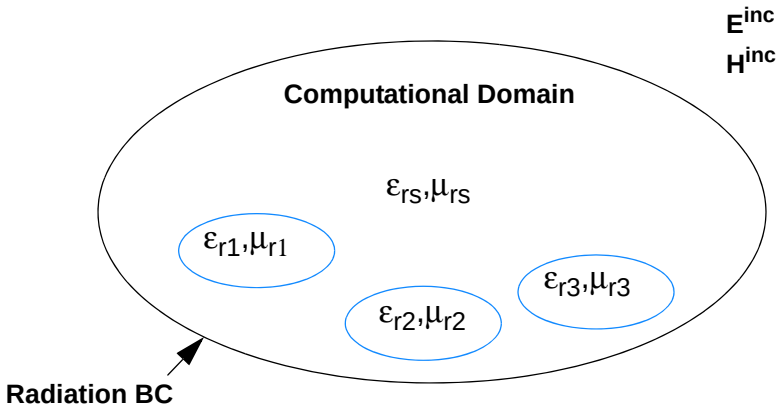
These boundary assignments are intrinsic and not displayed in the modeler.

Related Topics

Reviewing Boundaries and Excitations in the Solver View

General Approach for Scattered and Total Field Decomposition

Given our compositional domain:



- Incident field is known in absence of our computational domain
- Physical (total) field is sought in presence of our computational domain
- Radiation BC applies to the field generated by the objects of our computational domain (scattered field)
- Incident field might be known in the whole domain (scattered field description) or just on the surface ("space ports," total field description)
- HFSS Transient provides the total field. The fair comparison is to use a single frequency in the frequency domain and apply a harmonic excitation in the time domain with the same frequency.

Related Topics

Scattered and total field formulation pros and cons

Where the Solvers Apply Scattered or Total Field Formulations

Summary for Total and Scattered Fields

Scattered and total field formulation pros and cons

Scattered field formulation

- Mostly for plane waves and analytical waves
- Cancellation errors when total field is weak
- Problems arise at infinite scatterers (slot in an infinite conducting plate)
- Near/far incident waves are not known in the whole region

Total field formulation

- Applicable for any incident pattern

- Cancellation errors when scattered field is weak
- Infinite scatterers can be treated
- Applicable to near/far field, too, since excitations have to be known on surfaces
- New, space port concept (decomposition differs from the literature).

Related Topics

General Approach for Scattered and Total Field Decomposition

Scattered and total field formulation pros and cons

Where the Solvers Apply Scattered or Total Field Formulations

Summary for Total and Scattered Fields

Where the Solvers Apply Scattered or Total Field Formulations

Scattered Field Formulation

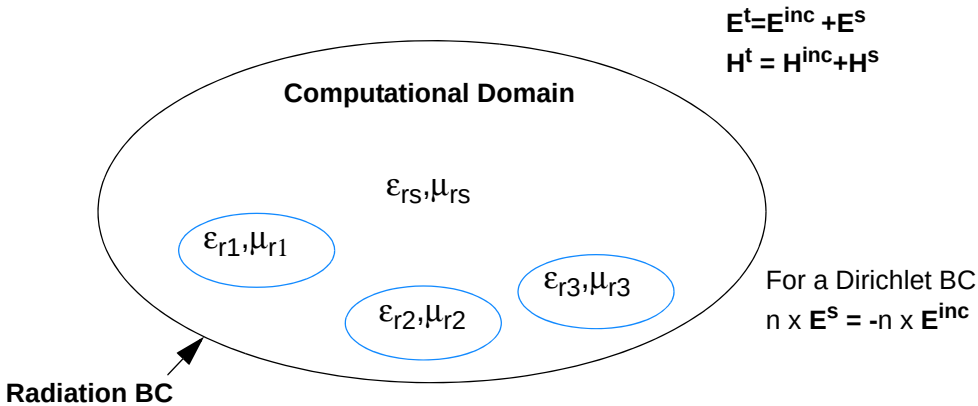
- Monostatic/bistatic RCS of finite scatterers
- Frequency Selective Surfaces (FSSs)

Total Field Formulation

- Monostatic/bistatic RCS of infinite scatterers
- Frequency Selective Surfaces (FSSs)
- When near/far field Link is involved
- EMC/EMI problems

For Scattered Field Decomposition

It is known in the whole domain.



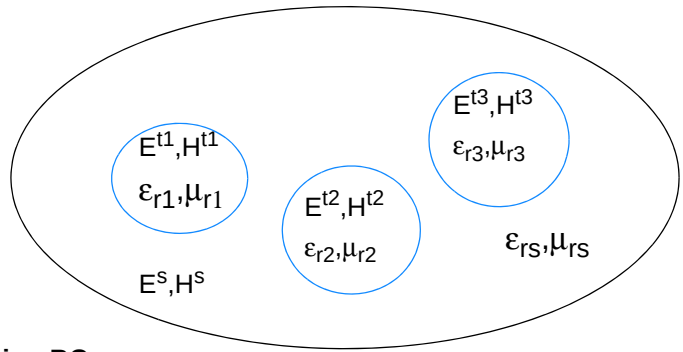
In summary, scattered field descriptions apply:

- Environment material other than vacuum
- Inhomogeneous magnetic material

- Special treatments on
  - Ports
  - Impedance BCs
  - PMC

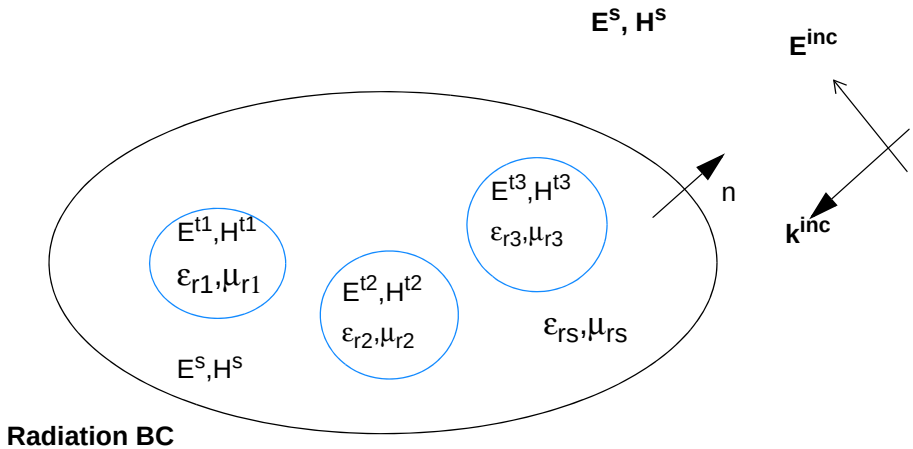
Total Field Description: Usual Approach

When the incident field is known, everywhere, the solvers take the usual approach. Both E and H jump with $n \times E^{inc}$ and $n \times H^{inc}$ on the interfaces, respectively.



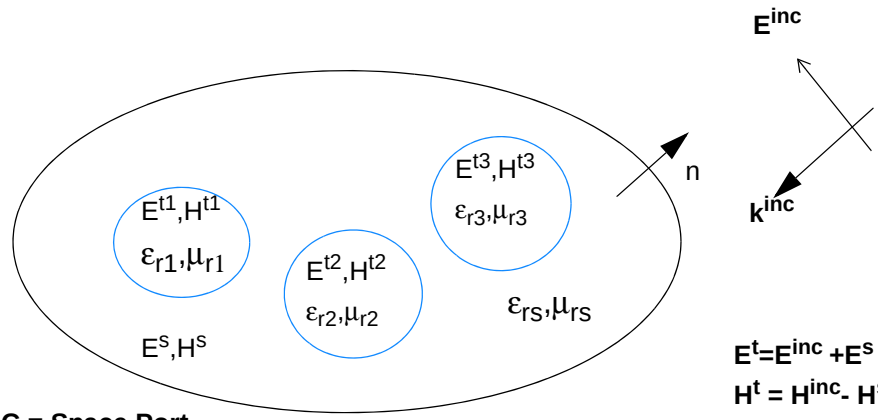
Total Field Description: Unusual Approach

In this case, the incident wave is just on known on the surface. Both E and H jump with $n \times E^{inc}$ and $n \times H^{inc}$ on the outer surface respectively.



Total Field Description: Space Port Concept

The Radiation Boundary Condition is treated as a space port. The field is decomposed just on the surface of the space port.



Radiation BC = Space Port

Related Topics

General Approach for Scattered and Total Field Decomposition

Scattered and total field formulation pros and cons

Where the Solvers Apply Scattered or Total Field Formulations

Summary for Total and Scattered Fields

Summary for Total and Scattered Fields

- Enhanced scattered field formulation
- Scattered field formulation for classical RCS problems
- Total field formulation with space ports is necessary for numerical incident waves (near/far)
- Total field formulation is necessary to analyze infinite scatterers (infinite plane, iris, etc)
- Total field formulation improves the accuracy in many cases

Related Topics

General Approach for Scattered and Total Field Decomposition

Scattered and total field formulation pros and cons

Where the Solvers Apply Scattered or Total Field Formulations

Excitations

Assigning excitations to an HFSS design enables you to specify the sources of electromagnetic fields and charges, currents, or voltages on objects or surfaces. This area of the Technical Notes includes information on the following topics:

- Wave Ports Theory
- Lumped Ports Theory
- Terminals Theory
- Floquet Ports Theory
- Incident Waves
- Voltage Sources
- Current Sources
- Magnetic Bias Sources
- Dembedding

Wave Ports Theory

By default, the interface between all 3D objects and the background is a perfect E boundary through which no energy may enter or exit. Wave ports are typically placed on this interface to provide a window that couples the model device to the external world.

HFSS assumes that each wave port you define is connected to a semi-infinitely long waveguide that has the same cross-section and material properties as the port. When solving for the S-parameters, HFSS assumes that the structure is excited by the natural field patterns (modes) associated with these cross-sections. The 2D field solutions generated for each wave port serve as boundary conditions at those ports for the 3D problem. In addition to serving as a boundary condition, a wave port also supplies port impedances and propagation constants that are useful in describing waveguides or transmission lines.

A wave port is restricted to reside at an external boundary of a 3D problem. In some instances, it makes sense to get around this restriction by defining a wave port in the interior of a 3D domain by capping the wave port surface with a PEC object. In doing this, in effect the wave port is viewed as residing at the external boundary of a 3D problem. However, in general lumped ports should be used when defining ports in the interior of a 3D domain.

Often there is a need to have a particular field direction of a mode pattern for consistent results. There are two different methods in HFSS to accomplish this task. The most basic method is where, in the presence of an integration line, HFSS resolves 180 degree ambiguity by enforcing the electric field mode pattern to have a positive line integral along the integration line for that mode (Mode Polarity). The second more advanced option is where HFSS aligns electric field mode patterns, when appropriate, in specific directions (Mode Alignment). This second option consists of a manual but general approach and an automated approach for a few port types. For more in-depth discussions refer to the following topics:

- Mode Polarity

- Mode Alignment

Related Topics

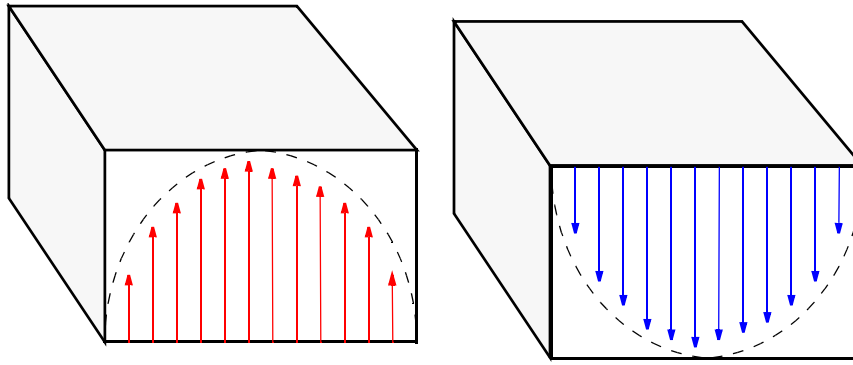
Assigning Wave Ports for Modal Solutions

Mode Polarity

When HFSS computes an excitation mode pattern, the direction of the field at $\omega t = 0$ is arbitrary; the field can point in one of at least two directions. This document addresses how to remove the ambiguity of the field direction when a mode is not part of a set of degenerate modes. For the case of degenerate modes, there is no unique mode alignment and this topic is addressed in Mode Alignment.

In the figure below, at $\omega t = 0$ the electric field pattern of the mode can point either up (left image) or down (right image). Either direction is correct - unless a preferred direction is specified. This issue becomes of concern when inspecting the phase angle of S-parameters.

For example, consider a 3D model containing two single-mode ports of the type shown in the figure. The phase of the computed S-parameters linking the two ports will flip by 180 degrees if the polarity of the mode at either port is reversed. Therefore when transmission S-parameter phase is of interest in models such as this, the user has the responsibility to properly set the mode polarity at each port.



To specify the polarity of a mode, a mode calibration must be performed by creating an appropriate integration line. For instance, if the preferred direction is up then the integration line should be drawn at the center of the waveguide starting at the bottom and ending at the top. In general, a mode calibration is automatically performed if an integration line has been defined for a mode where HFSS ensures that the real part of the line integral of the electric field of that mode is positive.

Mode Alignment

In some cases the requested set of modes to be used with a wave port contains degenerate modes. For that case, the modes that are degenerate need special care in obtaining a unique orientation of the mode patterns which is important, for example, when a specific orientation for a given mode is intended to excite the structure.

For example, in the case of a square waveguide the first two dominant modes form a degenerate mode set. Therefore, in order to guarantee that the typically desired orientation of the modes (TE_{01}

and TE_{10}) are used for an HFSS simulation, the user needs to enable one of the two supported mode alignment options. The first option is general but manual and requires detailed user knowledge of the mode patterns while the other is restricted to a few basic port types where HFSS automatically aligns the modes.

The general mode alignment option is enabled by defining appropriate integration lines for the modes that should be aligned and selecting **Align modes using integration lines** in the **Modes** tab of the Wave Port setup window. Note that not all modes requested for a wave port need to have an integration line defined - only the ones for which mode alignment is desired. It is important that you draw integration lines that make sense for the modes to be aligned along and that all the modes that form a set of degenerate modes have a line defined. For example, if the first two modes of a square waveguide should be TE_{10} and TE_{01}, you should draw two lines through the center of the waveguide where one line starts at the bottom and ends at the top while the other line starts on the left side and ends on the right side.

You enable the automated mode alignment option by selecting **Align modes analytically using coordinate system** in the **Modes** tab of the Wave Port setup window. By selecting this option, the user will be asked to draw a local coordinate system which is used to automatically align the modes in pre-defined directions. This option is only available for a select few port types discussed in Analytic Port Types.

In order to get intended outcome of either alignment option, it is necessary to include all modes in a set of degenerate modes. For example, assume that modes 3, 4, 5, and 6 are degenerate. You should either include all of them by specifying at least 6 modes or none of them by specifying at most 2 modes.

Analytic Port Types

Analytic mode alignment is available for certain port types by selecting **Align modes analytically using coordinate system** in the **Modes** tab in the Wave Port setup window. If analytic mode alignment is selected for a port that is not supported, the port setup becomes invalid. Here is a summary of the validation for analytic mode alignment:

- The port geometry must correspond to a known analytic type, as follows:
  - Rectangular. The port must have four linear edges which meet at right angles. Square ports are a special case of rectangular.
  - Circular. The port must have all curved edges. The radius, perimeter, and area of the face must be consistent with those of an ideal circle.
  - Faceted Circular. The port must have at least 16 edges, all linear and of equal length. The radius, perimeter, and area of the face must be consistent with those of an ideal polygon.
  - Circular or Faceted Coax.

    Explicit. The port face must have one set of inner and one set of outer edges. The inner radius, outer radius, face area and perimeter must be consistent with an ideal coaxial circle or polygon.

    Implicit. The port face itself must be circular or faceted circular, and it must have a circular or faceted circular conducting object touching it and aligned with the face center.

- The port must be assigned to a single face of a 3D object whose material is isotropic. Assignment to sheets is not allowed, as this makes it difficult to determine the object material.
- No other material can contact the port, except for the inner conductor of a coax.
- Only Perfect E and Finite Conductivity boundaries can contact the port.

When reading legacy projects prior to v12, HFSS automatically tests ports with the "Polarize E Field" option enabled. If they qualify for analytic alignment, that option will be selected, and U-V lines will be set from the first two modal integration lines.

Related Topics

Assigning Wave Ports for Modal Solutions

Technical Notes: Wave Ports

Technical Notes: Mode Alignment

Technical Notes: Alignment Groups for Waveports with Multiple Modes

Alignment Groups for Waveports with Multiple Modes

When assigning wave ports for modal solutions, alignment groups can be used only for a consecutive block of degenerate modes with each block of modes identified by a unique group number. There may be modes at the end of the modes table which have no assigned alignment group, but no unassigned modes in the middle of a block of degenerate modes containing an alignment group definition. Finally, each mode with an alignment group must have an integration line.

Alignment groups allow the solver to collect the modes within a group and ensure that each mode has non-zero voltage on its own integration line and zero voltage on every other integration line in the group. If the modes in a group are not degenerate or its defined integration lines are not appropriately defined, hf3d will terminate with an error message.

Any change in the mode groups will cause invalidation of solutions but not of the mesh.

Related Topics

Mode Alignment

Lumped Ports Theory

Lumped ports resemble wave ports, but can be located internally and have a complex user-defined reference impedance. Lumped ports are restricted to single mode ports and the S-parameters are always based on the user defined reference impedance. This mainly because no transmission line is being modeled in the interior domain which suggests an interpretation of the lumped port as a measurement probe being connected to the surface of the lumped port with the reference impedance specified by the user.

When a lumped port is used internal to a 3D problem, it makes sense to place the lumped port at a location where the field distribution would approximately be the same as the dominant mode of the port definition in the absence of the lumped port. A typical example of this is when a rectangular port is drawn between a conductor trace and a ground plane with the face oriented perpendicular to both conductors. The port mode resulting from this case is a parallel plate mode which will often closely resemble the field distribution at that location in the absence of the port.

A perhaps unsuspected result of using a complex reference impedance is that the S-parameters can be greater than one even for a passive device.

Note When a lumped port is used as an internal port, the conducting cap required for a traditional wave port must be removed to prevent short-circuiting the source.

Terminals Theory

HFSS can categorize microwave structures in terms of a black-box that relates voltages and currents flowing in and out of a given structure. The black-box has several terminals, each with an associated voltage/current pair. In HFSS, these terminals reside inside wave ports that enable post processing of a modal representation of the black-box into the terminal representation.

When a terminal project is solved using HFSS, the number of modes for a port is determined by the number of terminals touching the port. If N+1 distinct conductors touch the port, there are N terminals and one reference conductor usually referred to as ground. The modal port representations of the electric and magnetic fields are:

$$\mathrm{E} = \sum_{n=1}^{N} (a_n + b_n)\mathrm{e}_n \quad (1)$$

$$\mathrm{H} = \sum_{n=1}^{N} (a_n - b_n)\mathrm{h}_n \quad (2)$$

where a_n and b_n are unitless complex amplitudes of the ingoing and outgoing modal fields, respectively. The modal black-box representation of a given structure is given by HFSS in terms of the generalized scattering matrix, *S*,

$$\mathrm{b} = S\mathrm{a} \quad (3)$$

where a and b are the unitless complex modal coefficient vectors. In order to obtain a terminal representation of the black-box, N integration paths Ci and N integration loops Li are used to define N voltages and N currents, respectively,

$$v_i = \int_{C_i} (\vec{\mathbf{E}} \cdot \vec{\mathbf{dl}}) = \sum_{n=1}^{N} (a_n + b_n) \int_{C_i} \vec{\mathbf{e}}_n \cdot \vec{\mathbf{dl}} = \sum_{n=1}^{N} (a_n + b_n) V_{in} \quad i = 1,2\ldots,N \quad (4)$$

$$i_i = \oint_{L_i} (\mathbf{H} \cdot \mathrm{dl}) = \sum_{n=1}^{N} (a_n - b_n) \oint_{L_i} \mathrm{h}_n \cdot \mathrm{dl} = \sum_{n=1}^{N} (a_n - b_n) I_{in} \quad i = 1,2\ldots,N \quad (5)$$

In matrix notation, the equations become

$$\mathrm{v} = T(\mathrm{a} + \mathrm{b}) \quad (6)$$

$$i = U(a - b) \quad (7)$$

Combining these equations with (3), yields the following circuit or terminal description of the black-box representation of the structure

$$i = Yv \quad (8)$$

where the terminal admittance matrix Y is given by

$$Y = U(I - S)(I + S)^{-1} T^{-1} \quad (9)$$

In HFSS, a generalized terminal scattering matrix S_t is supplied to the user..

$$\beta = s_t \alpha \quad (10)$$

where α and β are ingoing and outgoing pseudo waves, respectively. The S_t matrix is obtained by assuming that the voltages and currents are measured by some device with a given renormalizing impedance. This assumption leads to the following relations.

$$v = Z_{ref}^{1/2} \quad (\alpha + \beta) \quad (11)$$

$$i = Z_{ref}^{-1/2} \quad (\alpha - \beta) \quad (12)$$

where Zref is a renormalizing impedance matrix representing the measuring device. In HFSS, the user has the option to use the terminal characteristic impedance matrix, *Zo*, as the renormalizing impedance matrix by using the 'Do Not Renormalize' option in the port post processing tab. This means that the measuring device is perfectly matched to the transmission line. By default, all the terminals are individually renormalized which results in Zref being a diagonal renormalizing impedance matrix. An expression for St is obtained by using equations (8) - (12)

$$S_t = (I + Z_{ref}^{1/2} \quad YZ_{ref}^{1/2} \quad)^{-1} (I - Z_{ref}^{1/2} \quad YZ_{ref}^{1/2} \quad) \quad (13)$$

Terminals in HFSS can also be used to create differential pairs as discussed in the Differential Pairs section.

Related Topics

Assigning Lumped Ports for Terminal Solutions

Assigning Wave Ports for Terminal Solutions

Differential Pairs

A differential pair signal resides on two conducting traces in the presence of a ground plane. To form the differential pair, the transmitter launches a wave that carries equal but opposite voltages on the two traces. For example, one trace is +1 volt and the other -1 volt. When two traces are routed in close proximity to each other, the waves traveling on those traces will exhibit similar amounts of system noise. A differential pair is an effective way to remove noise from a signal because when a receiver processes a differential signal it simply subtracts the voltages from the

two traces. This can dramatically reduce the amount of noise in the differential signal compared to if the traces were driven individually (single-ended signals).

In HFSS, a differential pair can be defined for two terminals residing in the same wave port or for terminals that reside in two different wave ports or lumped ports. In the case of wave ports, matched differential pairs can be defined by not renormalizing the wave ports involved in the differential pair definitions.

Related Topics

Technical Notes: Computing Differential Pairs

Computing Differential Pairs

To obtain various differential pair quantities, we first define differential and common voltages v_d and v_c in terms of the singled-ended terminal voltage pairs v_1 and v_2, (see Setting up Differential Pairs).

The differential and common voltages v_d and v_c are defined by

$$v_d = v_1 - v_2 \qquad (1)$$
$$v_c = \frac{v_1 + v_2}{2}$$

Consistent with power conservation, the corresponding differential and common currents, represented as i_d and i_c respectively, are defined by

$$i_d = \frac{i_1 - i_2}{2} \qquad (2)$$

$$i_c = i_1 + i_2$$

Equations (1) and (2) can be concisely represented as

$$v = Qe \qquad (3)$$

$$i = Q^{-T}u$$

where

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \qquad i = \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} \qquad e = \begin{bmatrix} v_d \\ v_c \end{bmatrix} \qquad u = \begin{bmatrix} i_d \\ i_c \end{bmatrix}$$

- Q is the real, non-singular matrix defined by

$$Q = \begin{bmatrix} \frac{1}{2} & 1 \\ -\frac{1}{2} & 1 \end{bmatrix} \qquad (4)$$

- Q^{-T} is the inverse transpose of Q defined by

$$Q^{-T} = \begin{bmatrix} 1 & \frac{1}{2} \\ -1 & \frac{1}{2} \end{bmatrix} \qquad (5)$$

Using eq. (3), we may easily transform between single-ended and differential quantities.

Differential Admittance and Impedance Matrices

The terminal admittance (Y) and impedance (Z) matrices discussed in the previous topics in the *Technical Notes* relate single-ended voltages and currents as $i = Yv$ and $v = Zi$, respectively. By first making differential voltage and current definitions, e and u eq. (3) can be used to derive Y_d and Z_d matrices that relate those qualities.

For example, if $i = Yv$, then substituting eq. (3) yields

$$Q^{-T}u = YQe.$$

Solving for u yields

$$u = Q^{T}YQe$$

and the matrix Y_d relating differential quantities e and u is defined by

$$Y_d = Q^{T}YQ \qquad (6)$$

A similar procedure applies to the terminal Z_d.

Differential S-Matrices

It is clear that an S-matrix can be computed for differential signals because it is possible to compute admittance and impedance matrices for differential signals. The differential S-matrix can be envisioned as relating in-going and out-going waves on imaginary transmission lines attached to the differential ports. Characteristic impedances must be specified for these lines unless they are considered to be matched loaded which is automatically the case when the waveport(s) that contain the single-ended terminals used to define the differential pair is not renormalized.

In the single-ended case, the characteristic impedance for a pair of transmission lines may be written in the form of a matrix relating the voltages and currents on the two (uncoupled) lines,

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} Z^{(1)}_{ref} & 0 \\ 0 & Z^{(2)}_{ref} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} \qquad (7)$$

where $Z^{(1)}ref$ and $Z^{(2)}ref$ are the user-specified renormalizing impedances. In the differential case, the matrix equation relating differential and common currents and voltages is written as:

$$\begin{bmatrix} v_d \\ v_c \end{bmatrix} = \begin{bmatrix} Z^{(d)}_{ref} & 0 \\ 0 & Z^{(c)}_{ref} \end{bmatrix} \begin{bmatrix} i_d \\ i_c \end{bmatrix} \qquad (8)$$

In this case, $Z^{(d)}ref$ and $Z^{(c)}ref$ denote the user-specified differential and common renormalizing impedances, respectively.

Floquet Ports Theory

The Floquet port in HFSS is used exclusively with planar-periodic structures. Chief examples are planar phased arrays and frequency selective surfaces when these may be idealized as infinitely large. The analysis of the infinite structure is then accomplished by analyzing a unit cell. Linked boundaries most often form the side walls of a unit cell, but in addition, at least one "open" boundary condition representing the boundary to infinite space is needed. The Floquet port is a specialized boundary condition to handle this case.

The Floquet port is closely related to a Wave port in that a set of modes, here "Floquet modes", is used to represent the fields on the port boundary. Fundamentally, Floquet modes are plane waves with propagation direction set by the frequency and geometry of the periodic structure. Just like Wave modes, Floquet modes have propagation constants and experience cut-off at a sufficiently low frequency.

When a Floquet port is present, HFSS performs a modal decomposition that gives additional information on the performance of the radiating structure. As in the case of a Wave port, this information is cast in the form of an S-matrix interrelating the Floquet modes. In fact, if Floquet ports and Wave ports are simultaneously present, the S-matrix will interrelate all Wave modes and all Floquet modes in the model. Floquet ports can be combined with lumped ports, but not with terminal ports.

Related Topics

Technical Notes: Formula Summary for HFSS Floquet Modes

Formula Summary for HFSS Floquet Modes

- Planar array lattice is specified by lattice vectors **a** and **b**.
- Lattice angle α defined by $a * b = ab\cos\alpha$
- Unit cell area A
- Global material is lossless, isotropic, and homogeneous, and specified by real ε_r and μ_r.
- Frequency is specified in the form of free-space wave number $k_0 = \frac{2\pi f}{c}$ where f is frequency and c is the speed of light in a vacuum.
- Local coordinate system (LCS) unit vectors $\hat{x}$ and $\hat{y}$ are related to lattice vectors by:

$$\hat{\mathbf{x}} = \hat{\mathbf{a}} \quad (1)$$

$$\mathbf{c} = \mathbf{b} - (\mathbf{b} \cdot \hat{\mathbf{a}})\hat{\mathbf{a}} \quad (2)$$

$$\hat{\mathbf{y}} = \frac{\mathbf{c}}{c} \quad (3)$$

where $\hat{a} = \frac{a}{}$

Unit vector $\hat{r}_s a$ denotes scan direction. In the global coordinate system (GCS) the components of $\hat{r}_s$ are given by ($\sin\theta_s\cos\phi_s$, $\sin\theta_s\sin\theta\phi_s$, $\cos\theta_s$) where scan angles θ_s and ϕ_s are specified as spherical angles in the GCS.

$$\psi_a = \sqrt{\mu_r\varepsilon_r}k_s\hat{\mathbf{r}} \cdot \mathbf{a} \quad (4)$$

$$\psi_b = \sqrt{\mu_r\varepsilon_r}k_0\hat{\mathbf{r}}_s \cdot \mathbf{b} \quad (5)$$

$$m,n\varepsilon\{\ldots,-2,-1,0,1,2,\ldots\} \quad (6)$$

$$k_a = \frac{2\pi m}{a} + \frac{\Psi_a}{a} \qquad \text{If} \quad k_z \neq 0 \tag{7}$$

$$k_b = \frac{2\pi n}{b} + \frac{\Psi_b}{ab} \tag{8}$$

$$k_x = k_a \tag{9}$$

$$k_y = \frac{k_b}{\sin\alpha} - k_a \cot\alpha \tag{10}$$

$$k_c^2 = k_x^2 + k_y^2 \tag{11}$$

$$k_z = \sqrt{\mu_r \varepsilon_r k_0^2 - k_c^2} \tag{12}$$

$$\Psi_{mn}(x,y) = \sqrt{\frac{1}{A}} e^{-jk_x x} e^{-jk_y y} \tag{13}$$

$$\Psi_{mn}^{(1)} = \left[\frac{k_y \hat{\mathbf{x}} - k_x \hat{\mathbf{y}}}{k_c}\right] \psi_{mn}(x,y) \qquad (k_c \neq 0) \tag{14}$$

$$\Psi_{mn}^{(2)} = \left[\frac{k_x \hat{\mathbf{x}} + k_y \hat{\mathbf{y}}}{k_c}\right] \psi_{mn}(x,y) \qquad (k_c \neq 0) \tag{15}$$

$$\tilde{\Psi}_{mn}^{(1)} = \sqrt{\frac{1}{A}} (\sin\theta_s \hat{\mathbf{x}} - \cos\theta_s \hat{\mathbf{y}}) \qquad (k_c \neq 0) \tag{16}$$

$$\tilde{\Psi}_{mn}^{(2)} = \sqrt{\frac{1}{A}} (\sin\theta_s \hat{\mathbf{x}} + \cos\theta_s \hat{\mathbf{y}}) \qquad (k_c \neq 0) \tag{17}$$

$$\vec{\mathbf{e}}_{mn}^{TE} = N_{mn}^{TE} \Psi_{mn}^{(1)} \tag{18}$$

$$\vec{\mathbf{h}}_{mn}^{TE} = Y_{mn}^{TE} \hat{\mathbf{z}} \times \vec{\mathbf{e}}_{mn}^{TE} \tag{19}$$

$$Y_{mn}^{TE} = \frac{k_z}{\mu_0 k_0} \frac{1}{\eta_0} \tag{20}$$

$$N_{mn}^{TE} = \sqrt{\mu_r k_0} \frac{1}{\sqrt{|k_z|}} \sqrt{\eta_0} \tag{21}$$

$$\hat{\mathbf{z}} e_{mn}^{TE} = 0 \tag{22}$$

$$\hat{\mathbf{z}} h_{mn}^{TE} = -\hat{\mathbf{z}} Y_{mn}^{TE} \quad N_{mn}^{TE} \quad \left(\frac{k_c}{k_z}\right) \psi_{mn} \tag{23}$$

$$\vec{\mathbf{e}}_{mn}^{TE} = N_{mn}^{TM} \Psi_{mn}^{(2)} \tag{24}$$

$$\vec{\mathbf{h}}_{mn}^{TM} = Y_{mn}^{TM} \hat{\mathbf{z}} \times \vec{\mathbf{e}}_{mn}^{TM} \tag{25}$$

$$Y_{mn}^{TM} = \frac{\varepsilon_r k_0}{k_z} \frac{1}{\eta_0} \tag{26}$$

$$N_{mn}^{TM} = \frac{1}{\sqrt{\varepsilon_r k_0}} \frac{k_z}{\sqrt{|k_z|}} \sqrt{\eta_0} \tag{27}$$

$$\hat{\mathbf{z}} e_{mn}^{TM} = -\hat{\mathbf{z}} N_{mn}^{TM} \quad \left(\frac{k_c}{k_z}\right) \psi_{mn} \tag{28}$$

$$\hat{\mathbf{z}} h_{mn}^{TM} = 0 \tag{29}$$

If $k_Z = 0$, use the preceding $k_z \neq 0$ formulas with $k_Z = \alpha$ where $0 < \alpha << 1$ is a small constant.
If $k_C = 0$

$$\vec{\mathbf{e}}_{mn}^{TEM1} = N^{TEM} \tilde{\Psi}^{(1)} \tag{30}$$

$$\vec{\mathbf{h}}_{mn}^{TEM1} = Y^{TEM} \hat{\mathbf{z}} \times \vec{\mathbf{e}}_{mn}^{TEM1} \tag{31}$$

$$\vec{\mathbf{e}}_{mn}^{TEM1} = N^{TEM}\tilde{\Psi}^{(2)} \tag{32}$$

$$\vec{\mathbf{h}}_{mn}^{TEM2} = Y^{TEM}\hat{\mathbf{z}} \times \vec{\mathbf{e}}_{mn}^{TEM2} \tag{33}$$

$$Y^{TEM} = \sqrt{\frac{\varepsilon_r}{\mu_r}}\frac{1}{\eta_0} \tag{34}$$

$$N^{TEM} = \frac{1}{\sqrt{Y^{TEM}}} \tag{35}$$

Incident Waves Theory

There are different kinds of incident wave sources in HFSS. Analytical wave forms are defined by formulas. The following analytical wave forms exist:

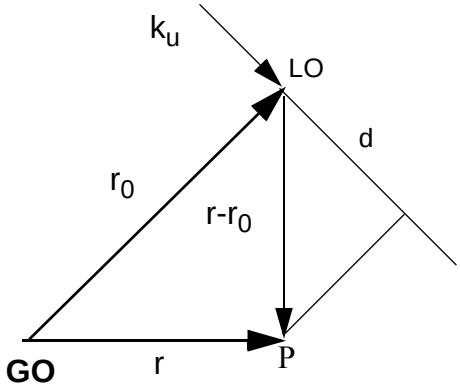
- Plane Wave
- Evanescent Plane Wave
- Electric Spherical Wave (Electric Hertzian Dipole)
- Magnetic Spherical Wave (Magnetic Hertzian Dipole)
- Cylindrical Wave
- Gaussian Beam
- Linear Antenna

Numerical Wave Forms can come from the solution of another HFSS project or from a non-HFSS project via DataLink. Two numerical wave forms exist:

- Near Field Wave
- Far Field Wave

Plane Wave

An incident plane wave is a wave that propagates in one direction and is uniform in the directions perpendicular to its direction of propagation. The angle at which the incident wave impacts the device is known as the angle of incidence.



The equation that HFSS uses to calculate the incident wave is

$$\boldsymbol{E}_{inc} = \boldsymbol{E}_0 e^{-j\boldsymbol{k}_0 k_u(\hat{\boldsymbol{r}} \cdot \boldsymbol{r}_0)}$$

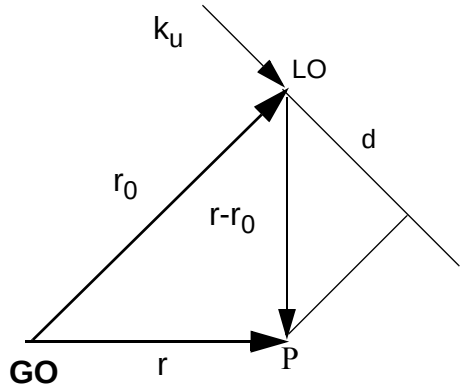
where

- E_{inc} is the incident wave.
- E_0 is the E-field polarization vector.
- k_0 is the free space wave number. It is equal to $\omega\sqrt{\mu_0\varepsilon_0}$.
- $\hat{k}$ is the propagation vector. It is a unit vector.
- r is the position vector and is equal to $x\hat{x} + y\hat{y} + z\hat{z}$.

Incident wave excitations are specified in a peak sense. That is, if the incident wave magnitude is 5 V/m, then the real time function of the incident field is $\mathbf{E}(\mathbf{t}) = 5\cos(\boldsymbol{k}*\boldsymbol{r}+\omega t)$.

Evanescent Plane Wave

An evanescent plane wave resembles the regular plane wave except that it decays in one direction. Its propagation constant is complex (see the equations).



$$d = \left|k_u \bullet (r - r_0)\right| \qquad \alpha = Re(|k|)$$

$$\beta = Im(|k|) \qquad \vec{E} = \vec{E}_0 e^{-\alpha d} e^{-jkd}$$

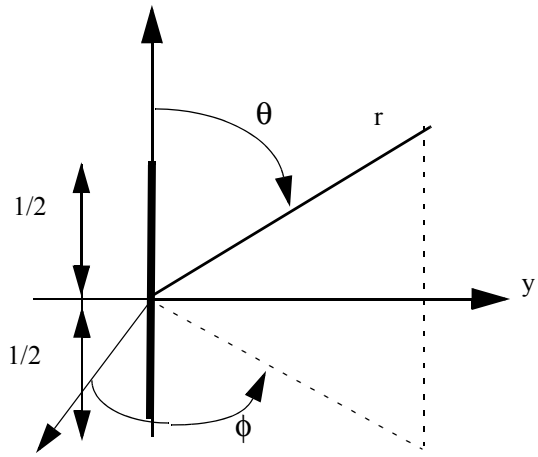
$$\vec{H}_0 = \frac{k}{\omega\mu_0\mu_r}(\vec{k_\mu} \times \vec{E}_0) = \frac{\beta - j\alpha}{\omega\mu_0\mu_r}(\vec{k_\mu} \times \vec{E})$$

Related Topics

Incident Waves

Electric Spherical Wave (Electric Hertzian Dipole)

The analytical formulas of an electric Hertzian dipole are:



$$H_r = 0 \qquad H_\theta = 0$$

$$H_\phi = j\frac{kI_0 l \sin\theta}{4\pi r}\left[1 + \frac{1}{jkr}\right]e^{-jkr} \qquad E_r = \eta\frac{I_0 l \cos\theta}{2\pi r^2}\left[1 + \frac{1}{jkr}\right]e^{-jkr}$$

$$E_\theta = j\eta\frac{kI_0 l \sin\theta}{4\pi r}\left[1 + \frac{1}{jkr} - \frac{1}{(kr)^2}\right]e^{-jkr} \qquad E_\Phi = 0$$

Related Topics

Incident Waves

Magnetic Spherical Wave (Magnetic Hertzian Dipole)

The analytical formulas of a magnetic Hertzian dipole are:

$$E_r = 0 \qquad E_\theta = 0$$

$$E_\phi = -j\frac{kI_0 l \sin\theta}{4\pi r}\left[1 + \frac{1}{jkr}\right]e^{-jkr} \qquad H_r = \frac{I_m l \cos\theta}{2\pi\eta r^2}\left[1 + \frac{1}{jkr}\right]e^{-jkr}$$

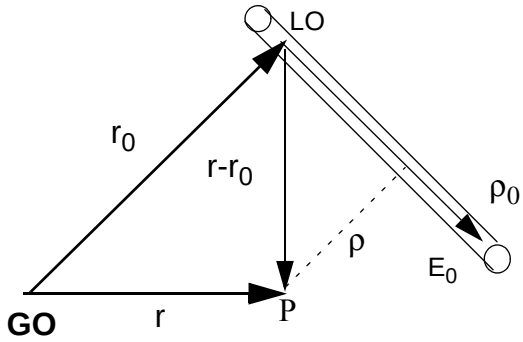
$$H_\theta = j\frac{kI_m l \sin\theta}{4\pi\eta r}\left[1 + \frac{1}{jkr} - \frac{1}{(kr)^2}\right]e^{-jkr} \qquad H_\Phi = 0$$

Related Topics

Incident Waves

Cylindrical Wave

The analytical formulas of a Cylindrical Wave are:



$$E_z(\rho) = \frac{C_1 e^{-jk\rho}}{\sqrt{\rho}} \qquad \text{where} \qquad C_1 = ZI^e \sqrt{k/8\pi}\, e^{j\pi/4}$$

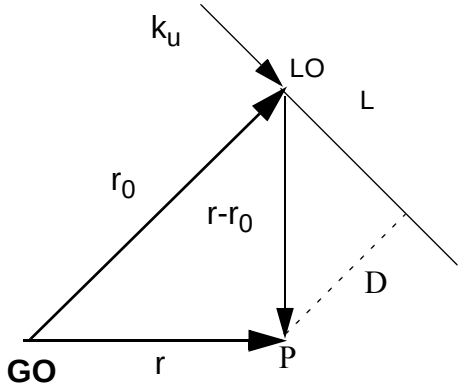
$$H_\phi(\rho) = \frac{C_2 e^{-jk\rho}}{\sqrt{\rho}} \qquad \text{with} \qquad C_2 = C_1/Z$$

Related Topics

Incident Waves

Gaussian Beam

The analytical formulas of a Gaussian Beam are:



Where

$$L = (r - r_0) \bullet k_u D = \left|(r - r_0) \times k_u\right|$$

$$W(L) = W_0\left[1 + \left(\frac{L\lambda}{\pi W_0^2\, n}\right)^2\right]^{0.5}$$

$$R(L) = L\left[1 + \left(\frac{\pi W_0^2\, n}{L\lambda}\right)^2\right] \qquad \Psi(L) = \operatorname{atan}\left[\frac{L\lambda}{\pi W_0^2\, n}\right]^2$$

$$E^{inc} = E_0 \frac{W_0}{W(L)} e^{-D^2/W(L)^2} e^{\left(-jk \bullet (r - r_0) - jk\frac{D^2}{2R(L)} + j\Psi(L)\right)}$$

For HFSS Transient, the Gaussian beam calculation is:

$$E = E_0 \times \exp(-tt^2) \tag{1}$$

where

$$tt = 2 \times f_{max} \times \frac{t - 1.5}{f_{max}} \tag{2}$$

where

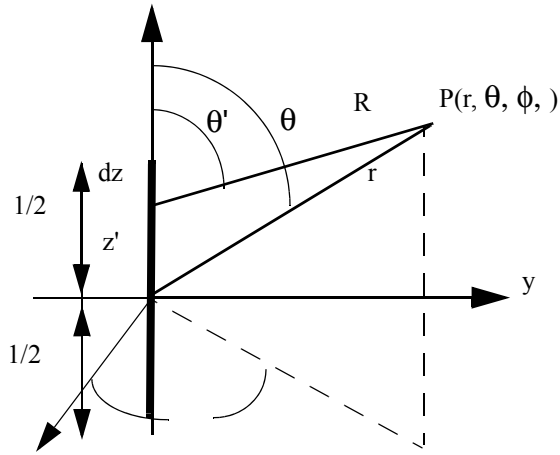
$$f_{max} = \frac{0876}{t_{rise}} \tag{3}$$

Related Topics

Incident Waves

Linear Antenna

The analytical formulas of a linear antenna are:



$$E_\theta \cong j\eta \frac{I_0 e^{-jkr}}{2\pi r}\left[\frac{\cos\left(\frac{kl}{2}\cos\theta\right) - \cos\left(\frac{kl}{2}\right)}{\sin\theta}\right]$$

$$H_\phi \cong \frac{E_\theta}{\eta} \cong j\frac{I_0 e^{-jkr}}{2\pi r}\left[\frac{\cos\left(\frac{kl}{2}\cos\theta\right) - \cos\left(\frac{kl}{2}\right)}{\sin\theta}\right]$$

Related Topics

Incident Waves

Voltage Sources

A voltage source in HFSS can be defined on surface located anywhere in the 3D problem space, but it typically makes sense to place the source on a surface between two conductors such that a user defined total voltage is maintained between the conductors.

A voltage source is implemented in HFSS by forcing the electric field on the source surface to maintain a user-specified voltage drop along the voltage line. You can specify any surface as the source surface, but the prescribed field pattern is best suited for rectangular planar surfaces or non-planar surfaces obtained from the side wall surface of cylinders with uniform cross sections such as a circular cylinder. The enforced electric field pattern is obtained by projecting a static uniform field onto the source surface with the electric field in the direction of the user defined voltage line.

A voltage source cannot be used to extract S-parameters. For S-parameter computations of an HFSS design involving a mixture of ports and voltage sources, all the voltage sources are shorted by treating the voltage source surface as a perfect electric conductor.

Current Sources

A current source in HFSS can be defined on a surface located anywhere in the 3D problem space, but it typically makes sense to place the source on a surface between two conductors such that it injects the user defined total current onto the conductors. A current source is implemented in HFSS by weakly enforcing the jump across the source surface of the tangential component of the magnetic field, $\Delta\vec{H}$, to be related to a surface current density, $\vec{Js}$:

$$\hat{n} \times \Delta\vec{H} = \vec{Js}$$

where $\hat{n}$ is the unit normal of the surface. The surface current density can be expressed as $\vec{Js} = J_0\hat{u}$ where J_0 is the magnitude and $\hat{u}$ is the unit vector in the direction of current flow.

The unit vector is in the direction of the user defined current flow direction line. The magnitude is obtained by first assuming that the surface is a rectangle such that the width, *w*, is equal to the area divided by the length of the current flow direction line. The magnitude is set equal to the total current specified by the user divided by *w*. Note that any source surface is allowed but considering the definition of the surface current density it is recommended to only use a planar rectangular surface.

A current source cannot be used to extract S-parameters. For S-parameter computations of an HFSS design involving a mixture of ports and current sources, all the current sources are kept opened by treating the current source surface as a natural boundary condition where no special behavior of the electric field is explicitly enforced.

Magnetic Bias Sources

When you create a ferrite material, you must define the net internal field that biases the ferrite by assigning a magnetic bias source. The bias field aligns the magnetic dipoles in the ferrite, producing a non-zero magnetic moment. In HFSS, a ferrite's permeability tensor is a direct result of an applied static magnetic bias field. The static field causes the tensor to assume an hermitian form, with cross coupling terms between field components perpendicular to the bias.

When the applied bias field is assumed to be uniform, you will specify net internal bias field as well as the tensor coordinate system through a rotation from the global coordinate system. When the applied bias field is non-uniform, neither specified coordinate system rotations nor specified bias are allowed. The permeability tensor's local coordinate system and bias field are calculated on a tetrahedron by tetrahedron basis, both of which are determined by the calculated static solution. HFSS references the static solution project as the source of the non-uniform magnetostatic field information during solution generation.

Multiple uniform bias excitations may be included in a design, but only non-uniform bias is allowed. In addition, a design cannot contain a bias excitation of both types. If only one excitation is included in a design, its type can be changed from uniform to non-uniform, and visa versa.

Related Topics

Technical Notes: Uniform Applied Bias Fields

Technical Notes: Non-uniform Applied Bias Fields

Uniform Applied Bias Fields

The applied DC bias that causes ferrite saturation is always in the positive z direction of the tensor coordinate system. Initially the tensor coordinate system is assumed to be aligned with the fixed coordinate system; the tensor's z-axis is the same as the model's z-axis. To model other directions of applied bias, the permeability tensor must be rotated so that its z-axis lies in another direction on the fixed coordinate system. This is accomplished by specifying the rotation angles about the axes when you assign a magnetic bias source to a model surface.

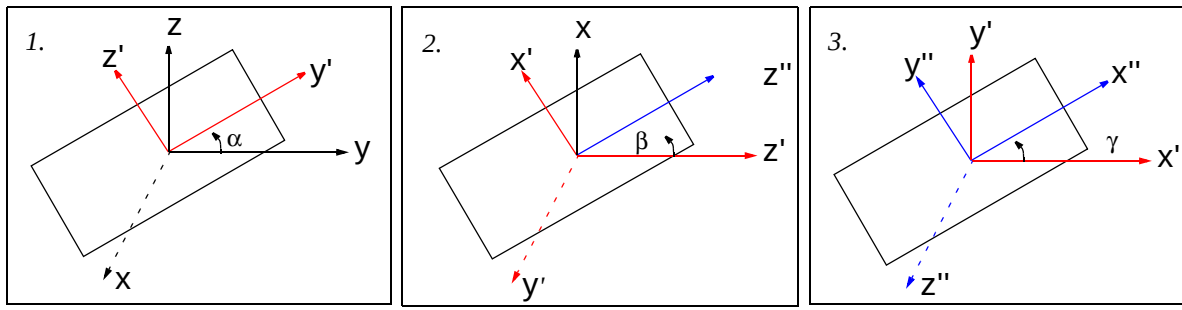
The rotation angles should be defined in the **Magnetic Bias Source** dialog box in such a way that the tensor coordinate system is obtained in the following manner:

1. Rotating the tensor coordinate system by α degrees (from the **X Angle**) around the fixed x-axis.
2. Rotating the resulting tensor coordinate system by β degrees (from the **Y Angle**) around the

new y-axis.

3. Rotating the new tensor coordinate system by γ degrees (from the **Z Angle**) around the new z-axis.

This concept is illustrated in the following graphic. In the first panel, the permeability tensor is rotated α degrees about the x-axis. In the second panel, the tensor is rotated β degrees about the y'-axis (the new y-axis). In the third panel, the tensor is rotated γ degrees about the z''-axis (the new z-axis). The resulting tensor has the coordinate system (x''y''z'') relative to the fixed coordinate system.



For example, to model the DC bias in the x direction you would rotate the tensor coordinate system so that its z-axis lies along the x-axis of the fixed coordinate system. To do this you would enter **0** for the **X Angle**, **90** for the **Y Angle**, and **0** for the **Z Angle**.

Non-uniform Applied Bias Fields

To accurately model a ferrite in an applied static magnetic bias field, the non-uniform magnetic bias fields must also be calculated. In HFSS, a ferrite's permeability tensor is a direct result of an applied static magnetic bias field. The static field causes the tensor to assume an hermitian form, with cross coupling terms between field components perpendicular to the bias. A uniform bias field is difficult to achieve in practice. Even if the bias field is nearly uniform, a non-ellipsoidal-shaped ferrite material will have non-uniform demagnetization, resulting in non-uniform fields in the ferrite.

Use the magnetostatic solver provided in the Maxwell 3D Field Simulator to generate a solution for non-uniform magnetostatic fields. Once a solution is generated it may be imported into HFSS.

Note To specify the non-uniform bias field, you must have purchased the Maxwell 3D Field Simulator. Refer to the Maxwell 3D Field Simulator documentation for instructions on solving for non-uniform magnetostatic fields.

Deembedding

Deembedding means adding or subtracting transmission line. In HFSS only wave ports and Floquet ports are allowed to be deembedded where the S-parameters are modified as described in Deembedding S-Matrices. Fields used for visualization and the calculator can also be modified by select-

ing Include Post Processing Effects in the Edit Sources panel. The new results appear instantaneously.

This section shows an example where subtracting transmission line can be useful and an example where adding transmission line can be useful. A following section describes When HFSS Needs Port Calibration in deembedding.

Consider the aperture-coupled patch antenna shown in Fig. 1. A microstrip trace located below a ground plane feeds a patch antenna located above the ground plane. A slot in the ground plane couples power from the microstrip trace to the antenna.

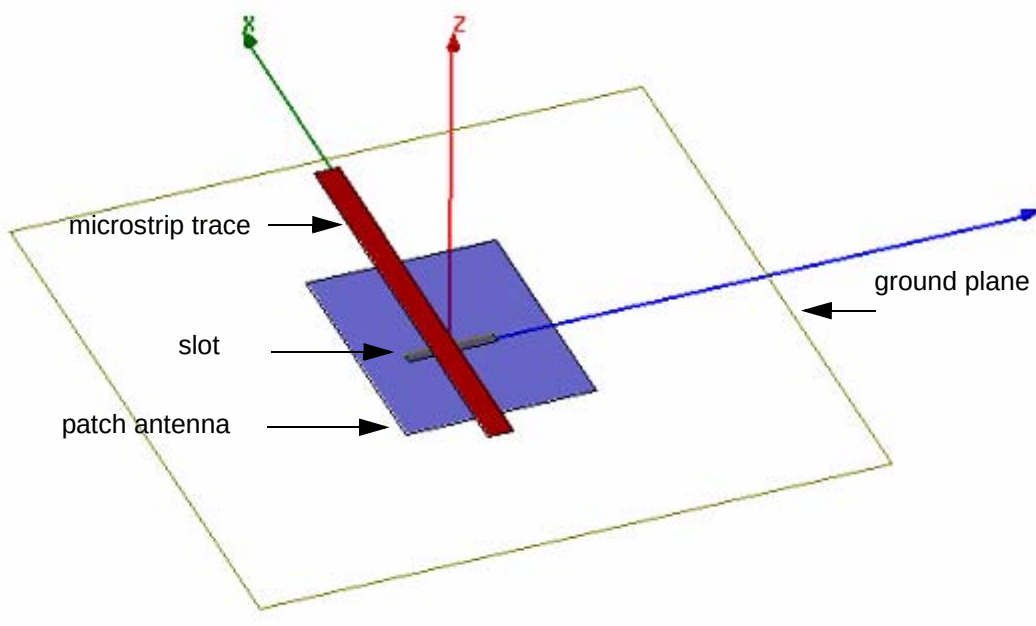


Fig. 1 Aperture-coupled patch antenna

A port is located at the beginning of the trace. It cannot be located very close to the slot, since a port has to be located in a position where a clean transmission-line mode is expected, at some distance from the first discontinuity. The port is shown in Fig. 2. One edge coincides with the ground plane. The other edges are such that the port width is several times the trace width and the port height several times the substrate thickness. The red arrows represent the port field solution. Notice that the

three non-ground-plane edges don't influence the port solution noticeably, so this port is large enough.

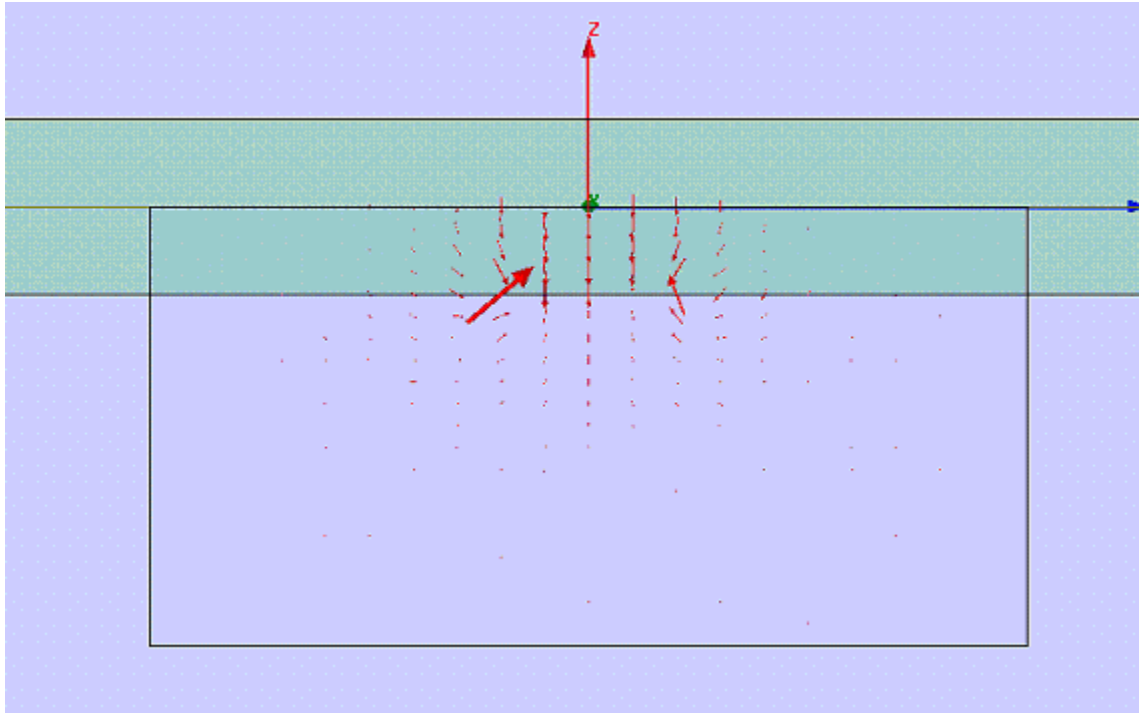


Fig. 2 Magnitude of S11 in dB

The S-parameter characteristic is shown in Fig. 3 in a rectangular plot and in Fig. 4 in a Smith chart. Note that the resonance frequency in Fig. 3 corresponds to the point in Fig. 4 where the curve is closest to the center of the chart. Also note that the curve circles the chart several times.

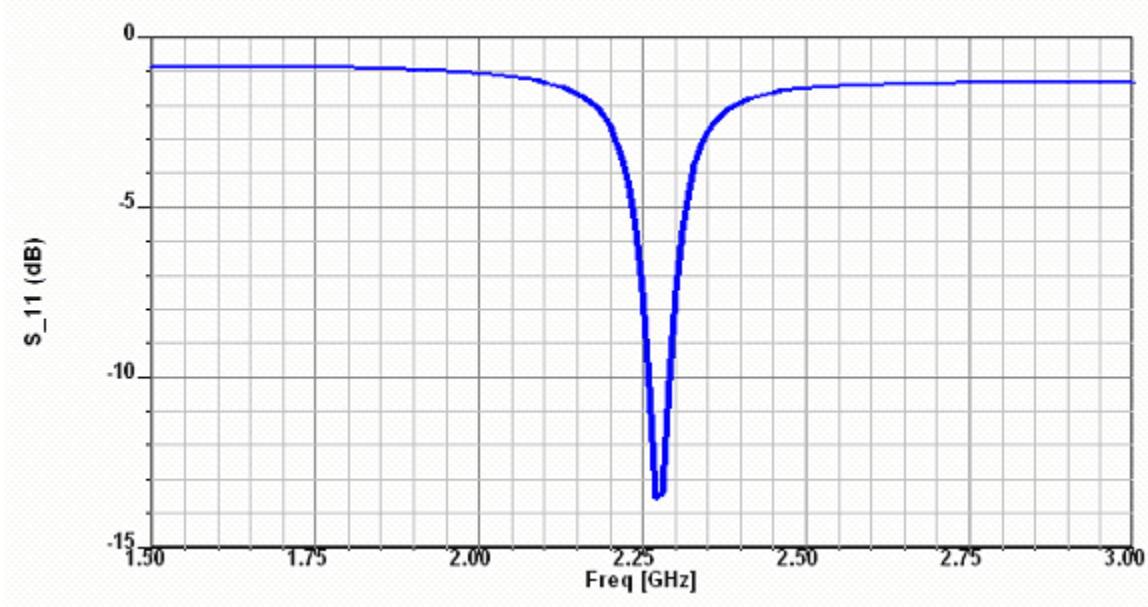


Fig. 3 Magnitude of S11 in dB

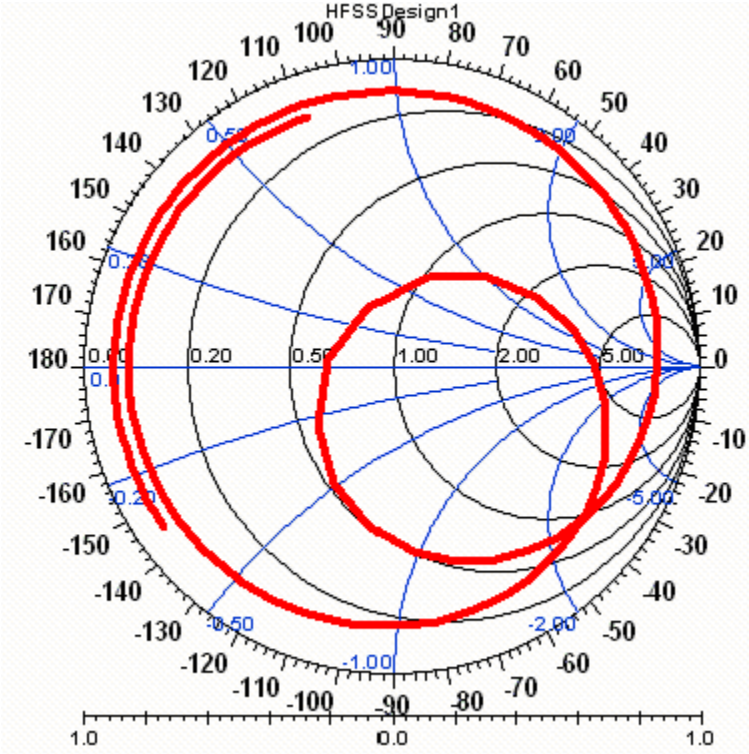


Fig. 4 Smith Chart before deembedding

Antenna designers who need to design a matching circuit close to the feed point may want to see the Smith chart with the reference point located close to the feed point of the antenna, that is. close to the slot, rather than some distance away from the feed point. That can be achieved by deembedding. Fig. 5 shows the Smith chart after deembedding.

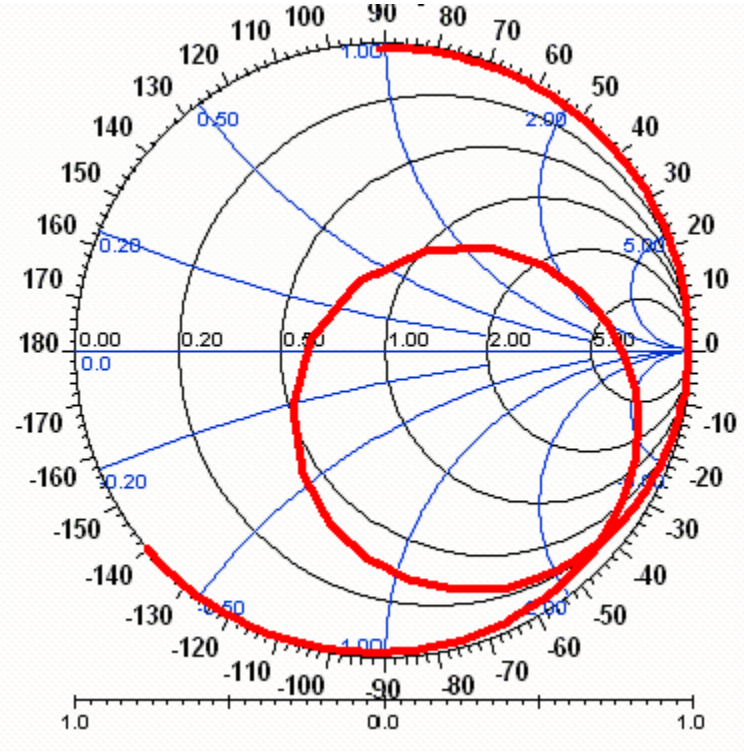


Fig. 5 Smith Chart after deembedding

Note that after this kind of deembedding, where transmission line was removed, the curve circles the Smith chart fewer times.

As an example where adding transmission line may be useful, consider an IC package or a connector mounted on a printed circuit board. Fig. 6 shows a connector on a printed circuit board. The connector designer has included short sections of traces on the board in his model. These are coupled differential lines, and because of the coupling they share one port. Again, there is some distance between the port and the first discontinuity, so the fields that reach the port form clean transmission-line modes. To model the connector accurately, this model is sufficient.

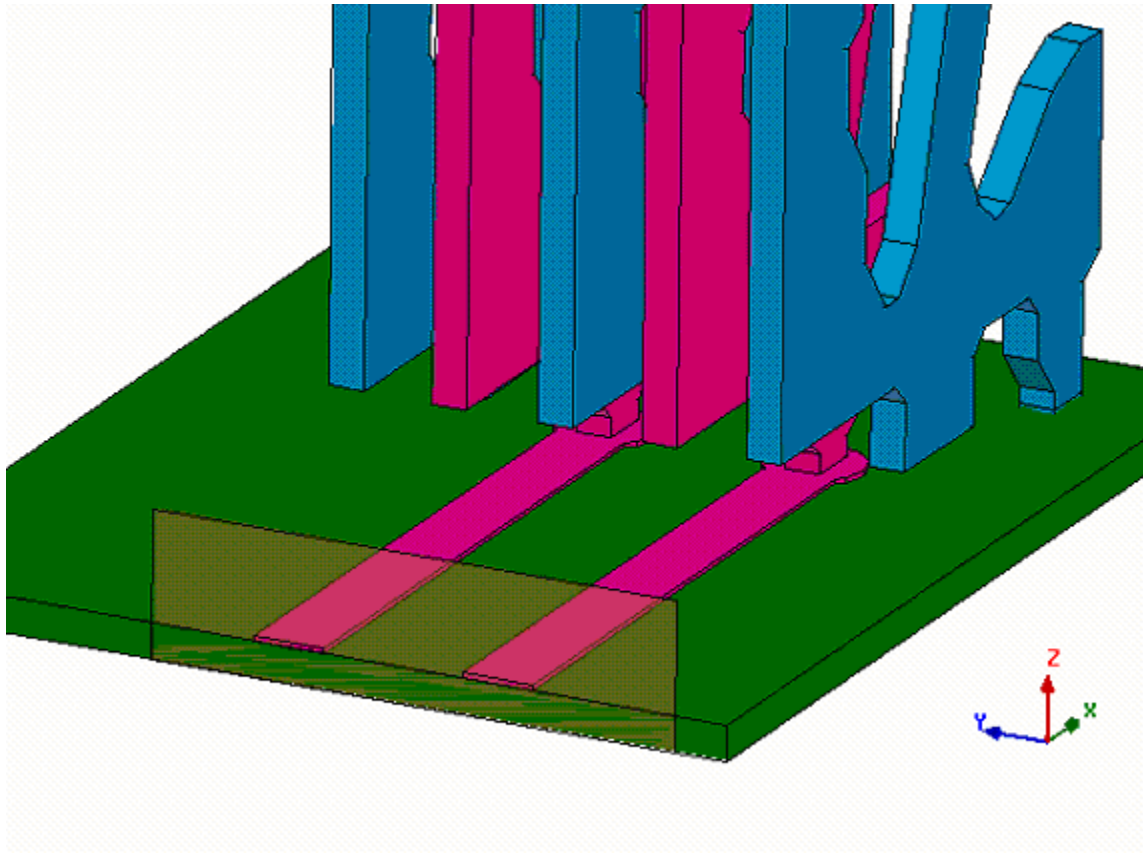


Fig. 6 Connector on printed circuit board

On the board, the traces may continue as uniform transmission lines for some distance. To evaluate system performance, it may be desirable to include that length in the S-parameters that are to be exported to a circuit simulator. With deembedding, uniform transmission line can be added as a matrix post processing step.

Another example where adding transmission-line length through deembedding may be useful is modeling long cables with frequency-dependent materials, for example, in back planes. In HFSS, only a short section would be modeled, and its behavior over the entire frequency range analyzed accurately. Analyzing only a short section has the benefits of both speed and accuracy. Then, with deembedding, the cable can be extended to any desired length. The S-parameters for the full length can be exported to a circuit simulator to be incorporated into a system.

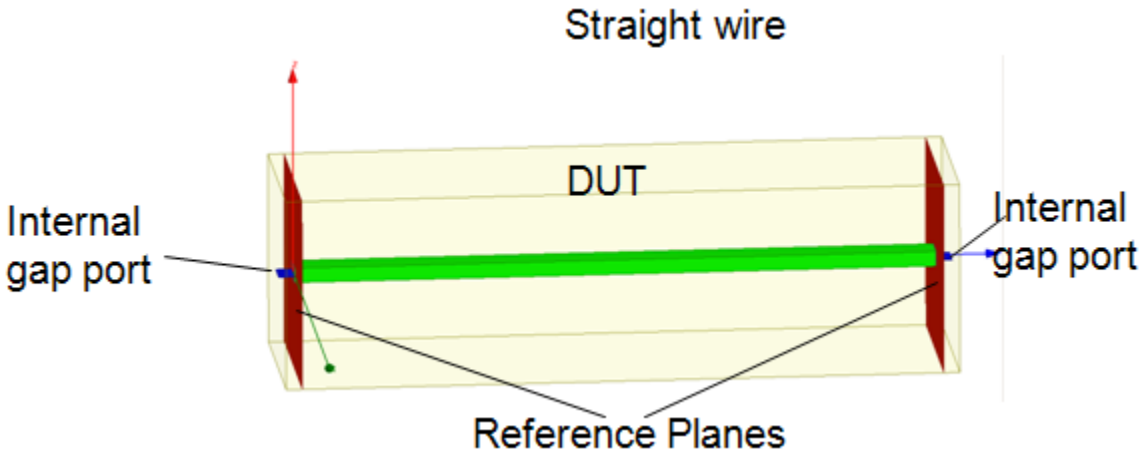
When HFSS Needs Port Calibration (deembedding)

- If regular (perfect) ports are used, HFSS does not need any port calibration
- • If imperfect ports are used (internal lumped port), HFSS also needs some port calibration

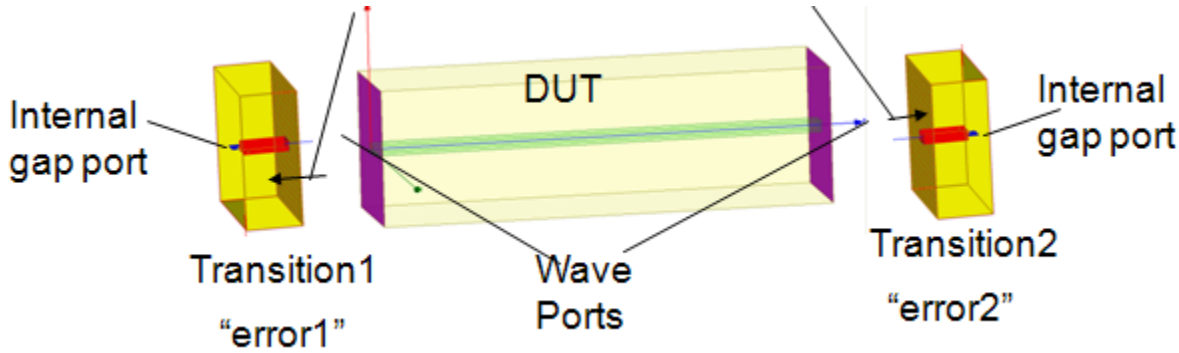
(deembedding)

- If wave ports are used port calibration is not required
- If lump ports are used, parasitic inductance of port geometry can be removed through calibration or de-embedding

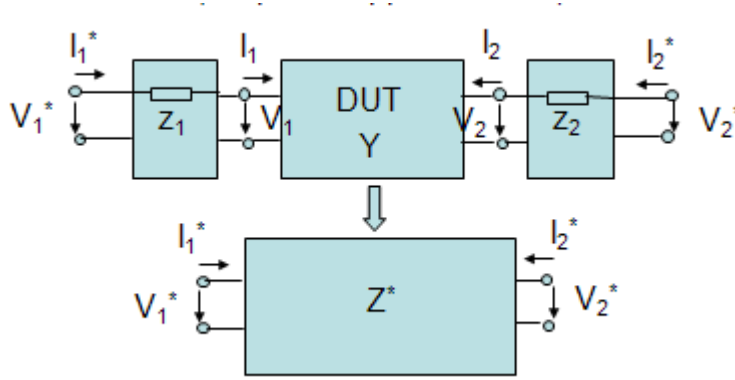
Deembedding extension/transition effects



Deembedded to the discontinuity



Deembedding extension/transition Z matrix method (simplified approximation)



Impedance matrix Z* is meaured/simulated

Where $\mathbf{Z_1}$ and **$\mathbf{Z_2}$** are known, calculate/deembed **Z** matrix

Z Matrix Method

Measured/Calculated

$$\mathbf{U}^* = \mathbf{Z}^*\mathbf{I}^* \qquad V^* = \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} \qquad I^* = \begin{bmatrix} I_1^* \\ I_2^* \end{bmatrix} \qquad Y^* = \begin{bmatrix} Y_{11}^* & Y_{12}^* \\ Y_{21}^* & Y_{22}^* \end{bmatrix} \tag{1}$$

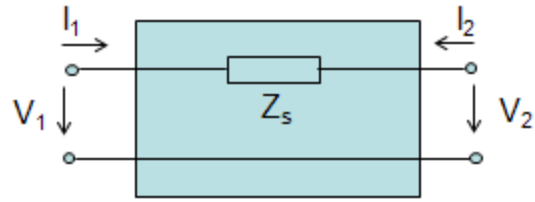
Dembedded

$$\mathbf{U} = \mathbf{ZI} \qquad V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \qquad I = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \qquad Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \tag{2}$$

$$U^* = U + (Z_D)I \qquad Z_D^{\ i} = Z_i \tag{3}$$

$$Z^*I = U + (Z_D)I \qquad U = Z^*I - (Z_D)I \tag{4}$$

S matrix calculation from the series impedances to be deembedded



Experimental Formula

$$L = 0.2 \cdot len \cdot \ln\left(\frac{2len}{(w+t)}\right) + \frac{0.223(w+t)}{len} + 0.5 \quad (5)$$

Where:

Input: w = width of the lumped port [mm]

len = length of the lumped port[mm]

t = 0.0

Output: L[nH]

And

$$Z_s = R_s + jX_s \qquad X_s = \omega L 10^{-9} \quad (6)$$

$$\mathbf{Y'} = (\mathbf{I} + \boldsymbol{S})^{-1}(\mathbf{I} - \mathbf{S}) \quad (7)$$

$$\mathbf{Y} = \langle \frac{1}{\sqrt{Z_o}} \rangle \mathbf{Y'} \langle \frac{1}{\sqrt{Z_o}} \rangle \qquad Z_o = 1\Omega \quad (8)$$

$$\mathbf{Y} = -\begin{bmatrix} \frac{1}{Z_s} & -\frac{1}{Z_s} \\ -\frac{1}{Z_s} & \frac{1}{Z_s} \end{bmatrix} \quad (9)$$

$$Z_s = \frac{1}{Y_{1,1}} \quad (10)$$

$$\mathbf{Z} = \mathbf{Z}^* - \langle \mathbf{Z}_s \rangle \quad (11)$$

Z* is the global impedance matrix

Z is the Global impedance matrix after deembedding

Materials Technical Notes

This section of the *Technical Notes* includes information on the following material properties:

Simple Materials

- Relative Permeability
- Relative Permittivity
- Bulk Conductivity
- Dielectric Loss Tangent
- Magnetic Loss Tangent

Frequency Dependent

- Frequency-dependent material properties.
- Debye Materials
- Frequency-Dependent Material Properties
- Djordjevic-Sarkar Causal Dielectric Model
- Djordjevic-Sarkar Model Parameter Calculation
- Expressions for Permittivity and Conductivity
- Conductivity and Permittivity Limits

Anisotropic materials

- Anisotropic materials
- Anisotropic relative permeability tensors
- Anisotropic relative permittivity tensors
- Anisotropic conductivity tensors
- Anisotropic dielectric loss tangent tensors
- Anisotropic magnetic loss tangent tensors
- Anisotropic materials and ports

Ferrite materials

- Magnetic Saturation
- Lande G Factor
- Delta H

Relative Permeability

The permeability of dielectrics is assumed to be complex, as follows:

$$\mu = \mu' - j\mu''$$

The relative permeability is a dimensionless quantity, defined as follows:

$$\mu' = \mu_r \mu_0$$

where μ_0 is the permeability of free space.

The permeability can also be expressed as

$$\mu = \mu'\left(1 - j\frac{\mu''}{\mu'}\right)$$

where μ' is the real portion of μ and μ''/μ' is the magnetic loss tangent.

Relative Permittivity

The permittivity of dielectrics is assumed to be complex, as follows:

$$\varepsilon = \varepsilon' - j\varepsilon''$$

which can also be expressed as

$$\varepsilon = \varepsilon'\left(1 - j\frac{\varepsilon''}{\varepsilon'}\right)$$

where ε' is the real portion of ε and from which the relative permittivity is defined as:

$$\varepsilon' = \varepsilon_r\varepsilon_0$$

where ε_0 is the permittivity of free space.

$\varepsilon''/\varepsilon'$ is the dielectric loss tangent.

If a material's losses due to bulk conductivity will be significant, such as in semiconductor dielectric materials, an additional bulk conductivity value, σ, must be added. From the time harmonic form of Maxwell's equations, the complex permittivity, ε_c is defined as

$$\nabla \times H = (\sigma + j\omega\varepsilon)E = j\omega\varepsilon_c E$$

where

$$\varepsilon_c = \varepsilon\left(1 - j\tan\delta - j\frac{\sigma}{\omega\varepsilon}\right)$$

Bulk Conductivity

HFSS is capable of including conductivity in the model either as a bulk material loss factor, similar to dielectric loss tangent, or as an impedance boundary condition applied to the outer surfaces of the object. The choice of bulk material loss instead of the boundary condition is made by selecting **Solve Inside** in the **Properties** window.

The choice between bulk material loss and the surface boundary condition is problem dependent. The boundary condition should be applied whenever the conductor is much thicker than the skin depth at the solution frequency. In this case, the unknowns within the conductor are not included in the unknown vector, resulting in a smaller matrix and a faster analysis. However, if the conductor is

not thick relative to the skin depth, the bulk material conductivity must be used to arrive at an accurate solution. With this assumption, the wave equation reduces to

$$\nabla\times\left(\frac{-1}{j\omega\mu}\nabla\times\boldsymbol{E}\right) = (j\omega\varepsilon+\sigma)\boldsymbol{E}$$

The threshold where HFSS will automatically chose whether an object will be solved inside can be accessed via the **Tools>Options>HFSS Options**.

Dielectric Loss Tangent

To represent a dielectric that dissipates the power of a high-frequency electric field, enter a dielectric loss tangent, $\varepsilon''/\varepsilon'$, property value for the material. The smaller the loss tangent, the less lossy the material.

The dielectric loss tangent may vary with frequency. To simulate the variances, define a function for the dielectric loss tangent.If a wide-band frequency sweep is being solved, the Frequency Dependent Material property may be appropriate.

Magnetic Loss Tangent

To represent a dielectric that dissipates the power of a high-frequency magnetic field, enter a magnetic loss tangent, μ''/μ', property value for the material. The smaller the loss tangent, the less lossy the material.

Note If you plan to do a fast frequency sweep for a design that includes dielectrics, make sure that the dielectric or magnetic loss tangent does not vary significantly over the requested frequency range. If they do, the results may not be what you expect. In cases where the loss tangent does vary significantly over the frequency range in which you are interested, copy and solve the design several times, adjusting the loss tangent and associated frequency range for the copied design so that the loss tangent is relatively stable over the design's requested frequency range.

Ferrite Materials

Ferrite materials are used to model the interaction between a microwave signal and a material whose magnetic dipole moments are aligned with an applied bias field. The gyrotropic quality of the ferrite is evident in the permeability tensor which is Hermitian in the lossless case. The Hermitian tensor form leads to the non-reciprocal nature of the devices containing microwave ferrites. If the microwave signal is circularly polarized in the same direction as the precession of the magnetic dipole moments, the signal interacts strongly with the material. When the signal is polarized in the opposite direction to the precession, the interaction will be weaker. Because the interaction between the signal and material depends on the direction of the rotation, the signal propagates through a ferrite material differently in different directions.

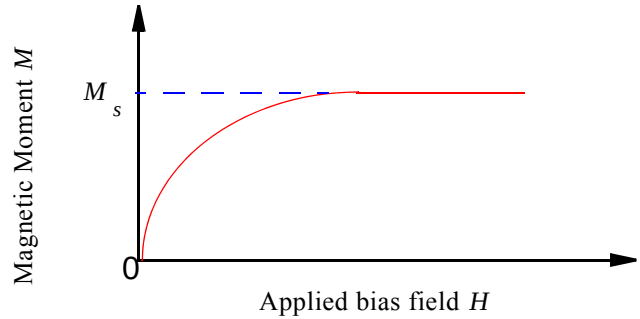
If you assign a ferrite material to an object, you must assign a magnetic bias source to the object.

- Magnetic Saturation
- Lande G Factor

- Delta H
- Ferrite Permeability Tensor in HFSS

Magnetic Saturation

A material with a non-zero magnetic saturation is considered to be a ferrite. When a ferrite is placed in a uniform magnetic field, the magnetic dipole moments of the material begin to align with the field. As the strength of the applied bias field increases, more of the dipole moments align. The magnetic saturation, M_s, is a property that describes the point at which all of the magnetic dipole moments of the material become aligned. At this point, further increases in the applied bias field strength do not result in further saturation. The relationship between the magnetic moment, M and the applied bias field, H, is shown below.



The magnetic saturation, $4\pi M$, is entered in gauss.

Lande G Factor

The Lande g factor is a ferrite property that, on a microscopic level, describes the total magnetic moment of the electrons according to the relative contributions of the orbital moment and the spin moment. When the total magnetic moment is due entirely to the orbital moment, g is equal to one. When the total magnetic moment is due entirely to the spin moment, g is equal to two. For most microwave ferrite materials, g has a range from 1.99 to 2.01. The Lande g factor is dimensionless.

Delta H

Delta H is the full resonance line width at half-maximum, which is measured during a ferromagnetic resonance measurement. It relates to how rapidly a precessional mode in the biased ferrite will damp out when the excitation is removed. The factor ΔH doesn't appear in the permeability tensor; instead, the factor α appears. The factor α is computed from

$$\alpha = -\frac{\gamma \mu_0 \Delta H}{2\omega}$$

The factor α changes the κ and χ terms in the permeability tensor from real to complex, which makes the tensor complex non-symmetric (where it had been hermitian for lossless ferrites).

Enter the full resonance line width at half maximum in the **Delta H** value box. Delta H is measured at a specific frequency. That frequency needs to be entered by the user at the interface. 9.4 GHz is the frequency where Delta H is typically measured.

Ferrite Permeability Tensor in HFSS

Gyrotropic Permeability

The ferrite capability of HFSS is based on the Polder susceptibility tensor small signal approximation of the Landau-Lifshitz equation of motion of a magnetic dipole in a uniform bias field [1] [2].

$$\bar{\mu} = \mu_0 \begin{bmatrix} 1+\chi & j\kappa & 0 \\ -j\kappa & 1+\chi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Where

$$\chi = \frac{\omega_o \omega_m}{\omega_o^2 - \omega^2} \quad (2)$$

$$\kappa = \frac{\omega \omega_m}{\omega_o^2 - \omega^2} \quad (3)$$

With

$$\omega = 2\pi f \quad (4)$$

$$\omega_o = -\gamma \mu_o H_o \quad (5)$$

$$\omega_o = -\gamma \mu_o M_s \quad (6)$$

And

$$\gamma = g_l \gamma_e \quad (7)$$

γ_e is half of the electron charge to mass ratio and g_l is the Lande g factor. The Lande g factor is typically between 1 and 2, with 1 corresponding to orbital angular momentum and 2 for spin.

If the ferrite has magnetic losses, we replace ω_o by $\omega_o + j\omega\alpha$ where α is computed from the ferromagnetic resonance linewidth:

$$\alpha = \frac{\mu_o \gamma \Delta H}{4\pi f_{FMR}} \quad (8)$$

When HFSS assembles the finite element matrices for ferrite materials it computes the permeability tensor, 1, based on several different inputs:

1. **Frequency** - ω
2. **Material properties** - all of which are specified in the material manager
   a. Saturation Magnetization - M_s

b. Lande g factor - g_l

c. Loss factor - computed from ΔH and f_{FMR}

3. **Magnetostatic bias field** - Magnetic Bias source, either:

a. Uniform bias - H_o and direction specified in the interface

b. Non-uniform bias - H_o and local tensor $\hat{z}$ direction determined by the magnetostatic field solution from Maxwell3D. When the Magnetic Bias source is nonuniform, the permeability tensor will be different in each ferrite tetrahedron.

References

[1] David Pozar, Microwave Engineering, Addison-Wesley, 1990.

[2] Daniel D. Stancil, Theory of Magnetostatic Waves, Springer-Verlag, 1992.

Anisotropic Materials

Anisotropic materials have characteristics that vary with direction. These characteristics are defined by their anisotropy tensors. You must define three diagonals each for anisotropic permittivity, electric loss tangent, conductivity, permeability, and magnetic loss tangent. Each diagonal represents a tensor of your model along an axis. The alignment of the materials axis with respect to the object is discussed in Assigning Material Property Types.

- Anisotropic relative permeability tensors
- Anisotropic relative permittivity tensors
- Anisotropic conductivity tensors
- Anisotropic dielectric loss tangent tensors
- Anisotropic magnetic loss tangent tensors
- Anisotropic materials and ports

Related Topics

Assigning Material Property Types

Setting Coordinate Systems

Creating a Relative Coordinate System

Change the Orientation of an object

Relative Permeability

Anisotropic Relative Permeability Tensors

Relative Permittivity

Bulk Conductivity

Dielectric Loss Tangent

Magnetic Loss Tangent

Anisotropic Relative Permeability Tensors

The permeability tensor for an anisotropic material is described by

$$[\mu] = \begin{bmatrix} \mu_1\mu_0 & 0 & 0 \\ 0 & \mu_2\mu_0 & 0 \\ 0 & 0 & \mu_3\mu_0 \end{bmatrix}$$

where

- μ_1 is the relative permeability along one axis of the material's permeability tensor.
- μ_2 is the relative permeability along the second axis.
- μ_3 is the relative permeability along the third axis.
- μ_0 is the permeability of free space.

The relationship between *B* and *H* is:

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow v \\ z \rightarrow w \end{matrix} \quad \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = [\mu] \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix}$$

To specify the relative permeability for an anisotropic material, enter the μ_1, μ_2, and μ_3 values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. If the relative permeability is the same in all directions, use the same value for μ_1, μ_2, and μ_3. These values can also be entered as variables.

Anisotropic Relative Permittivity Tensors

The permittivity tensor for an anisotropic material is described by

$$\varepsilon = \begin{bmatrix} \varepsilon_1\varepsilon_0 & 0 & 0 \\ 0 & \varepsilon_2\varepsilon_0 & 0 \\ 0 & 0 & \varepsilon_3\varepsilon_0 \end{bmatrix}$$

where

- ε_1 is the relative permittivity of the material along one tensor axis.
- ε_2 is the relative permittivity along the second axis.
- ε_3 is the relative permittivity along the third axis.
- ε_0 is the permittivity of free space.

The relationship between *E* and *D* is then

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow v \\ z \rightarrow w \end{matrix} \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = [\varepsilon] \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

To specify the relative permittivity for an anisotropic material, enter the ε_1, ε_2, and ε_3 values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. These values can also be entered as variables.

Anisotropic Conductivity Tensors

The conductivity tensor for an anisotropic material is described by

$$[\sigma] = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$$

where

- σ_1 is the conductivity along one axis of the material's conductivity tensor.
- σ_2 is the conductivity along the second axis.
- σ_3 is the conductivity along the third axis.

The relationship between *J* and *E* is then:

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow v \\ z \rightarrow w \end{matrix} \begin{bmatrix} J_x \\ J_y \\ J_z \end{bmatrix} = [\sigma] \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

To specify the conductivity for an anisotropic material, enter the σ_1, σ_2, and σ_3 values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. The values of σ_1 and σ_2 apply to axes that lie in the xy cross-section being modeled. The values of σ_3 apply to the z-component. These values affect current flowing in dielectrics between the conductors. These values can also be entered as variables.

Anisotropic Dielectric Loss Tangent Tensors

The dielectric loss tangent tensor for an anisotropic material is described by

$$[\bar{\varepsilon}] = \begin{bmatrix} \varepsilon'_1(1 - j\tan\delta_1) & 0 & 0 \\ 0 & \varepsilon'_2(1 - j\tan\delta_2) & 0 \\ 0 & 0 & \varepsilon'_3(1 - j\tan\delta_3) \end{bmatrix}$$

where

- $\tan\delta_1$ is the ratio of the imaginary relative permittivity to the real relative permittivity in one direction.

$$\tan\delta_1 = \frac{\varepsilon''_1}{\varepsilon'_1}$$

- $\tan\delta_2$ is the ratio of the imaginary relative permittivity to the real relative permittivity in the second direction.

$$\tan\delta_2 = \frac{\varepsilon''_2}{\varepsilon'_2}$$

- $\tan\delta_3$ is the ratio of the imaginary relative permittivity to the real relative permittivity in the third orthogonal direction.

$$\tan\delta_3 = \frac{\varepsilon''_3}{\varepsilon'_3}$$

- ε'_1, ε'_2, and ε'_3 are the real relative permittivities specified earlier.ε

The relationship between *D* and *E* will then be

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow v \\ z \rightarrow w \end{matrix} \quad \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = [\bar{\varepsilon}] \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

To specify the electric loss tangent for an anisotropic material, enter the $\tan\delta_1$, $\tan\delta_2$, and $\tan\delta_3$ values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. These values can also be entered as variables.

Anisotropic Magnetic Loss Tangent Tensors

The magnetic loss tangent tensor for an anisotropic material is described by

$$[\bar{\mu}] = \begin{bmatrix} \mu'_1(1 - j\tan\delta_{M1}) & 0 & 0 \\ 0 & \mu'_2(1 - j\tan\delta_{M2}) & 0 \\ 0 & 0 & \mu'_3(1 - j\tan\delta_{M3}) \end{bmatrix}$$

where

- $\tan\delta_{M1}$ is the ratio of the imaginary relative permeability to the real relative permeability in one direction.

$$\tan\delta_{M1} = \frac{\mu''_1}{\mu'_1}$$

- $\tan\delta_{M2}$ is the ratio of the imaginary relative permeability to the real relative permeability in the second direction.

$$\tan\delta_{M2} = \frac{\mu''_2}{\mu'_2}$$

- $\tan\delta_{M3}$ is the ratio of the imaginary relative permeability to the real relative permeability in the third orthogonal direction.

$$\tan\delta_{M3} = \frac{\mu''_3}{\mu'_3}$$

- μ'_1, μ'_2, and μ'_3 are the real relative permeabilities specified earlier.

The relationship between *B* and *H* will then be

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow v \\ z \rightarrow w \end{matrix} \quad \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = [\bar{\mu}] \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix}$$

To specify the magnetic loss tangent for an anisotropic material, enter the $\tan\delta_{M1}$, $\tan\delta_{M2}$, and $\tan\delta_{M3}$ values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. These values can also be entered as functions.

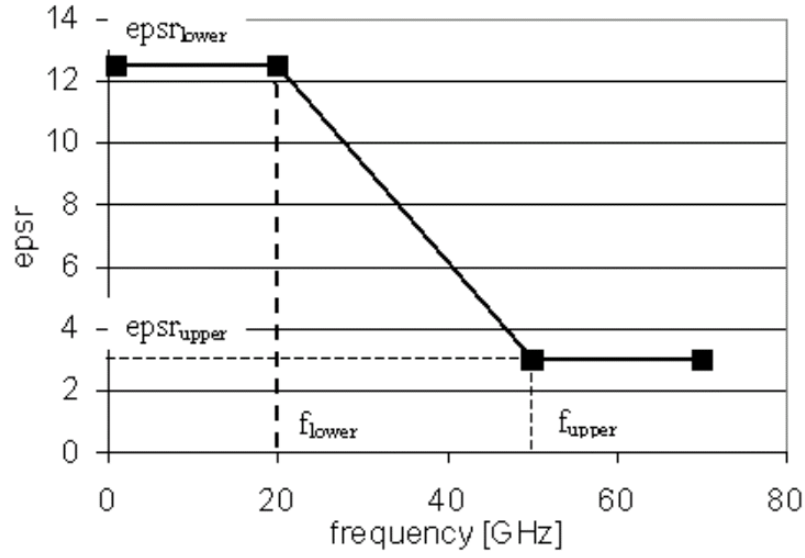
Anisotropic Materials and Ports

An anisotropic material can be in contact with a port under the following conditions:

- One principal axis of the anisotropic material is aligned normal to the port.

Frequency-Dependent Material Properties

The properties of some materials vary with the frequency of the field excitation. This frequency dependence is often linear within a certain frequency range and constant outside of the frequency range, as shown below,



where

- $epsr_{lower}$ is the relative permittivity of a material below the frequency range 20 - 50 GHz.
- $epsr_{upper}$ is the relative permittivity of a material above the frequency range 20 - 50 GHz.
- f_{lower} is the lower frequency, below which the material property is constant.
- f_{upper} is the upper frequency, above which the material property is constant.

In general, to account for this variance within a given frequency range, use the **Piecewise Linear Material Input** window dialog box to specify a property's values at frequencies below and above the frequency range. Based on these values, HFSS automatically creates a linear dataset that specifies the property's values at the desired frequencies during solution generation. This dataset can be modified with additional points if desired.

Debye Materials

If the material is a lossy dielectric with a lower frequency near DC, use the **Loss Model Material Input** dialog box to specify the material's conductivity at DC or, if you prefer, its loss tangent value at the lower frequency. HFSS also enables you to specify the lossy dielectric material's high frequency/optical permittivity.

In materials commonly encountered in microwave applications ion and dipole polarization dominate. These polarization types can be described by Debye's relaxation polarization model:

$$\varepsilon_{rcomplex} = \varepsilon_{roptical} + \frac{(\varepsilon_{rstatic} - \varepsilon_{roptical})}{1 + j\omega\tau}$$

where

τ = the relaxation time.

$\varepsilon_{rstatic}$ = the static permittivity.

$\varepsilon_{roptical}$ = the high frequency/optical permittivity.

HFSS uses the values you specify in the **Loss Model Material Input** dialog box in Debye's equation above to determine the relative permittivity at any frequency.

Debye's model is valid for most microwave applications. If the frequency exceeds the limit of Debye's model, other models that take atomic and electron polarization into account are available.

Frequency-dependent materials are appropriate for problems solved using a discrete or interpolating sweep. In a fast sweep, frequency dependent material properties are evaluated at the center frequency which may result in less accurate results at the sweep extremes, depending on the sweep range and frequency dependence of material property.

In the simulation of high speed connectors or PCB boards, it is important to take the losses into account. Especially, in the case of transient analysis, where the improper specification of the frequency dependence of the materials would lead to unphysical results. This section discusses the loss mechanism of dielectric materials. Magnetic losses could also be taken into account, but materials commonly encountered in these applications are non-magnetic.

A lossy dielectric material is characterized by two measured values at a certain frequency: dielectric constant ε_r and loss tangent $tg\delta$. There are two problems with the specification of the frequency dependence of dielectric materials:

- The frequency range, in which the solution has to be calculated, is much wider than the range where measured material data are available. It is very common to have just two measured points. A low frequency point, which can be taken as DC value and a higher frequency point, which serves to describe the behavior of the material at high frequency. The question arises: What is an appropriate model across the measured range as well as outside the measured range?
- A measurement always suffers from errors. If the measured data are not consistent, it could cause unphysical effects during the time domain analysis. A question arises again: how to check whether the measured data are consistent, and how to adjust them if they are not.

To answer the questions raised we need to discusses the polarization loss mechanism of dielectric materials.

Figure 1 shows the different polarization mechanisms occur in the frequency domain. It can be seen from the figure that up to the microwave region, ion and dipole polarization dominate. These polarization types can be described by Debye's relaxation polarization model:

$$\varepsilon_{rcomplex} = \varepsilon_{roptical} + \frac{(\varepsilon_{rstatic} - \varepsilon_{roptical})}{1 + j\omega\tau} \quad (1)$$

where

τ is the relaxation time,$\varepsilon_{rstatic}$,and $\varepsilon_{roptical}$ are the static and high frequency permittivity, respectively. Debye's model is valid for the most of microwave applications. If the frequency exceeds the limit of Debye's model, there exists other models which take the atomic and electron polarization into account [1].

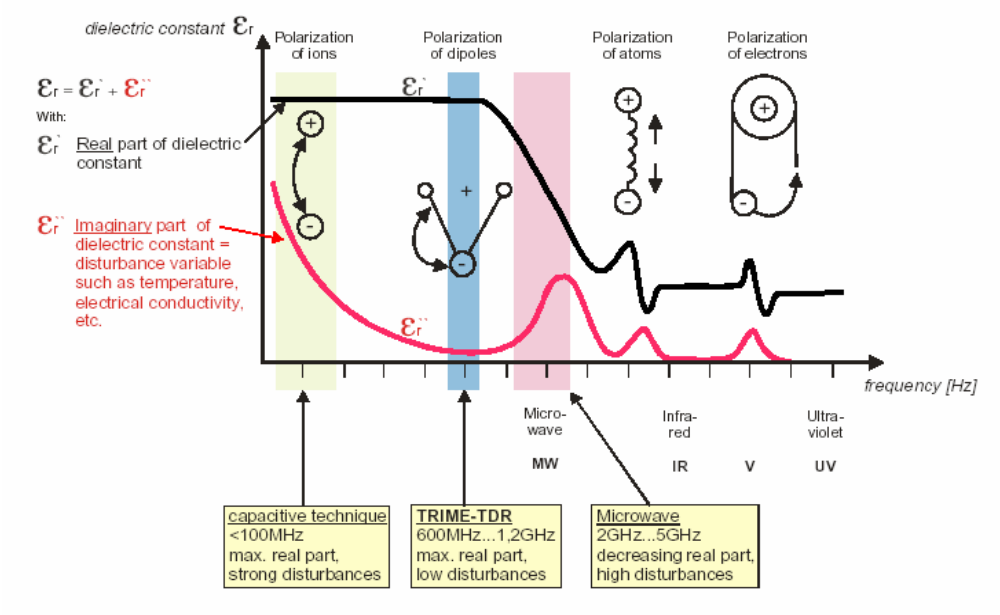


Figure 1 : Polarization mechanisms

The question is now how to fit Debye's model to the actual material characteristics. In order to do this, let us express the real part of the dielectric constant and the conductivity from Eq. (1):

$$\varepsilon_r = \varepsilon_{roptical} + \frac{(\varepsilon_{rstatic} - \varepsilon_{roptical})}{1 + (\omega\tau)^2} \quad (2)$$

$$\sigma = \sigma_o + \frac{\omega^2 \varepsilon_o \tau (\varepsilon_{rstatic} - \varepsilon_{roptical})}{1 + (\omega\tau)^2} \quad (3)$$

where

σ_o is the DC conductivity.

Eqs (2) and (3) have 4 parameters: σ_o, $\varepsilon_{rstatic}$, $\varepsilon_{roptical}$ and τ.

Usually, measured data are available at two frequencies for a regular material. The measured data are the dielectric constant and the loss tangent. One set of the measured data is at low frequency ($f_1 \approx 1MHz, \varepsilon_{r1}, \sigma_1$) and the other set is at higher frequency ($f_2 \approx 1 \sim 2\,GHz, \varepsilon_{r2}, \sigma_2$).

The static or dc values can be considered as the low frequency measured data. So,

$$\varepsilon_{rstatic} = \varepsilon_{r1} \tag{4}$$

and

$$\sigma_o = \omega_1 \varepsilon_o \varepsilon_{r1} \tan\delta_1 \tag{5}$$

The critical need is to predict the high frequency behavior of the material.

If we know $\varepsilon_{roptical}$ from measurement, the high frequency behavior of Debye's model is set. We just need to calculate the relaxation time as:

$$\tau = \frac{b - \sqrt{b^2 - 4}}{2\omega_2} \tag{6}$$

where

$$b = \frac{\omega_2 \varepsilon_o (\varepsilon_{rstatic} - \varepsilon_{roptical})}{\sigma_2} \tag{7}$$

and

$$\sigma_2 = \omega_2 \varepsilon_o \varepsilon_{r2} \tan\delta_2 - \sigma_o \tag{8}$$

Knowing τ, Eqs. (2) and (3) prescribe the material characteristic in the whole frequency region.

If we do not know $\varepsilon_{roptical}$, it can be calculated by simultaneously solving the following two equations to get $\varepsilon_{roptical}$ and τ:

$$\varepsilon_{r2} = \varepsilon_{roptical} + \frac{(\varepsilon_{rstatic} - \varepsilon_{roptical})}{1 + (\omega\tau)^2} \tag{9}$$

$$\sigma_2 = \sigma_o + \frac{\omega^2 \varepsilon_o \tau (\varepsilon_{rstatic} - \varepsilon_{roptical})}{1 + (\omega\tau)^2} \tag{10}$$

where ε_{r2} is measured and σ_2 is determined by Eq.(8).

Solving equations (9) and (10), we get:

$$\tau = \frac{\varepsilon_o \left| \varepsilon_{rstatic} - \varepsilon_{r2} \right|}{\left| \sigma_2 - \sigma_o \right|} \tag{11}$$

This method also yields the parameters of the Debye's model, but the accuracy of the method, especially at high frequencies, depends on the accuracy and the consistency of the measurement at frequency f_2.

Typical Debye's model material characteristics can be seen in Fig. 2, where $\varepsilon' = \varepsilon_r$.

$$\varepsilon'' = \frac{\omega\tau(\varepsilon_{rsrtatic} - \varepsilon_{roptical})}{1 + (\omega_2\tau)^2} \tag{12}$$

$$\varepsilon_{roptical} = \varepsilon_{rstatic} - \left| \varepsilon_{rstatic} - \varepsilon_{r2} \right| \left(\frac{1 + (\omega_2\tau)^2}{(\omega_2\tau)^2} \right) \tag{13}$$

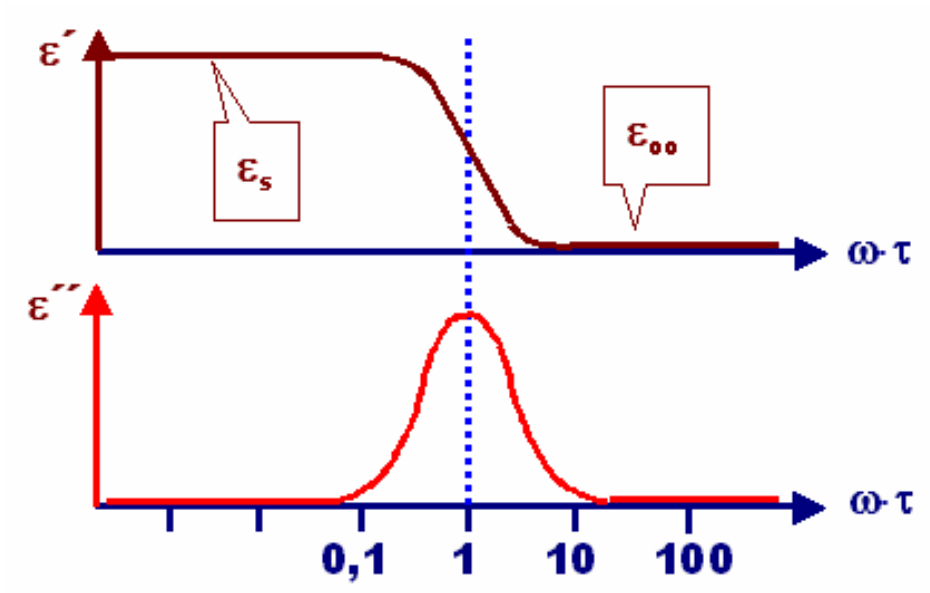


Figure 2 : Material characteristics of a Debye model

The model was verified by using a double strip line filled by FR4 material. A 10 mils section of the line was modeled by HFSS and de-embedded into a 34 inch long line. The structure of HFSS model can be seen in Fig. 3.

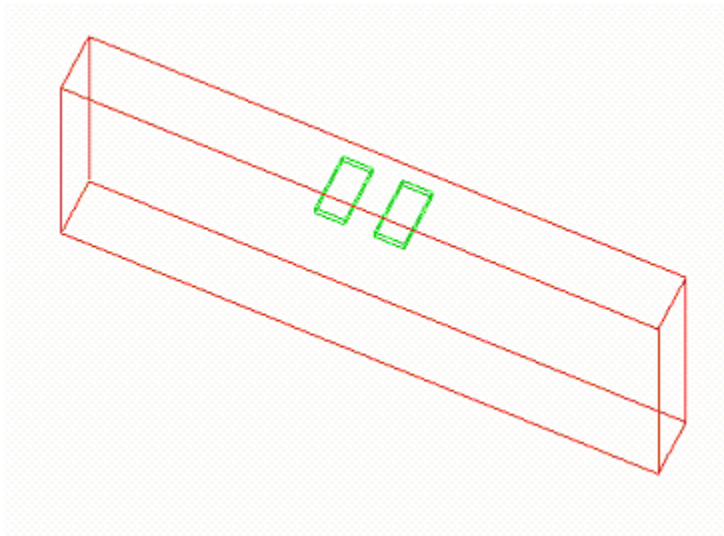


Figure 3 : A 10 mils long structure of a double strip line

The input data panel of HFSS for FR4 can be seen in Fig. 4:

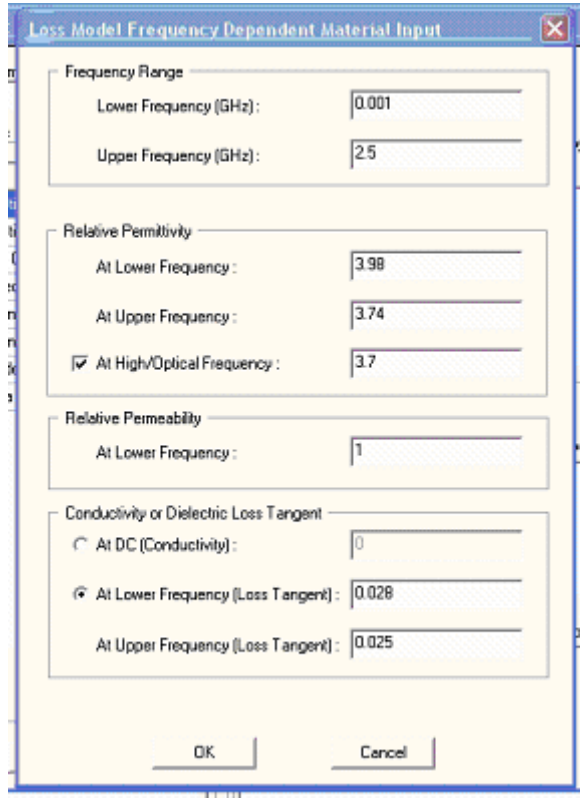


Figure 4 : Input data panel for FR4 material in HFSS

It can be seen from the panel that:

$$\varepsilon_{rstatic} = 3.98 \qquad \varepsilon_{roptical} = 3.7 \qquad \tan\delta_2 = 0.025 \qquad \tan\delta_1 = 0.028$$

The calculated and measured frequency response of S_{12} can be seen in Fig. 5. The agreement between the calculated and measured values is good.

The transient response of the line to an input pulse also has been calculated. The schematic arrangement of the model in Maxwell Spice is shown in Fig. 6. The output signal is plotted in Fig. 7. The green curve is the transient response of an ideal reference line (lossless) of the same length.

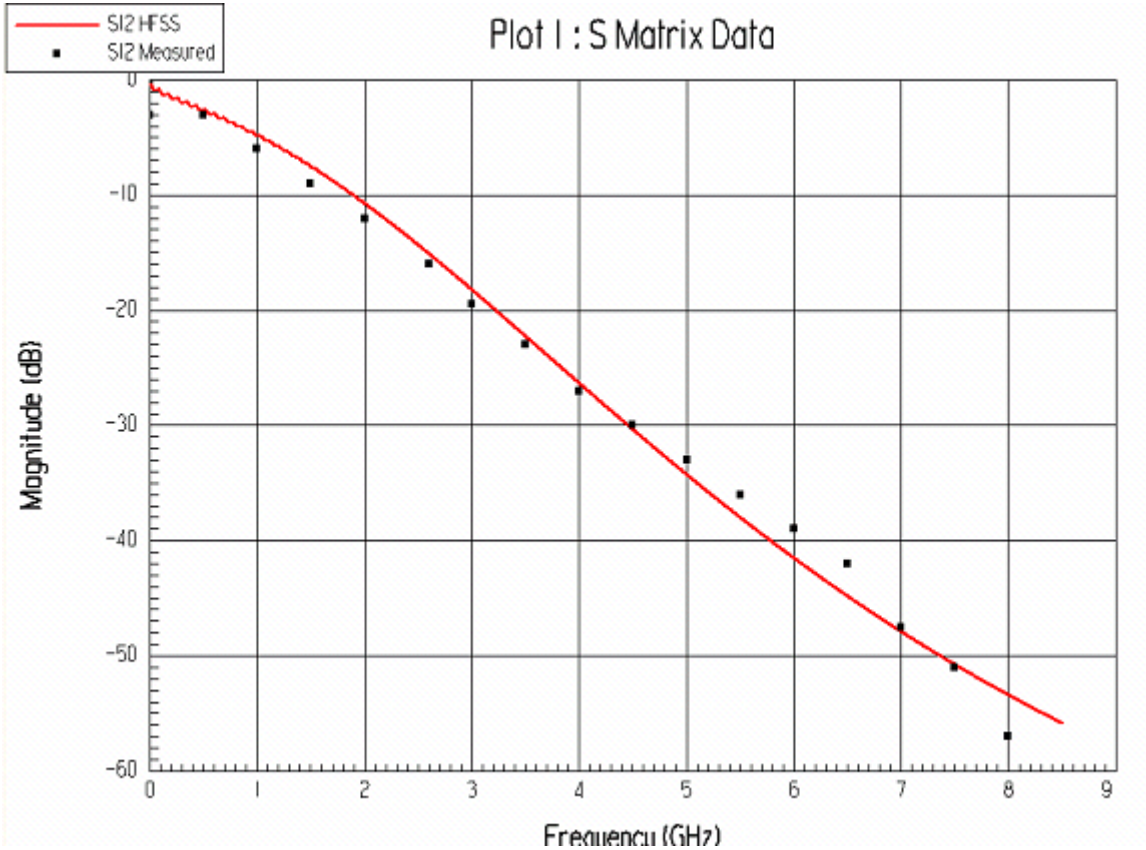


Figure 5 : Calculated and measured frequency response of S_{12}

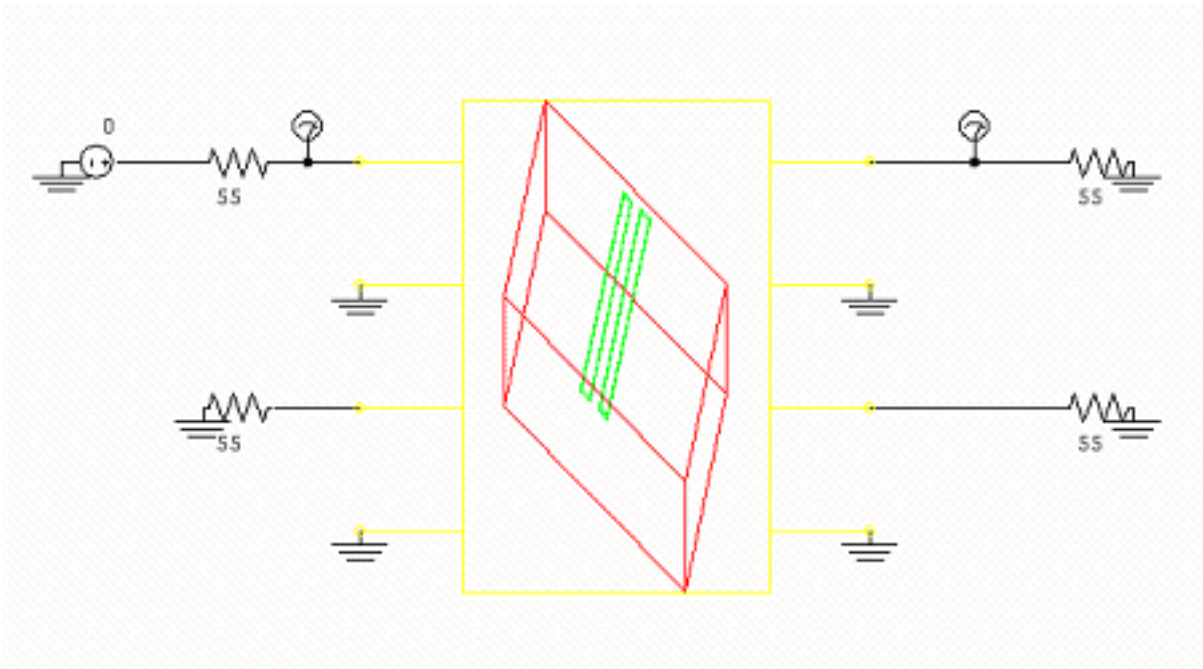


Figure 6 : Schematic arrangement in Maxwell Spice

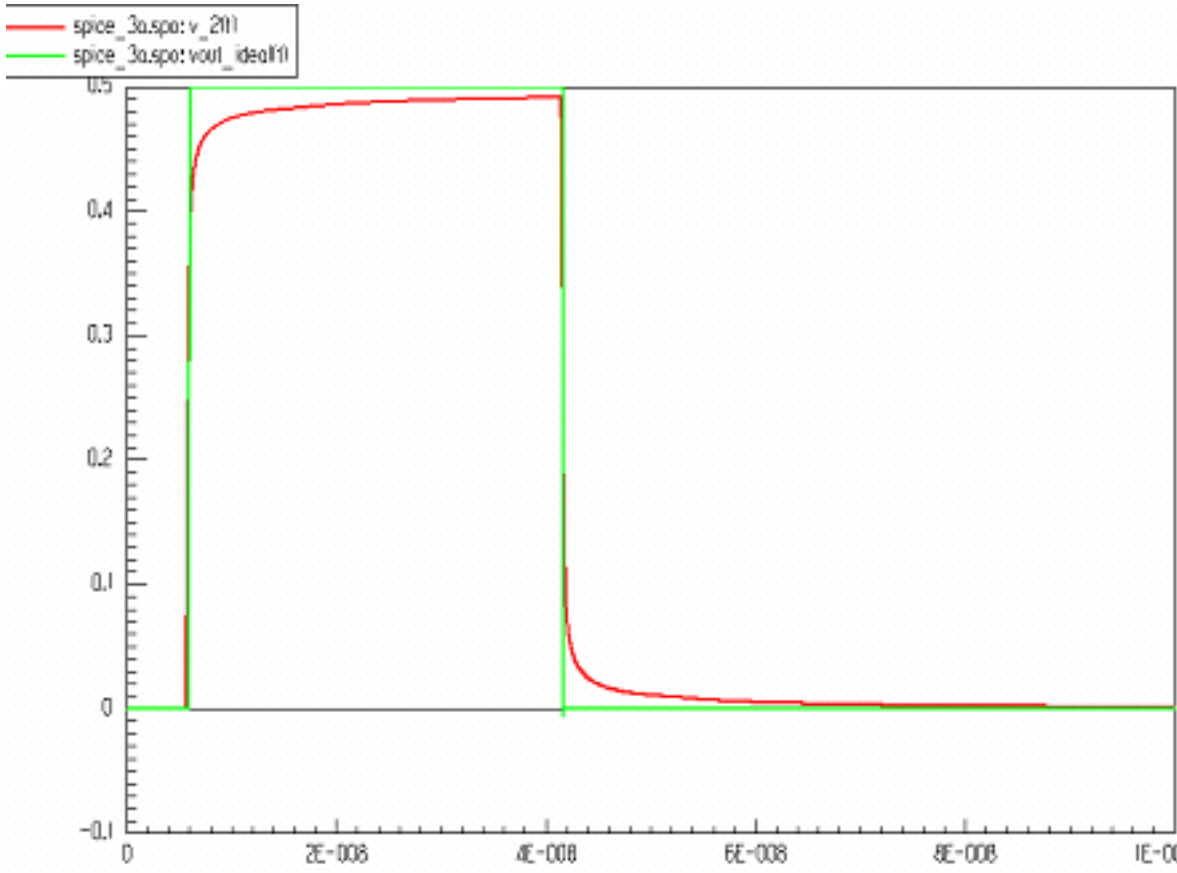


Figure 7 : Transient response by Maxwell Spice

Debye's material loss model predicts the dielectric losses in the whole frequency range within an acceptable accuracy. The model can be set up by using low frequency and higher frequency measured dielectric constants and loss tangents and the optical dielectric constant. If the latter is not available, the model might be less accurate at high frequencies. Using Debye's model, no unphysical phenomena can be observed at the transient response.

References

[1] E.U. Condon and Hugh Odishaw, *Handbook of Physics*, McGraw Hill Book Company, Inc., New York Toronto London, 1958. Pp. 4-113 - 4-119.

[2] G.R. Strobl, *The Physics of Polymers,* Chapter 5, 2nd Ed. Springer, NY, 1977.

Related Topics

Defining Frequency-Dependent Material Properties

Multipole Debye Model

The classical Debye model is based on a single pole or time constant. The Multi-Pole Debye Model attempts to generalize the Debye model by adding multiple poles providing greater flexibility in matching arbitrary measurement data.

A lossy dielectric has a complex relative permittivity denoted as follows:

$$\varepsilon(j\omega)$$

It can be expressed as follows:

$$\varepsilon(j\omega) = \varepsilon_r(j\omega) + j\varepsilon_i(j\omega) = \varepsilon_r(j\omega)[1 - j\delta(j\omega)] = \varepsilon_r(j\omega) - j\varepsilon_r(j\omega)\delta(j\omega)$$

where

$\varepsilon_r(j\omega)$ = real valued permittivity and $\delta(j\omega)$ = loss tangent

If the relative permittivity and loss tangent measured at a finite set of frequency points are known, a rational function model can be fit to this data. This approximating rational function can be expressed in pole-residue form as

$$\tilde{\epsilon}(j\omega) = k_0 + \sum_{i=1}^{N} \frac{k_i}{j\omega - p_i}$$

where

p_i = the i-th pole and

k_i = the corresponding residue

Both are potentially complex valued quantities.

N = number of poles which needs to be determined.

k_0 = a high frequency limit, or "optical" permittivity and is also a fitting parameter.

The fitting problem can be expressed as follows: minimize the sum of the least squares differences:

$$E = \sum_{n=1}^{N_f} |\tilde{\epsilon}(j\omega_n) - \epsilon(j\omega_n)|^2$$

where

N_f = number of frequencies

ω_n = measurement frequencies that you supply.

HFSS will pre-process the measured data to ensure that it obeys the Hilbert transform (Kramers-Kronig) causality conditions. It then determines a set of stable poles that provide a good model to the frequency variation of the measured data. Finally the residues are determined through least-squares fitting.

The resulting model ensures a causal response for the dielectric when used in time domain simulation. This is because the impulse response of the multi-pole Debye model can be expressed analytically in the time domain as follows:

$$E(t) = k_0\delta(t) + \sum_{i=1}^{N} k_i \exp(p_i t)\, u(t)$$

where

$\delta(t)$ =Dirac Delta function

$u(t)$ = the Heaviside step-unit function

The above impulse response is zero for time t < 0 and so, it is casual.

Recall the approximating multi-pole Debye model function:

$$\tilde{\epsilon}(j\omega)$$

After computing this approximating multi-pole Debye model, HFSS splits the model into its real and imaginary parts. The real part corresponds to the approximation for the relative permittivity and the imaginary part divided by the real part gives the approximation for the (negative of the) loss tangent. The resulting (very complicated) expressions are then filled in for the new material model created.

Djordjevic-Sarkar Causal Dielectric Model

In the paper "Wideband Frequency-Domain Characterization of FR-4 and Time-Domain Causality" (A. R. Djordjevic, R. D. Biljic, V. D. Likar-Smiljani, and T. K. Sarkar, *IEEE Trans. on Electromagnetic Compatibility*, Nov. 2001, p. 662), the following simple analytic model was proposed for the frequency dependence of the complex relative permittivity in FR-4 dielectric materials:

$$\varepsilon(\omega) = \varepsilon'(\omega) + j\varepsilon''(\omega) = \varepsilon_\infty + \frac{\Delta\varepsilon}{\ln(\omega_B/\omega_A)} \ln\left(\frac{\omega_B + j\omega}{\omega_A + j\omega}\right) + \frac{\sigma}{j\omega\varepsilon_0} \qquad (1)$$

where

ε_∞ = high frequency ("optical") relative permittivity

$\Delta\varepsilon = \varepsilon_{DC} - \varepsilon_\infty$ = difference between the DC and optical permittivity

σ_{DC} = DC conductivity, typically very close to 0 in FR-4

ε_0 = permittivity of free space $(8.8542\times10^{-12}\text{F/m})$

$\omega = 2\pi f$ = angular frequency

ω_A = lower (angular) transition frequency; below this frequency the permittivity quickly approaches its DC value

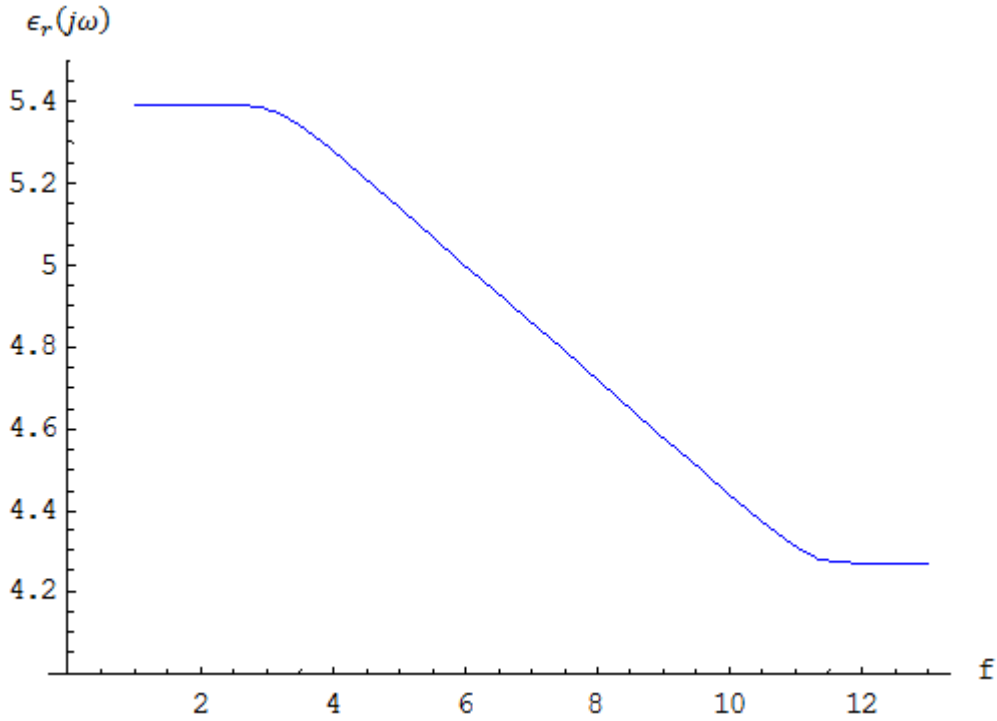
ω_B = upper (angular) transition frequency; above this frequency the permittivity quickly approaches its high-frequency (optical) limit

This model is an analytic function and hence, causal in the time domain. It is very simple and only five parameters define it.

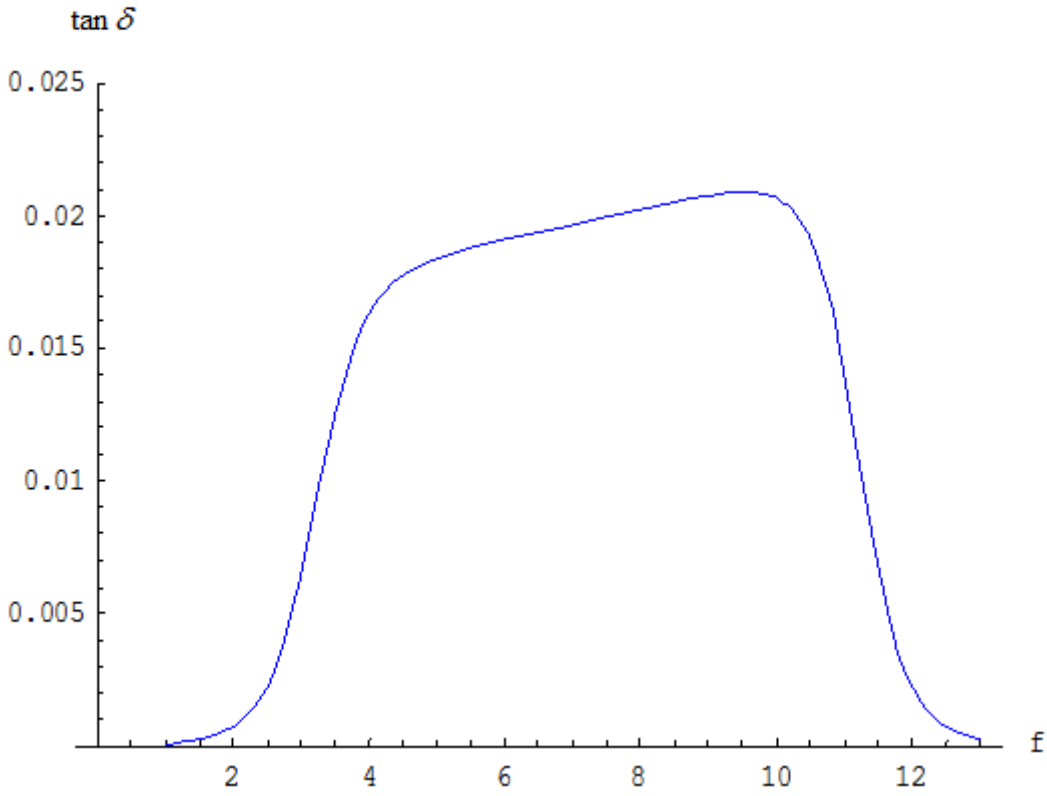
The loss tangent is defined as follows:

$$\tan\delta = -\frac{\varepsilon''(\omega)}{\varepsilon'(\omega)}$$

The behavior of the real part of permittivity versus frequency using the Djordjevic Sarkar model is given by way of the following plot:



The loss tangent predicted by the model (for the case $\sigma = 0$) is shown below.



Notice that both curves have two transition frequencies, the first around 10^3 Hz and the second around 10^{11} Hz.

These correspond to the values $\omega_A = 10^4 \text{rad/s}$ and $\omega_B = 10^{12} \text{rad/s}$ used in the model.

The lower transition frequency is evident in the experimental data for FR-4; the upper transition frequency is best thought of as a fitting parameter. Measured data for FR-4 up to 10 GHz does not show the upper transition frequency. Typically, the upper transition frequency is placed at very high frequency, beyond the limits of practical interest.

Another major feature of the model is that the loss tangent is approximately constant between the two transition frequencies. This constant loss tangent behavior is also observed experimentally in FR-4, except at low frequencies, where it decreases. At very low frequency the loss tangent starts to rise again due to nonzero DC conductivity.

Djordjevic-Sarkar Model Parameter Calculation

The model parameters can be calculated if the following information about the material is known:

- DC conductivity, σ_{DC} (optional; default value is 0)

- DC permittivity, ε_{DC} (optional; you have to choose the default. See below)
- A triplet of real numbers, consisting of
  - the measurement frequency ω_1
  - the real permittivity ε_1 at this frequency
  - the loss tangent tan δ_1 at this frequency

From the model equation, observe that

$$\begin{aligned}\varepsilon(\omega_1) &= \varepsilon_\infty + \frac{\Delta\varepsilon}{\ln(\omega_B/\omega_A)}\ln\left(\frac{\omega_B + j\omega_1}{\omega_A + j\omega_1}\right) + \frac{\sigma_{DC}}{j\omega_1\varepsilon_0} \\ &\cong \varepsilon_\infty + \frac{\Delta\varepsilon}{\ln(\omega_B/\omega_A)}\ln\left(\frac{\omega_B + j\omega_1}{j\omega_1}\right) + \frac{\sigma_{DC}}{j\omega_1\varepsilon_0} \\ &= \varepsilon_\infty + \frac{\Delta\varepsilon}{\ln(\omega_B/\omega_A)}\ln\left(\frac{\sqrt{\omega_B^2+\omega_1^2}}{\omega_1}\right) - j\left[\frac{\Delta\varepsilon}{\ln(\omega_B/\omega_A)}\cdot\tan^{-1}\left(\frac{\omega_B}{\omega_1}\right) + \frac{\sigma_{DC}}{\omega_1\varepsilon_0}\right]\end{aligned} \tag{3}$$

We know the imaginary part of $\varepsilon(\omega_1) = -\varepsilon_1 . \tan\delta_1$. Therefore we get,

$$\varepsilon_1 \tan\delta_1 = \frac{\Delta\varepsilon}{\ln(\omega_B/\omega_A)}\cdot\tan^{-1}\left(\frac{\omega_B}{\omega_1}\right) + \frac{\sigma_{DC}}{\omega_1\varepsilon_0} \tag{4}$$

and so

$$K = \frac{\Delta\varepsilon}{\ln(\omega_B/\omega_A)} = \frac{\varepsilon_1\tan\delta_1 - \frac{\sigma_{DC}}{\omega_1\varepsilon_0}}{\tan^{-1}\left(\frac{\omega_B}{\omega_1}\right)} \tag{5}$$

The following equation for the real part also holds:

$$\varepsilon_1 = \varepsilon_\infty + K\ln\left(\frac{\sqrt{\omega_B^2+\omega_1^2}}{\omega_1}\right) \tag{6}$$

Inserting the above expression allows us to solve for the high-frequency permittivity:

$$\varepsilon_\infty = \varepsilon_1 - K \ln\left(\frac{\sqrt{\omega_B^2 + \omega_1^2}}{\omega_1}\right) \tag{7}$$

If you know the DC value of real permittivity, you can find $\Delta\varepsilon = \varepsilon_{DC} - \varepsilon_\infty$ and use the above expression to calculate the lower transition frequency.

$$\omega_A = \frac{\omega_B}{\exp\left(\frac{\Delta\varepsilon}{K}\right)} \tag{8}$$

Check the computed value of lower transition frequency to make sure that it is much less than the measurement frequency. If not, raise the DC permittivity; this will reduce the value of the lower transition frequency. If you do not know the DC permittivity value, you may choose one yourself to satisfy the condition:

$$\omega_A << \omega_1$$

A simple heuristic is to choose $\Delta\varepsilon = 10 \cdot \tan\delta_1 \cdot \varepsilon_\infty$.

Expressions for Permittivity and Conductivity

The final expressions are as follows:

Relative Permittivity:

$$\varepsilon(f) = \varepsilon_\infty + \frac{K}{2} \ln\left(\frac{f_B^2 + f^2}{f_A^2 + f^2}\right) \tag{9}$$

Conductivity:

$$\sigma(f) = \sigma_{DC} + 2\pi f \varepsilon_0 K \cdot \left[\tan^{-1}\left(\frac{f}{f_A}\right) - \tan^{-1}\left(\frac{f}{f_B}\right)\right] \tag{10}$$

Here $f_A = \omega_A / 2\pi$ and $f_B = \omega_B / 2\pi$ is the frequency in Hz.

In the above expressions:

$$f_B = 10^{12} / 2\pi \tag{11}$$

$$K = \frac{\varepsilon_1 \tan \delta_1 - \frac{\sigma_{DC}}{\omega_1 \varepsilon_0}}{\tan^{-1}\left(\frac{\omega_B}{\omega_1}\right)} \tag{12}$$

$$\varepsilon_\infty = \varepsilon_1 - \frac{1}{2} K \ln\left[\left(\frac{\omega_B}{\omega_1}\right)^2 + 1\right] \tag{13}$$

$\Delta\varepsilon = \varepsilon_\infty - \varepsilon_{DC}$ if the DC permittivity is known; else $\Delta\varepsilon = 10 \cdot \tan \delta_1 \cdot \varepsilon_\infty$

$$f_A = \frac{f_B}{\exp\left(\frac{\Delta\varepsilon}{K}\right)} \tag{14}$$

Conductivity and Permittivity Limits

The slope parameter K must be > 0. Therefore we must have

$$\varepsilon_1 \tan \delta_1 - \frac{\sigma_{DC}}{\omega_1 \varepsilon_0} > 0 \tag{15}$$

This can occur if you enter a DC conductivity higher than the effective conductivity at the measurement frequency. This is impossible in the Djordjevic model, because its conductivity increases monotonically with frequency. Another way for it to happen is for the measurement frequency to be too low (Hz instead of GHz, for example) or for the permittivity value to be unreasonably low.

For the model to be passive, we must have $\varepsilon_\infty \geq 1$

Therefore from (5) and (7) the following is true.

$$\varepsilon_\infty = \varepsilon_1 - \frac{\varepsilon_1 \tan\delta_1 - \frac{\sigma_{DC}}{\omega_1 \varepsilon_0}}{2\tan^{-1}\left(\frac{\omega_B}{\omega_1}\right)} \ln\left[\left(\frac{\omega_B}{\omega_1}\right)^2 + 1\right] \geq 1 \tag{16}$$

Define the quantity

$$\tan\delta_{\max} = \frac{2\tan^{-1}\left(\frac{\omega_B}{\omega_1}\right)}{\ln\left[\left(\frac{\omega_B}{\omega_1}\right)^2 + 1\right]} \tag{17}$$

Using this in 16 simplifies the expression as follows.

$$\varepsilon_1 \cdot \tan\delta_{\max} - \left(\varepsilon_1 \tan\delta_1 - \frac{\sigma_{DC}}{\omega_1 \varepsilon_0}\right) \geq \tan\delta_{\max} \tag{18}$$

Then, the condition for passivity can be written as follows:

$$\varepsilon_1 \cdot \left(\tan\delta_{\max} - \tan\delta_1\right) \geq \tan\delta_{\max} - \frac{\sigma_{DC}}{\omega_1 \varepsilon_0} \tag{19}$$

For this to hold, with $\varepsilon > 0$, the following two conditions are necessary

$$\tan\delta_{\max} - \tan\delta_1 > 0 \quad \text{or} \quad \tan\delta_1 < \tan\delta_{\max} \tag{20}$$

$$\varepsilon_1 \geq \varepsilon_{1,\min} = \frac{\tan\delta_{\max} - \frac{\sigma_{DC}}{\omega_1 \varepsilon_0}}{\tan\delta_{\max} - \tan\delta_1} \tag{21}$$

We see from the expression for tan δ_{max} given in (17) that the maximum loss tangent is a function of only the upper corner frequency and the measurement frequency.

Reducing the upper corner frequency will generally increase the value of tan δ_{max} at least as long as the following condition is realized.

ω_B is much greater than ω_1

But reducing the upper corner frequency is not very attractive, because it has not been observed in experimental data. Therefore it is an upper bound on the loss tangents that the Djordjevic-Sarkar model can handle. Equation (21) indicates that if the permittivity is too low, then the optical permittivity will go below 1. If this condition is violated, two possible fixes are to raise the value of ε or to reduce the measurement frequency. But by the same argument as before, reducing the upper corner frequency is not attractive and it is the minimum value of the permittivity. If you reduce the value of the measured loss tangent; this makes the denominator in (21) larger and lowers the value of $\varepsilon_{1,min}$.

Force per Unit Area in HFSS

This is mainly for the purpose of mapping surface force density in HFSS to Work Bench Mechanical. We will also provide it for post processing within HFSS. This feature should work regardless of HFSS solution types as long as surface forces exist. Surface forces may exist when one side is conductor but the other is not, or finite conductivity and layered impedance boundary.

The task is to calculate the force per unit area ($\mathbf{f}_s$) distribution on the surface or on a section of the surface of a solid body in HFSS. We presume the solution is known, so both the electric and the magnetic field is available in any point.

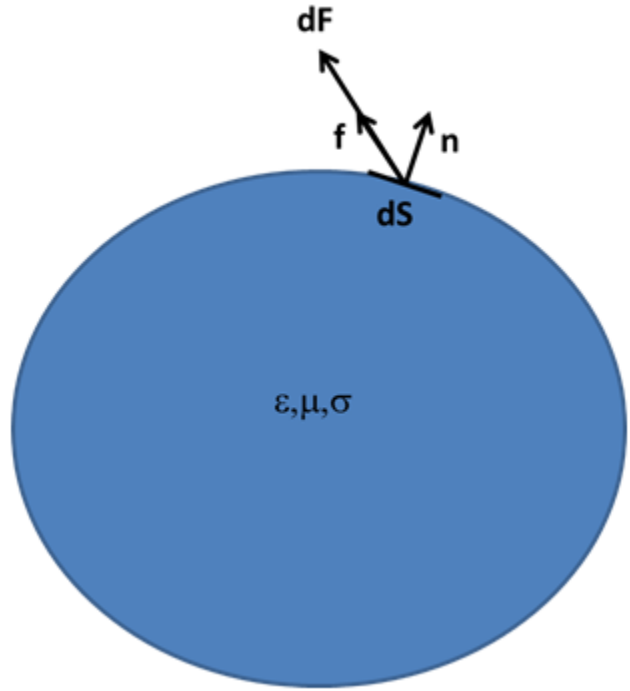


Fig. 1: Force and force per unit area on a solid body

Fig. 1 shows a solid body of homogeneous or inhomogeneous material characteristics (ε,μ, σ). **dS** is an elementary section of its surface and **n** is the normal vector of the surface. If **dF** [N] is the force acting on the elementary surface **dS**, then:

$$\mathbf{f}_s = \frac{\mathbf{dF}}{dS} \tag{1}$$

where $\mathbf{f}_s$ is the force per unit area. Knowing the electromagnetic field, **dF** can be calculated as:

$$\mathbf{dF} = \mathbf{T}^{em}\mathbf{n}dS \tag{2}$$

then

$$\mathbf{f}_s = \mathbf{T}^{em}\mathbf{n} \tag{3}$$

where $\mathbf{T}^{em}$is the electromagnetic Maxwell's stress tensor, which consists of the electric and the magnetic stress tensor as:

$$\mathbf{T}^{em} = \mathbf{T}^{e} + \mathbf{T}^{m} \tag{4}$$

(5)

$$\mathbf{T}^{e} = \begin{bmatrix} D_1E_1 - 0.5DE & D_1E_2 & D_1E_3 \\ D_1E_2 & D_2E_2 - 0.5DE & D_2E_3 \\ D_1E_3 & D_2E_3 & D_3E_3 - 0.5DE \end{bmatrix}$$

(6)

$$\mathbf{T}^{e} = \begin{bmatrix} B_1H_1 - 0.5BH & B_1H_2 & B_1H_3 \\ B_1H_2 & B_2H_2 - 0.5BH & B_2H_3 \\ B_1H_3 & B_2H_3 & B_3H_3 - 0.5BH \end{bmatrix}$$

$$DE = D_1E_1 + D_2E_2 + D_3E_3 \quad (7)$$

$$BH = B_1H_1 + B_2H_2 + B_3H_3 \quad (8)$$

Indices 1,2 and 3 mean the first, second and third component. For example, in Cartesian coordinates, the meaning is x, y and z.

Using the notations:

(9)

$$\mathbf{T}^{em} = \begin{bmatrix} T_{11}^{em} & T_{12}^{em} & T_{13}^{em} \\ T_{21}^{em} & T_{22}^{em} & T_{23}^{em} \\ T_{31}^{em} & T_{32}^{em} & T_{33}^{em} \end{bmatrix}; \qquad n = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}$$

The force per unit area from Eq. (3) is:

(10)

$$\mathbf{f}_S = \mathbf{T}^{em}\mathbf{n} = \begin{bmatrix} T_{11}^{em} n_1 & + T_{12}^{em} n_2 & + T_{13}^{em} n_3 \\ T_{21}^{em} n_1 & + T_{22}^{em} n_2 & + T_{23}^{em} n_3 \\ T_{31}^{em} n_1 & + T_{32}^{em} n_2 & + T_{33}^{em} n_3 \end{bmatrix}$$

Note two things:

Stress tensor is defined on the surface of **S**

Force density might have both normal and tangential components on the surface

The filed quantities in formulas (1) - (10) are time dependent or static. We want to have the forces in steady state time harmonic case, too, by using the complex notations as the field quantities are defined in HFSS. It is derived for the magnetic stress tensor. The derivation for the electric stress tensor can be got by replacing B_k and H_k by D_k and E_k, respectively, where index k refers to the k-th component. The real time function of the k-th component of a complex field quantity can be written as:

$$B_k = Re(B_{ck} e^{j\omega t}) = \frac{1}{2}(B_{ck} e^{j\omega t} + B_{ck}^* e^{-j\omega t}) \quad (11)$$

where subscript c denotes complex quantity and superscript '*' means complex conjugate. The stress tensor just contains products of $B_k H_k$ where k =1,2,3 and l =1,2,3. Applying Eq. (10), we get:

(12)

$$B_k H_k = \frac{1}{4}(B_{ck} e^{j\omega t} + B_{ck}^* e^{-j\omega t})(H_{cl} e^{j2\omega t} + H_{cl}^* e^{-j2\omega t}) =$$

$$\frac{1}{4}(B_{ck} H_{cl}^* + B_{ck}^* H_{cl} + B_{ck}^* H_{cl}^* e^{-j2\omega t})$$

Eq. (12) has a constant component and a sinusoidal component of 2ω. If we are interested in the time average force, Eq. (12) should be integrated for one period and divided by the period time:

(13)

$$\langle B_k H_k \rangle = \frac{1}{T}\int_0^T \frac{1}{4}(B_{ck} H_{cl}^* + B_{ck}^* H_{cl} + B_{ck} H_{cl} e^{j2\omega t})dt = \frac{1}{2}\mathrm{Re}(B_{ck} H_{cl}^*)$$

Performing the calculations for k=1,2,3 and l=1,2,3 the elements of the magnetic and electric stress tensor can be calculated as:

$$\langle B_k H_k \rangle = \frac{1}{2}\mathrm{Re}(B_{ck} H_{cl}^*); \qquad \langle D_k E_k \rangle = \frac{1}{2}\mathrm{Re}(D_{ck} E_{cl}^*) \tag{14}$$

The time average surface force density is:

(15)

$$\mathbf{f}_S = \mathbf{T}^{em} = \begin{bmatrix} \langle T_{11}{}^{em} \rangle n_1 + \langle T_{12}{}^{em} \rangle n_2 + \langle T_{13}{}^{em} \rangle n_3 \\ \langle T_{21}{}^{em} \rangle n_1 + \langle T_{22}{}^{em} \rangle n_2 + \langle T_{23}{}^{em} \rangle n_3 \\ \langle T_{31}{}^{em} \rangle n_1 + \langle T_{32}{}^{em} \rangle n_2 + \langle T_{33}{}^{em} \rangle n_3 \end{bmatrix}$$

The volume force acting on a closed surface is:

$$\langle \mathbf{F} \rangle = \oint_S \langle \mathbf{f}_S \rangle dS \tag{16}$$

If a surface is selected to calculate the force, smoothing should not go through this surface.

The calculations just need post processing operations.

The relativistic electromagnetic momentum (radiation pressure), which is associated with the volume sources, has been neglected in the formulas. If it is necessary, it can be taken into account

20

Scripting

For information on how to create, edit and use VBS scripts in HFSS see:

. in html format.

. in pdf format.

The pdf of the scripting guide provides a format and function better suited for printing than the chm. You can print ranges of pages encompassing topics and subtopics as needed.

You can also access help for the VBS scripting commands via the menu bar:

- Click **Help>Scripting Contents**
- Click **Help>Scripting Index**
- Click **Help>Search Scripting**

This chapter contains the following sections:

Related Topics

Desktop Scripting with IronPython

Converting VBScript Function calls to IronPython Syntax

Recording a Script

Once you start to record a script, your subsequent actions are added to the script. Each interface command has one or more associated script commands that are recorded to the script. The script is recorded to a text file in .vbs (VBScript) file format.

1. On the **Tools** menu, click **Record Script**.

   The **Save As** dialog box appears.
2. Use the file browser to locate the folder in which you want to save the script, such as:

   `C:\Ansoft\HFSS2014\Scripts`

   Then double-click the folder's name.
3. Type the name of the script in the **File name** text box, select the script type as Visual Basic Script (.vbs) or IronPython (*.py) and then click **Save**.

   The script is saved in the folder you selected by the file name *filename.vbs*.
4. Perform the steps that you want to record.
5. When you have finished recording the script, click **Stop Script Recording** on the **Tools** menu.

Stopping Script Recording

- On the **Tools** menu, click **Stop Script Recording**.

  HFSS stops recording to the script.

Running a Script

To run a script from HFSS:

1. Click **Tools>Run Script**.

   The **Open** dialog box appears.
2. Use the file browser to locate the folder in which you saved the script, and then double-click the folder's name.
3. Type the name of the script in the **File name** text box, or click its name, and then click **Open**.

   HFSS executes the script.

   To supply script arguments when running from **Tools>Run Script**, use the edit field at the bottom of the file selection dialog. You can access the script arguments using the AnsoftScriptHost.arguments collection from vbscript. This is a standard COM collection.

To run a script from a command line, use:

-runscriptandexit or **-runscript** arguments to the HFSS command line syntax.

You can give **-scriptargs** parameter to the script and specify arguments.

If you run the script from DOS prompt as a .vbs file (that is, you don't launch HFSS, but just launch vbs directly, or use wscript.exe or cscript.exe), the arguments will be in the WSH.arguments collection, not the AnsoftScriptHost.arguments collection. To handle this, you can write this:

```
on error resume next
dim args
```

```
Set args = AnsoftScript.arguments
if(IsEmpty(args)) then
Set args = WSH.arguments
End if
on error goto 0
'At this point, args has the arguments no matter if you are
running
'under windows script host or Ansoft script host
msgbox "Count is " & args.Count
for i = 0 to args.Count - 1
  msgbox args(i)
next
```

Pausing and Resuming a Script

To pause a script during its execution:

- Click **Tools>Pause Script**.

To resume a script after pausing it:

- Click **Tools**>**Resume Script**.

Stopping a Script

Click**Tools>Stop Script**.

HFSS stops executing the script that has been paused.

21

ANSYS Workbench Integration Overview

ANSYS Workbench combines the strength of its core product solvers with the project management tools necessary to manage project workflow. In ANSYS Workbench, analyses are built as *systems*, which can then be combined into a *project*. The project is driven by a schematic workflow that manages the connections between the systems.

From the schematic, you can interact with applications (called workspaces) that are native to ANSYS Workbench and that display within the ANSYS Workbench interface. Native workspaces include: Project Schematic, Engineering Data, and Design Exploration (Parameters and Design Points).

You can also launch applications that are data-integrated with ANSYS Workbench, meaning the application's interface remains separate, but the data from the application communicates with the native ANSYS Workbench data. Thus, data can be passed back and forth between any ANSYS Electromagnetics product on a Workbench Project Schematic and any supported ANSYS or ANSYS Electromagnetics desktop product. Depending on the application, data integration can include basic actions such as saving projects, as well as more complex actions such as the coupling of ANSYS Electromagnetics product variables to Workbench Design Exploration parameters.

Data-integrated applications include the following ANSYS Electromagnetics products: Designer , HFSS, Maxwell/RMxprt, Q3D Extractor, and Simplorer.

Note For detailed information on working with ANSYS Workbench, please refer to the Workbench documentation.

Integrating ANSYS Electromagnetics Products with ANSYS Workbench 15.0

Workbench Data Integration Overview

ANSYS Electromagnetics - ANSYS Multiphysics Coupling

ANSYS Electromagnetics CAD Integration Through Workbench

ANSYS Electromagnetics to ANSYS Geometry Transfer

Integrating ANSYS EM Products with ANSYS Workbench15.0

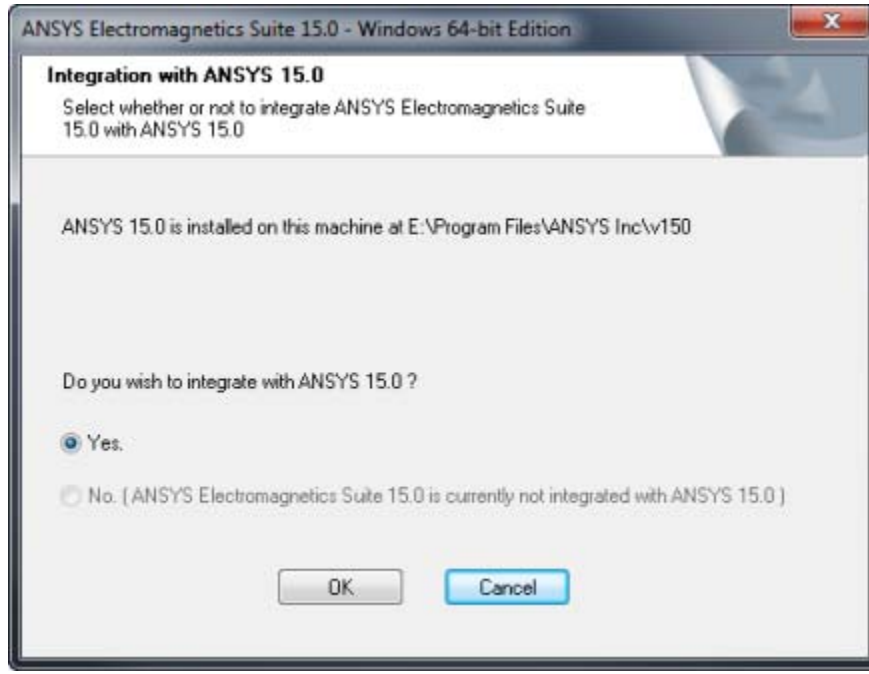
You can integrate ANSYS Electromagnetics products with ANSYS Workbench 15.0 in one of two ways:

- During product installation, or
- After product installation

Integration with ANSYS 15.0 During ANSYS EM Product Installation

You can choose to integrate ANSYS Electromagnetics products with ANSYS 15.0 during installation as follows:

1. If you have not already installed ANSYS 15.0, do so before proceeding. Refer to the ANSYS documentation for instructions on installing the ANSYS 15.0 software.
2. Launch **Autorun.exe** or **Setup.exe** for the ANSYS Electromagnetics product you wish to install.
3. Proceed through the installation dialogs.
4. When the **Integration with ANSYS 15.0** dialog displays, select the **Yes** radio button to have the installer automatically integrate the ANSYS Electromagnetics product with ANSYS 15.0.



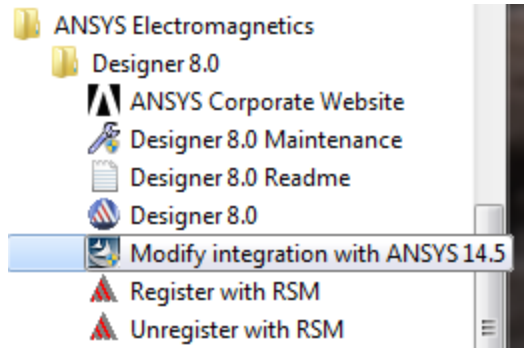
Note If you choose not to integrate the product with ANSYS 15.0 during installation, you can perform this step after installation.

5. Complete the product installation and exit the installer.

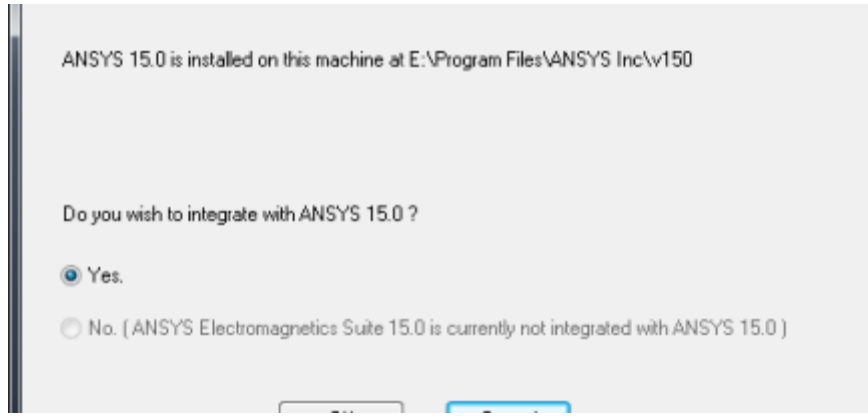
Integration with ANSYS 15.0 after ANSYS EM Product Installation

If you want to integrate an ANSYS Electromagnetics product with ANSYS 15.0 after installation, do the following:

1. Select **All Programs**>**ANSYS Electromagnetics**>*AnsysEM_Product*>**Modify Integration with ANSYS 15.0** from the **Start** menu.

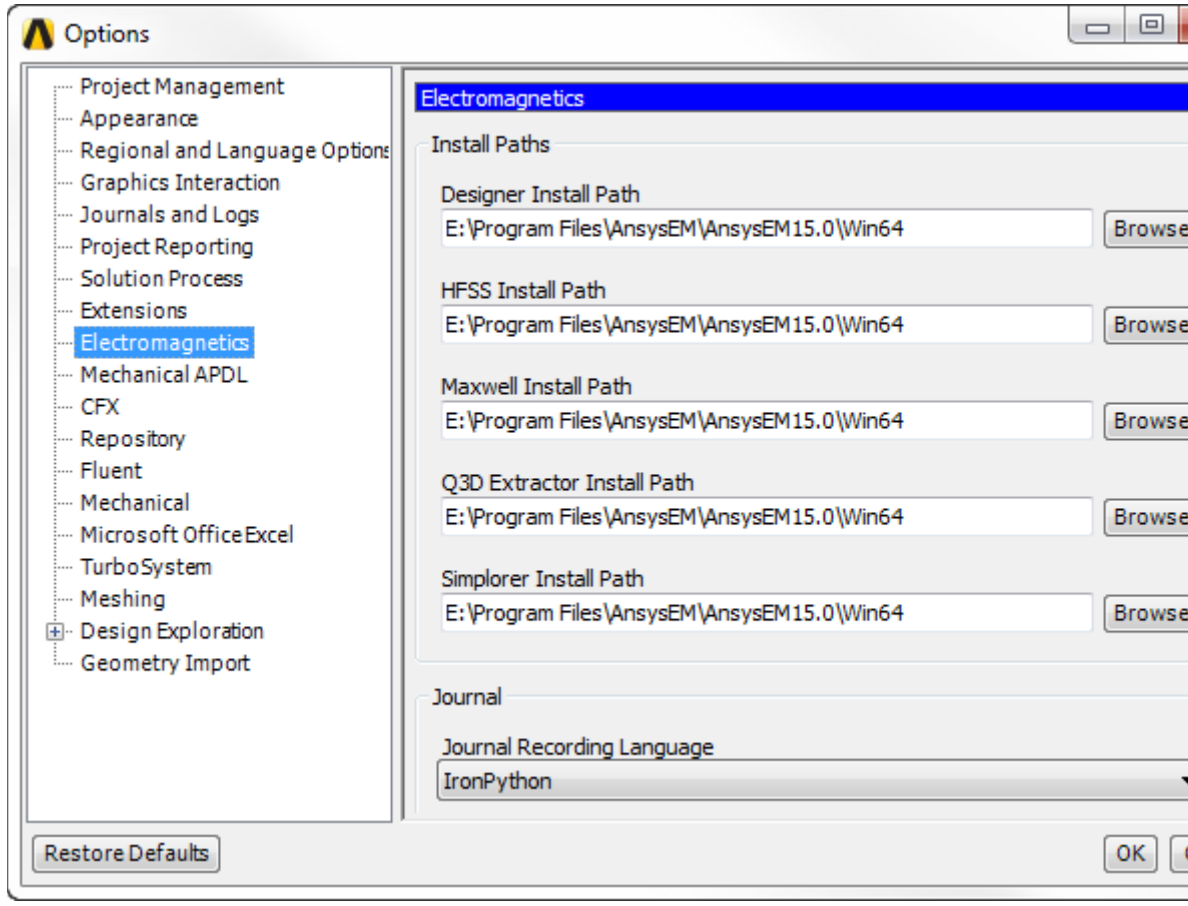


2. In the **Integration with ANSYS 15.0** dialog box, ensure that the **Yes** radio button is selected,



then click OK to complete the integration process.

You can confirm that ANSYS 15.0 is "aware" of the ANSYS Electromagnetics integrated applications via the ANSYS Workbench **Options** dialog, which shows the paths to the integrated AnsysEM applications.

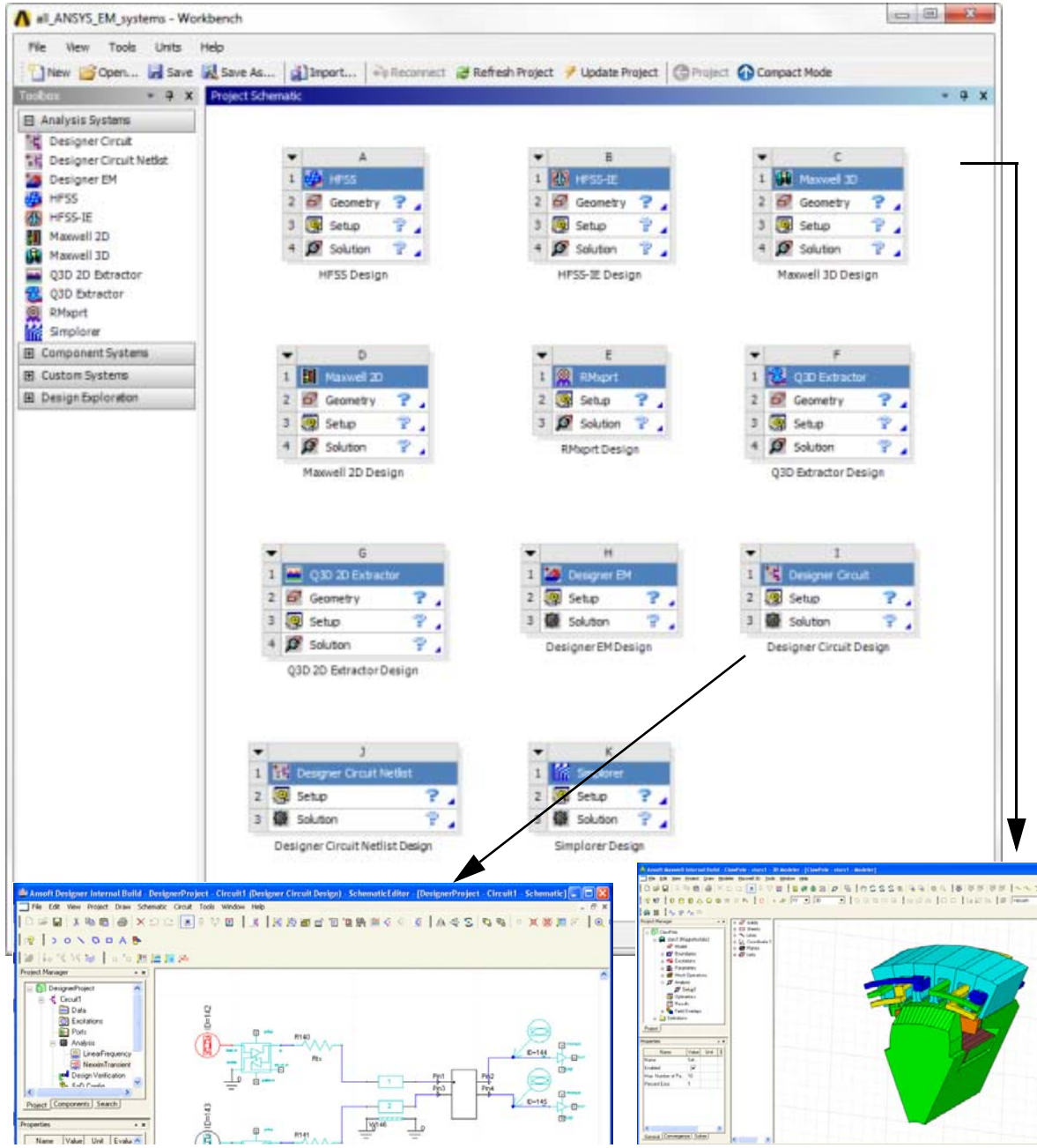


The Journal Recording Language option permits you specify the default journal recording language.

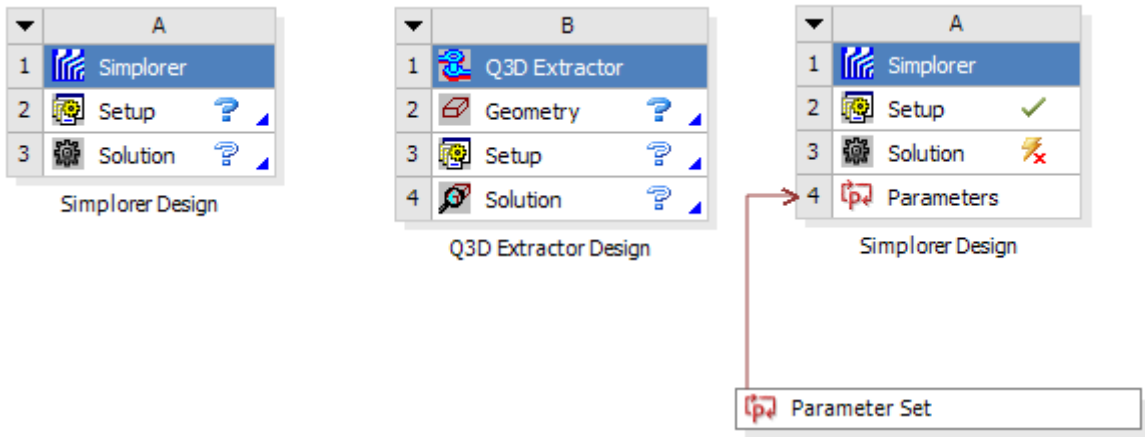
- Default is IronPython. You can change to VB.
- Changes in the recording option applies to any EBU system launched after the change.
- Any open EBU desktops will continue recording in the language that was in effect when they were launched.

Workbench Data Integration Overview

ANSYS EM data-integrated applications can reside on a Workbench **Project Schematic** as shown.



Objects, such as instances of ANSYS Electromagnetics projects, that are placed on a Workbench Project Schematic are referred to as *systems*. ANSYS Electromagnetics circuit/system products: Rmxprt 14.0, Designer 6.1, and Simplorer 10.0, appear on Workbench Project Schematics as systems with two "cells" - **Setup** and **Solution**. ANSYS Electromagnetics field products: HFSS 14.0, Maxwell 14.0, and Q3D Extractor, add an additional **Geometry** cell. If you invoke ANSYS **DesignXplorer** to use variables for refining a design, a **Parameters** cell is added with a link to the associated Workbench **Parameter Set**. Refer to the ANSYS 14.0 Workbench help for details on working with systems, cells, and parameter sets.



ANSYS Electromagnetics desktop products integrate with Workbench commands, services, and DesignXplorer in a similar manner. Here are some of the ways in which ANSYS Electromagnetics products integrate with Workbench:

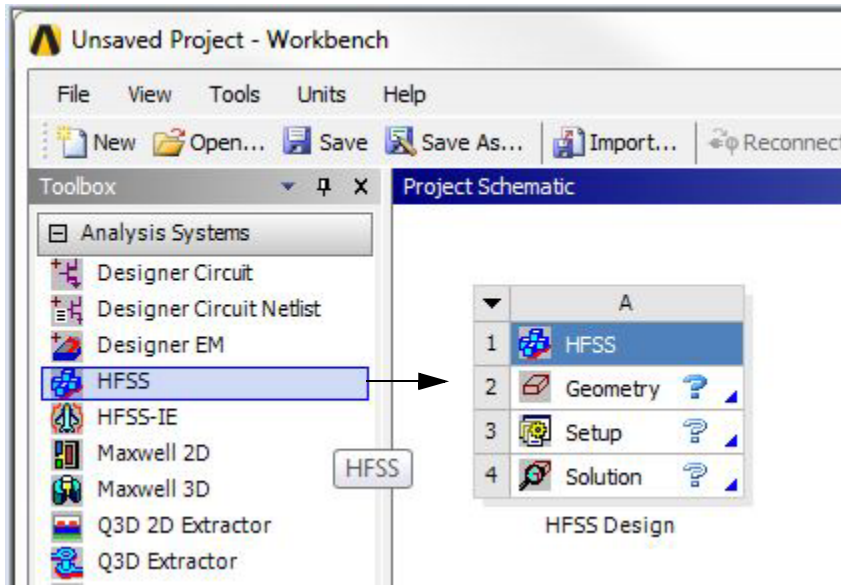
- Adding new analysis systems
- Importing existing desktop projects
- Editing models
- Analyzing models
- Performing parameter studies
- Scripting

In addition to these major features, Workbench also allows you to Archive, Save, Backup, Duplicate, and Delete ANSYS Electromagnetics projects used in a Workbench project. Progress information and messages from integrated ANSYS Electromagnetics projects are also displayed in Workbench.

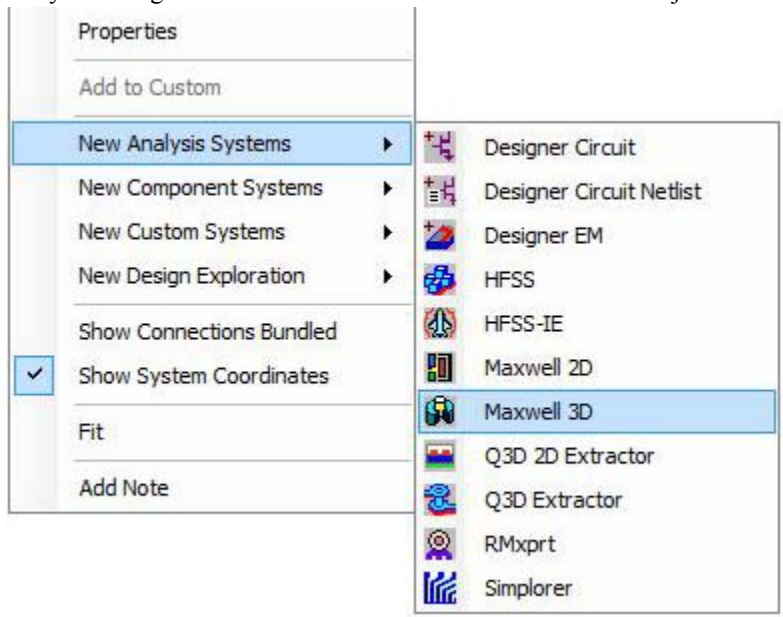
Note Detailed information for how to use ANSYS Workbench 15.0 for these operations can be found in the ANSYS 15.0 documentation and online help.

Adding New Analysis Systems

A new ANSYS Electromagnetics Analysis System can be added to a Workbench Project Schematic either by dragging and dropping it from the Toolbox:



or by selecting it from the context menu in the Workbench Project Schematic window:



Related Topics

Workbench Data Integration Overview

Importing existing desktop projects

Editing models

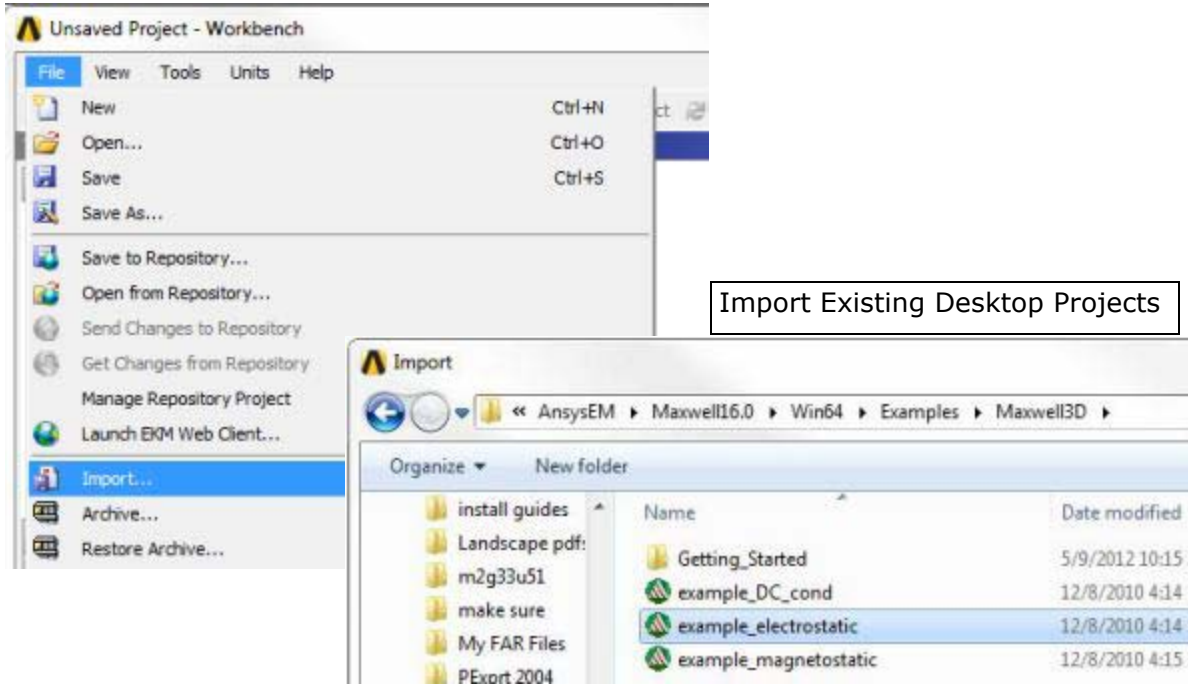
Analyzing models

Performing parameter studies

Scripting

Importing ANSYS EM Projects into ANSYS Workbench

You can import existing ANSYS Electromagnetics desktop projects to a Workbench Project Schematic. When you do, a copy of the ANSYS Electromagnetics project is put into the Workbench Project folder. The original ANSYS Electromagnetics project remains intact.



Related Topics

Workbench Data Integration Overview

Adding new analysis systems

Editing models

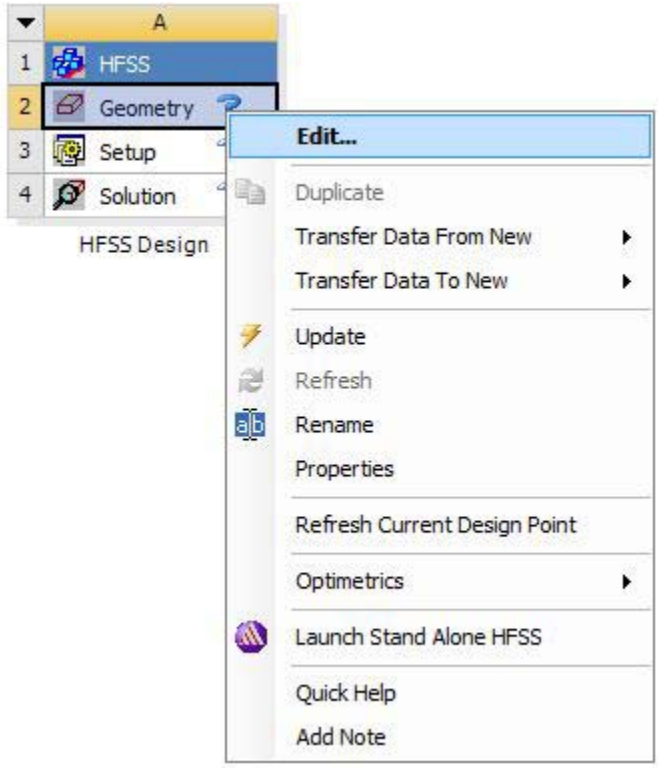
Analyzing models

Performing parameter studies

Scripting

Editing ANSYS EM Models in Workbench

You can edit various properties and parameters (geometry, setup, solution, etc.) of the ANSYS Electromagnetics project either by right-clicking on the project in Workbench and selecting **Edit** on the context menu; or by double-clicking the project. Doing so launches the ANSYS Electromagnetics desktop application and loads the project so that you can setup your project in a familiar desktop environment. Changes made to the ANSYS Electromagnetics project are saved to the project instance in the Workbench project folder.



Related Topics

Workbench Data Integration Overview

Adding new analysis systems

Importing existing desktop projects

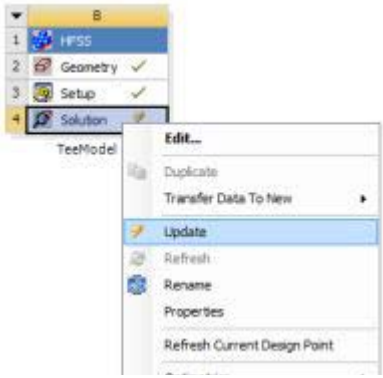
Analyzing models

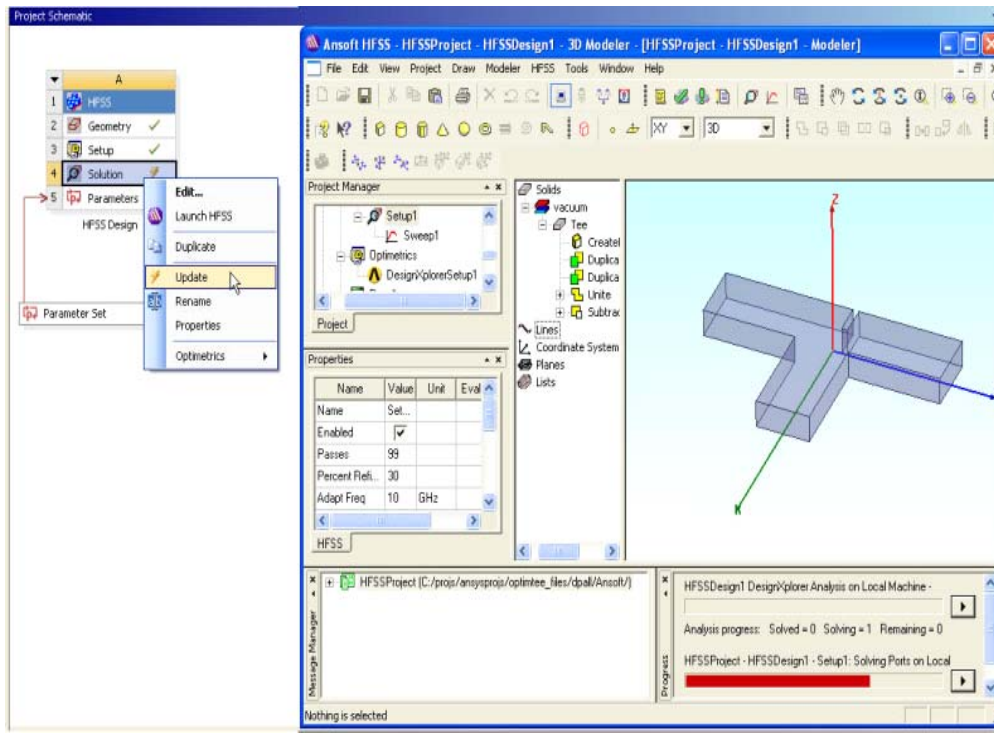
Performing parameter studies

Scripting

Analyzing ANSYS EM Models in Workbench

You can use Workbench's **Update** command to run analyses in the integrated ANSYS Electromagnetics project. Progress information is also shown in Workbench.





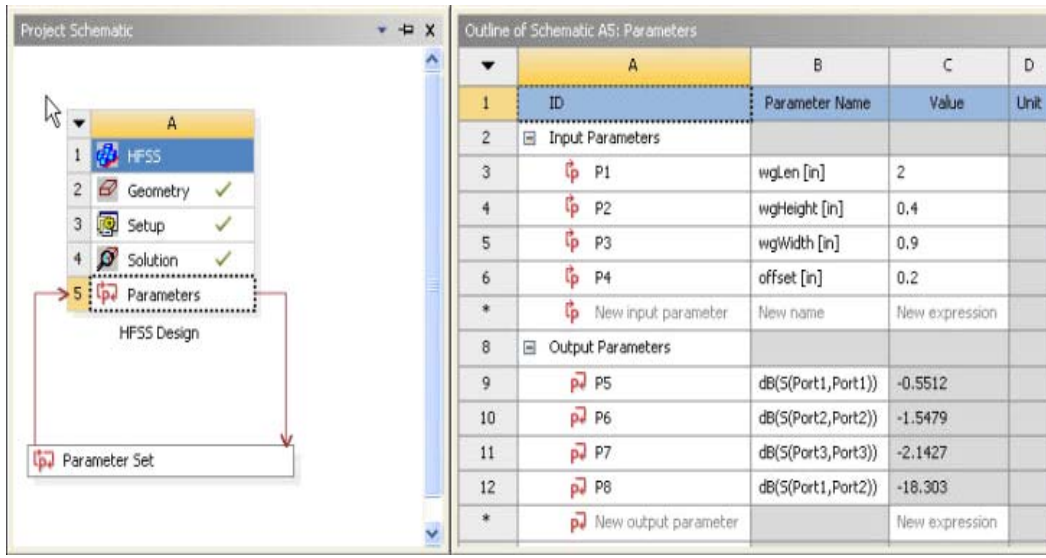
Related Topics

Workbench Data Integration Overview

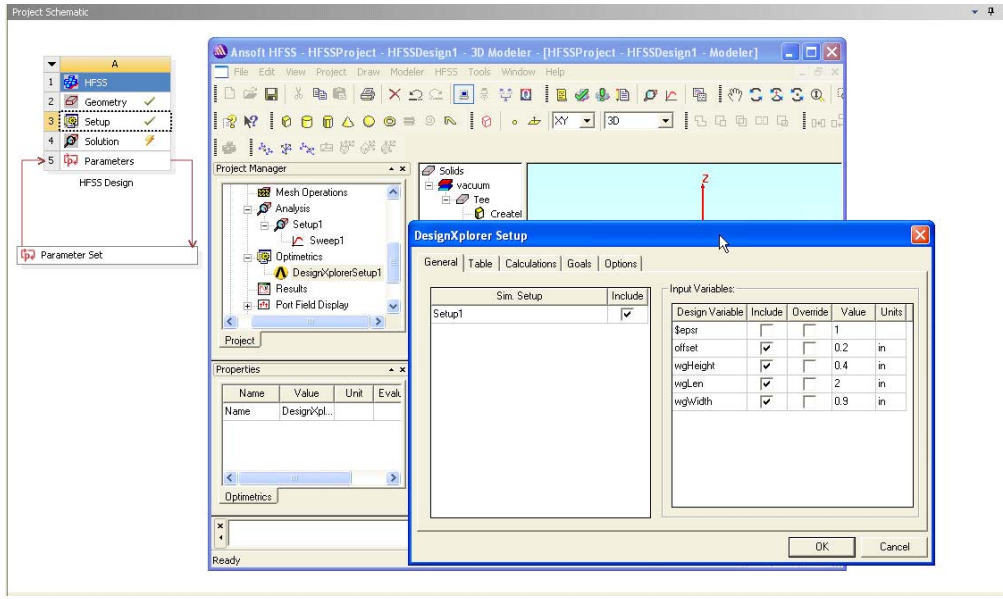
Adding new analysis systems

Performing Parameter Studies in Workbench

Workbench **Parameter Sets** allow you to change parameter values and units of measure, or add new parameters. Parameter data is passed back to the ANSYS Electromagnetics application for updated analyses.



Parameters from the ANSYS Electromagnetics project are exposed to Workbench through the **DesignXplorer** setup. The ANSYS Electromagnetics system's cell status on the Workbench project is updated as changes are made in the ANSYS Electromagnetics application desktop.



Related Topics

Workbench Data Integration Overview

Adding new analysis systems

Importing existing desktop projects

Editing models

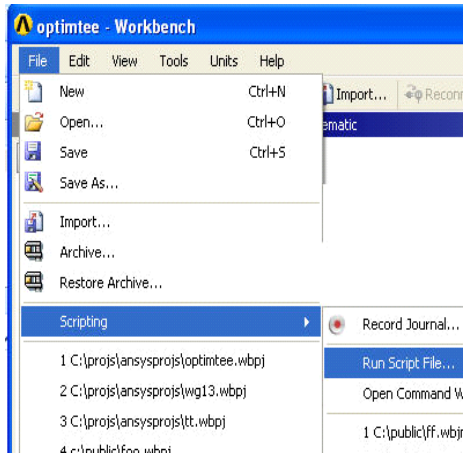
Analyzing models

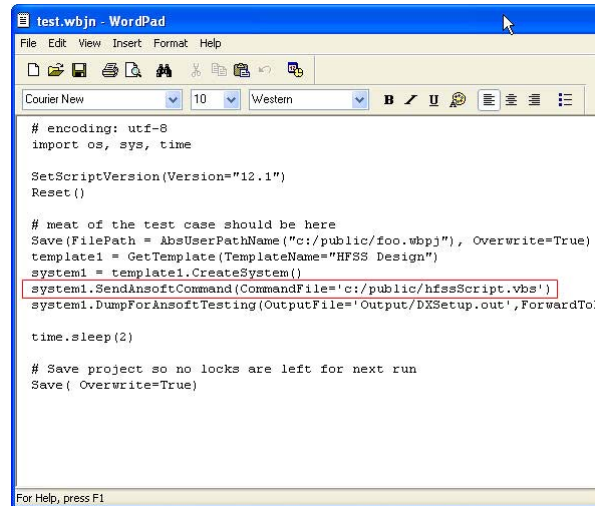
Scripting

Feedback Iterator

Scripting in Workbench

Scripts that include ANSYS Electromagnetics projects can be recorded and played back via Workbench.





Related Topics

Workbench Data Integration Overview

Adding new analysis systems

Importing existing desktop projects

Editing models

Analyzing models

Performing parameter studies

ANSYS EM - ANSYS Multiphysics Coupling

Data integration provides improved multiphysics workflow between ANSYS Electromagnetics designs and ANSYS applications such as Mechanical and Thermal. Coupling is provided through project schematic links. Heat losses and force data are automatically transferred to ANSYS Mechanical - there is no need to export/import "transfer xml files". Edits you make in ANSYS Electromagnetics applications are automatically transferred to the ANSYS application through a Workbench **Refresh** command. Workbench commands also enable easier automation of iterative coupling of thermal feedback. The following sections provide some examples of multiphysics coupling.

Multiphysics Coupling on Workbench with ANSYS Thermal

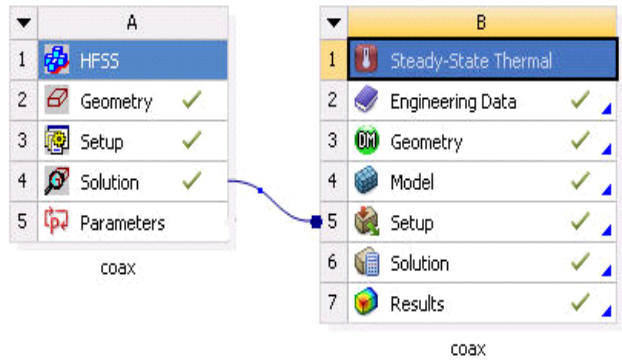
Multiphysics Coupling on Workbench with ANSYS Structural

Multiphysics Coupling between ANSYS EM Field Systems on Workbench

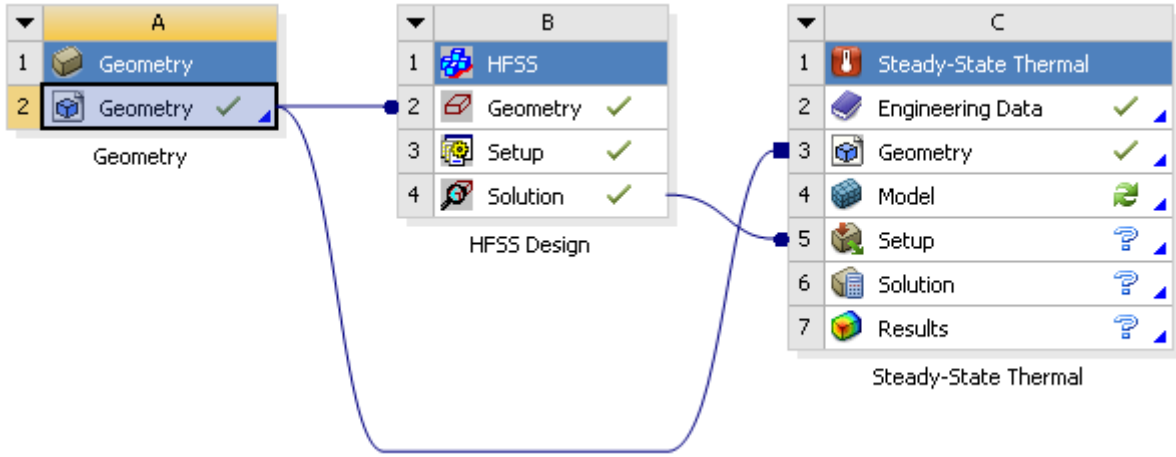
Multiphysics Coupling on Workbench with ANSYS Thermal

Using data integration, HFSS, Maxwell, and Q3D Extractor provide heat losses (heat generation and heat flux) to ANSYS Thermal. You first need to enable feedback as described in Setting the Temperature of Objects.

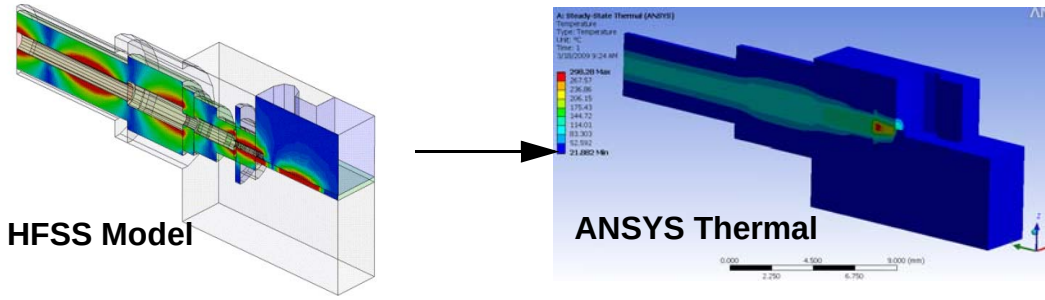
Note how the HFSS design is linked visually to ANSYS Thermal on the Workbench project schematic.



Geometry sharing is possible, provided you activate Beta Options in Workbench. A geometry from the Component Systems in the Toolbox in Workbench Schematic can be shared as shown in the image below.



In this example, HFSS coax model Solution provides heat loss data as a thermal load to the ANSYS Thermal Setup. The resulting analysis shows a thermal "hotspot", providing the user with the information needed to adjust the design's material to fix the problem.



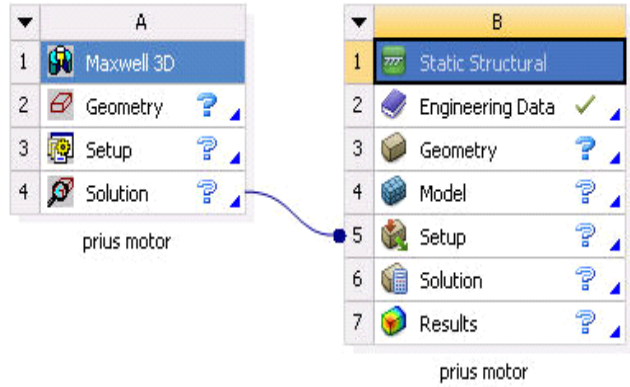
You enable this feature by checking Enable Feedback in the dialog for Setting the Temperature of Objects.

Related Topics

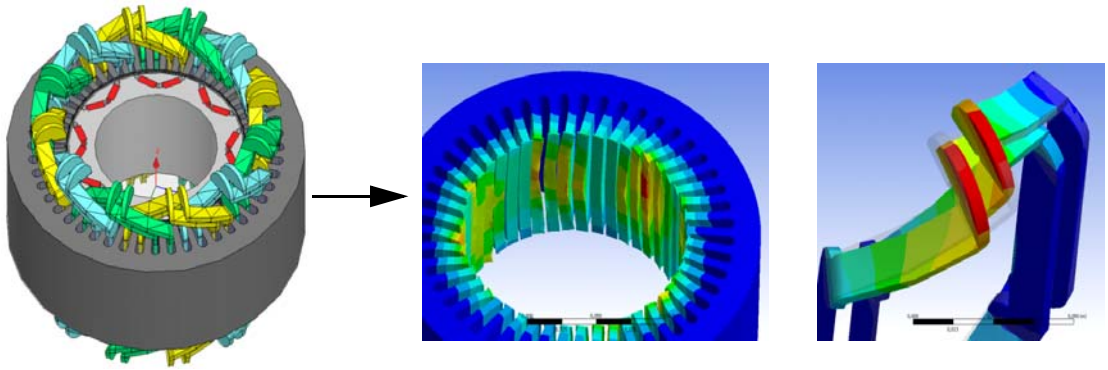
Feedback Iterator

Multiphysics Coupling on Workbench with ANSYS Structural

Using data integration, Maxwell 2D and Maxwell 3Dcan provide forces to ANSYS Structural.



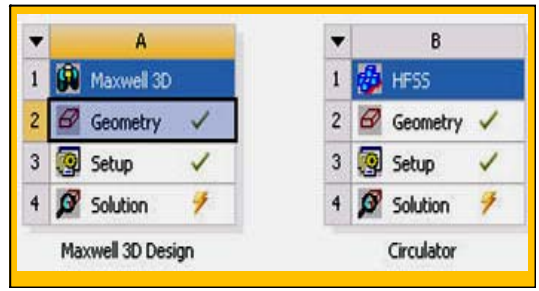
In this example, the Maxwell 3D electromagnetic force density **Solution** is used as the load in ANSYS Structural to determine how these forces deform the motor's stator and coils.

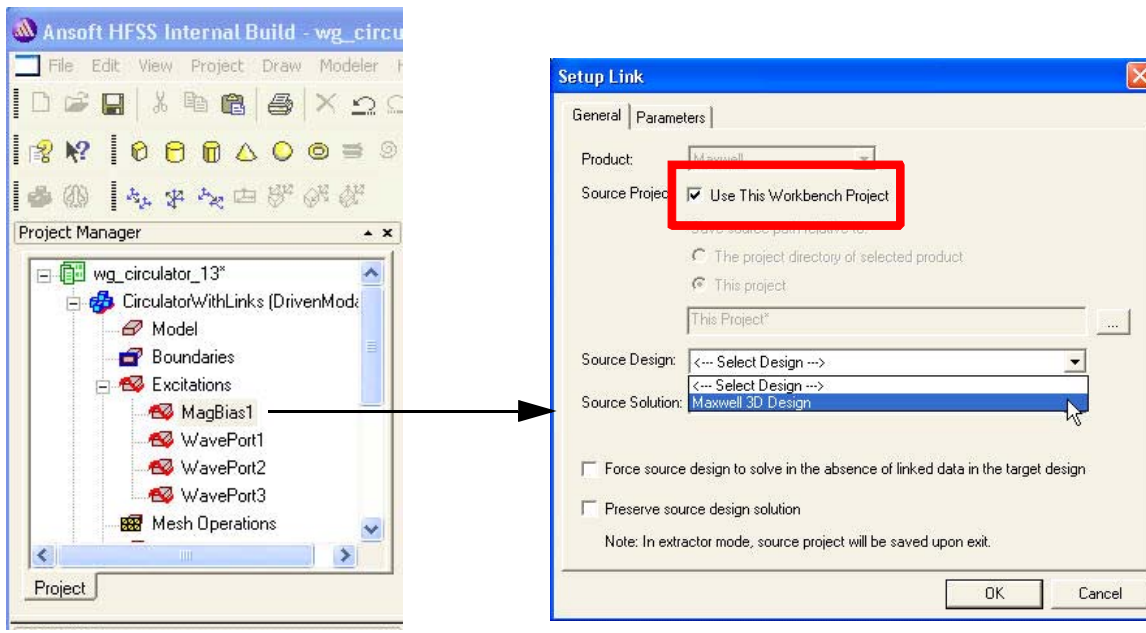


Maxwell 3D Model **Deformation of the Stator** **Coil Deformation**

Multiphysics Coupling between ANSYS EM Field Systems on Workbench

You can setup links between ANSYS Electromagnetics field systems that reside on a Workbench project schematic. Linking is setup in the ANSYS Electromagnetics application as shown in the example below.





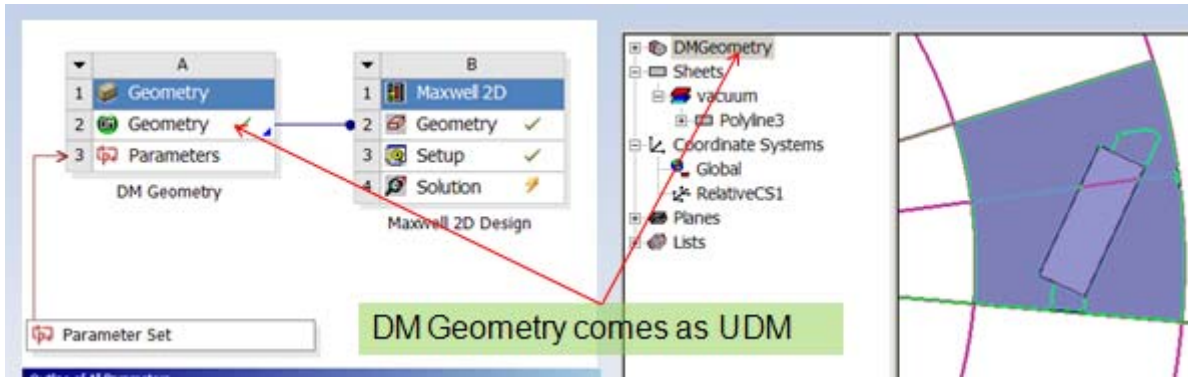
Related Topics

Feedback Iterator

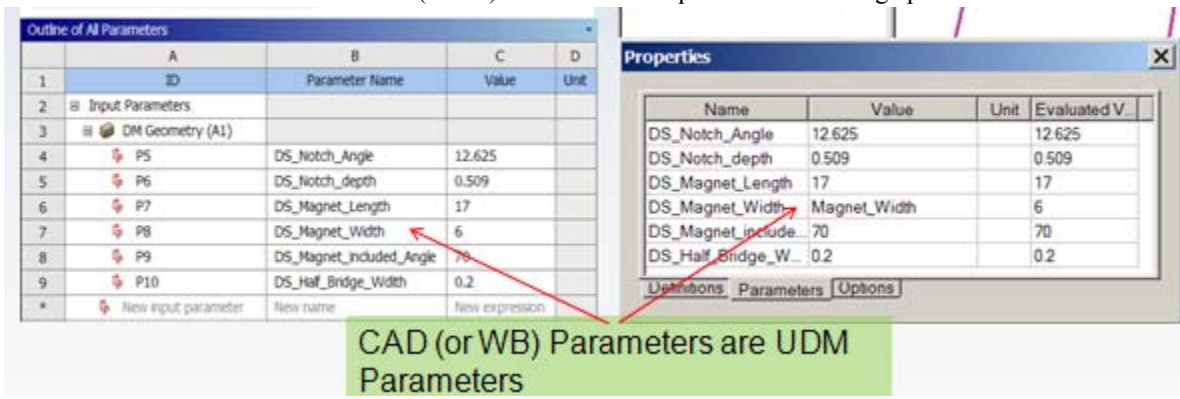
ANSYS EM CAD Integration through Workbench

ANSYS Electromagnetics CAD integration is a Workbench feature available for ANSYS Electromagnetics 3D Products - HFSS, Maxwell and Q3D, as the ANSYS Framework. The feature is available only through Workbench, not available with standalone ANSYS Electromagnetics products.

ANSYS CAD integration provides a bi-directional dynamic link through Workbench, which makes it possible to get updated geometry from CAD and to modify CAD parameters in ANSYS Electromagnetics products to return updated geometry. The feature is non-associative due to a need to reassign boundaries if modified CAD model is to be used. The process creates a User Defined Model (UDM) for each geometry source



The User Defined Model (UDM) format makes it possible to exchange parameters.



See here for further description of the UDM feature and function.

ANSYS Electromagnetics CAD integration makes it possible to consume geometry from multiple upstream source which can be any CAD or ANSYS Electromagnetics product. This feature supports direct interfaces with all major CAD systems.

- Pro/E Wildfire
- UG NX

- CATIA V5
- SolidWorks
- Autodesk Inventor
- ANSYS Design Modeler (DM)
- ANSYS SpaceClaim Direct Modeler (SCDM)

CAD software must be installed on user machine

- Not required on solve nodes

Platforms Supported

- Windows 32 bit
- Windows 64 bit
- Linux is NOT supported but it is possible setup project on windows and use Linux nodes for solves.

See the following sections:

CAD Integration Functionality

CAD Integration and Geometry Sharing

Bi-Directional CAD Integration

CAD Integration Model Edits

Multiple Geometry Links for CAD Integration

CAD Integration Functionality

Healing with CAD Integration

Important Geometry Options for CAD Integration

CAD Integration Functionality

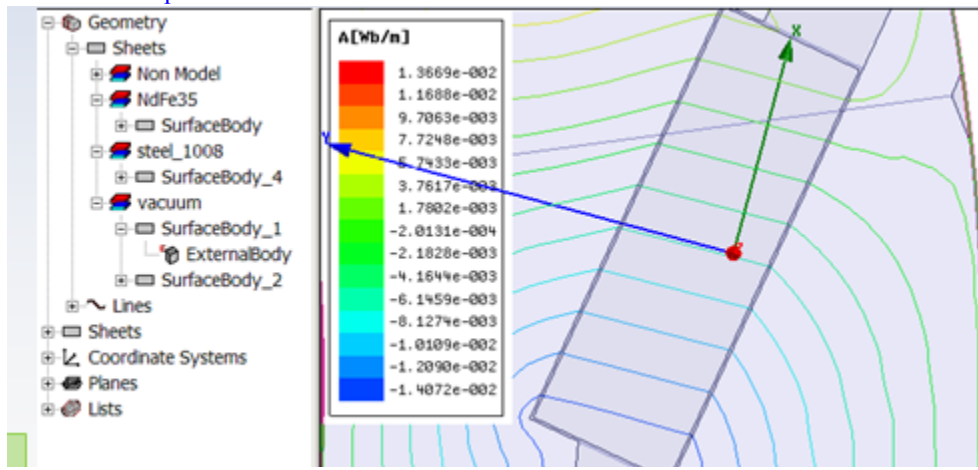
CAD Integration includes the following functionality.

- DX analysis

  WB Update Project

  WB Update All Design Points
- Parametric analysis with DSO
- Animation
- Geometry

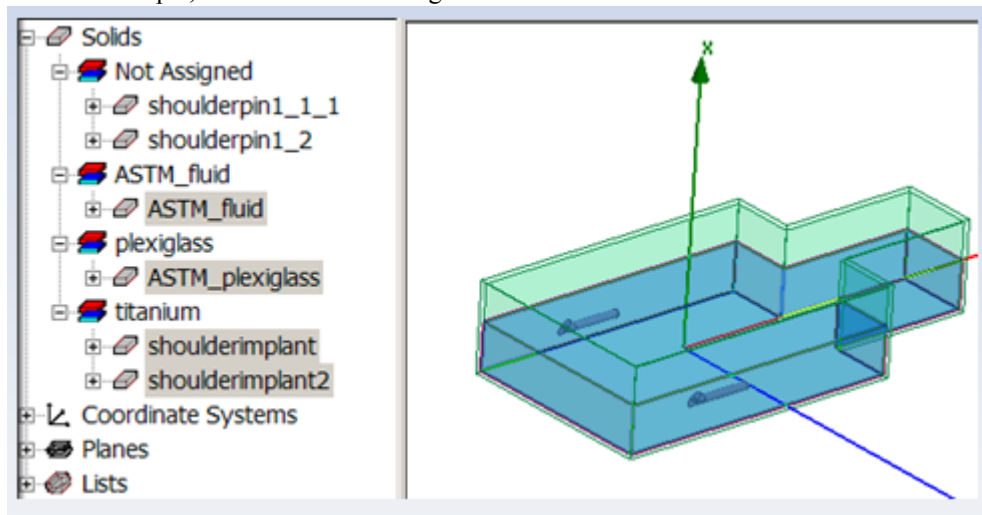
- Field plots



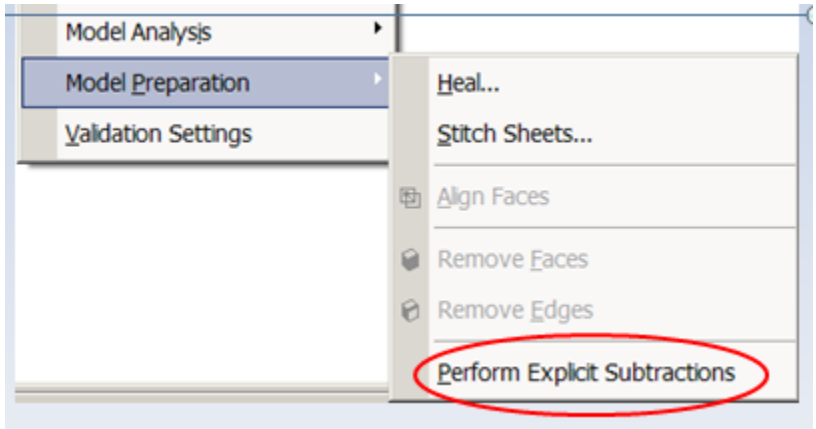
For Explicit Subtraction

- ANSYS Mechanical does not do implicit subtraction
- ANSYS Electromagnetics modeler command **Modeler>Model Preparation>Perform Explicit Subtraction** can do explicit subtractions before sending geometry to ANSYS.

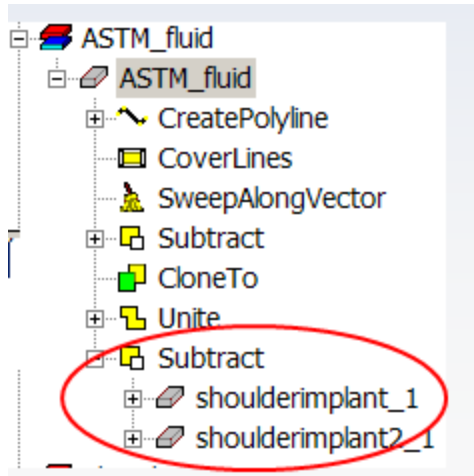
For example, consider the following model.



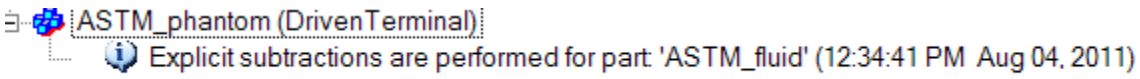
Perform Explicit Substraction can be performed.



The results appear in the History tree as shown:



The Message window reports this action.



Related Topics

User Defined Model (UDM)

ANSYS EM to ANSYS Geometry Transfer

Material Assignment Transfer

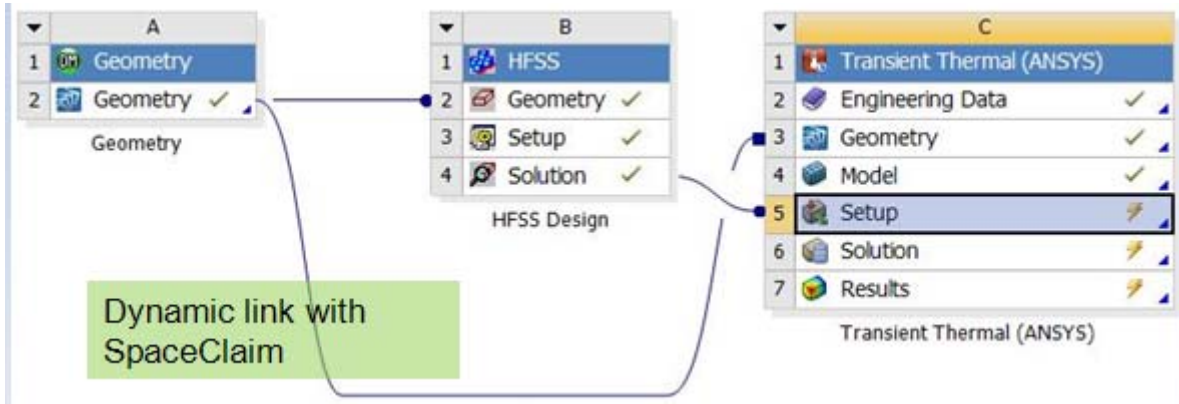
Geometry Transfer through ANSYS DesignModeler (DM)

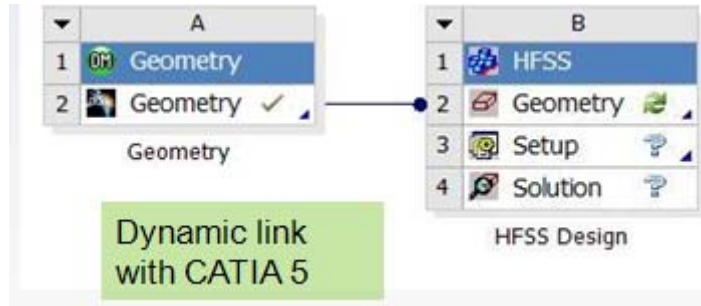
CAD Integration and Geometry Sharing

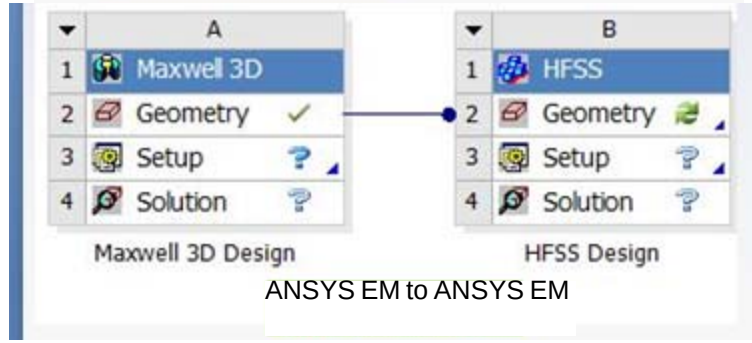
CAD model comes into ANSYS Electromagnetics as User Defined Model (UDM).

The input to ANSYS Electromagnetics from CAD is:

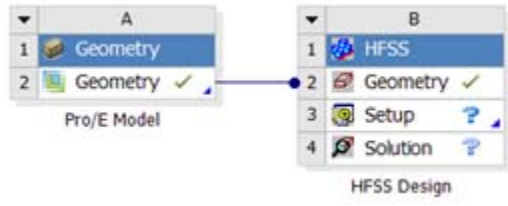
- Geometry/Topology with persistent IDs
- CAD parameters
- Material assignment
- Attributes like name, and color



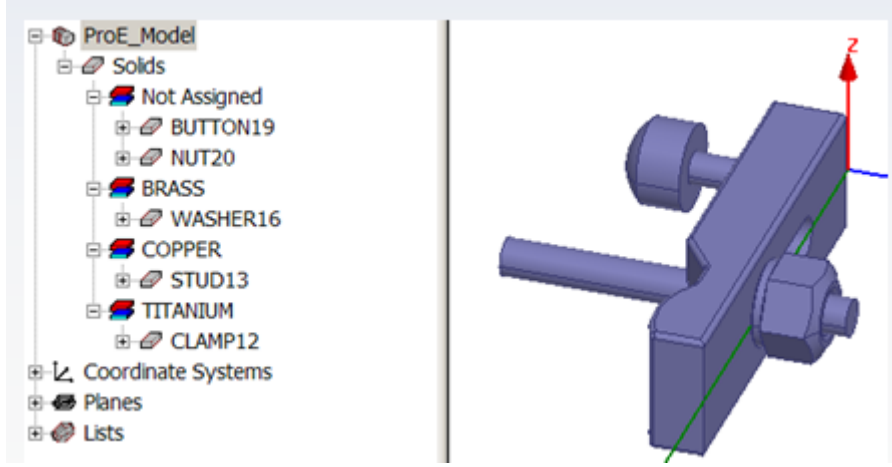




For example, in Workbench, a Pro/E Model can be linked to ANSYS.



The geometry can then be viewed in HFSS as a UDM.

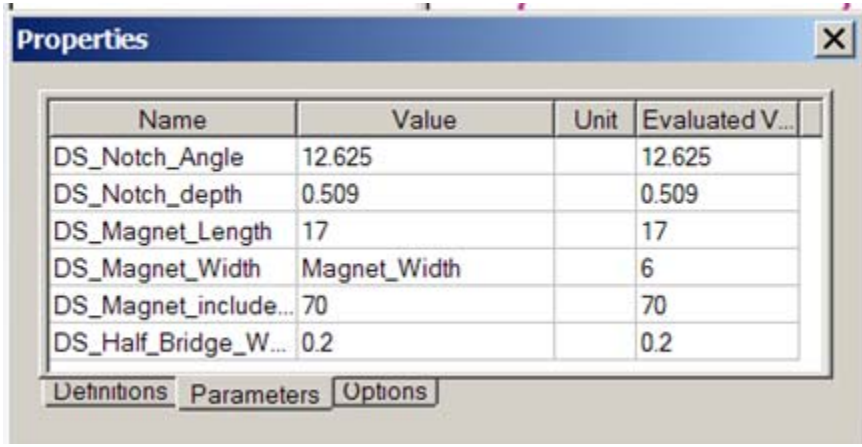


The CAD or WB model parameters appear in the Workbench:

Outline of All Parameters

	A	B	C	D
1	ID	Parameter Name	Value	Unit
2	Input Parameters			
3	DM Geometry (A1)			
4	P5	DS_Notch_Angle	12.625	
5	P6	DS_Notch_depth	0.509	
6	P7	DS_Magnet_Length	17	
7	P8	DS_Magnet_Width	6	
8	P9	DS_Magnet_included_Angle	70	
9	P10	DS_Half_Bridge_Wdith	0.2	
*	New input parameter	New name	New expression	

Though the ANSYS Electromagnetics CAD Integration, the linked UDM includes the same parameters.



Related Topics

ANSYS EM CAD Integration Through Workbench

Bi-Directional CAD Integration

CAD Integration Model Edits

Multiple Geometry Links for CAD Integration

CAD Integration Functionality

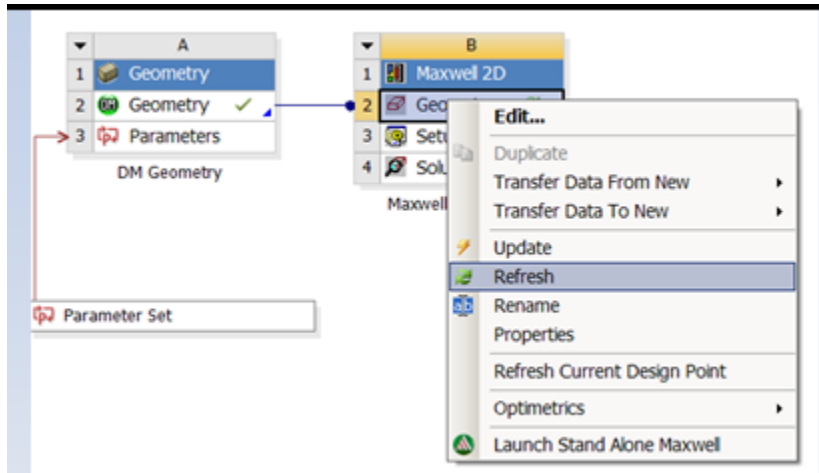
Healing with CAD Integration

Important Geometry Options for CAD Integration

Bi-Directional CAD Integration

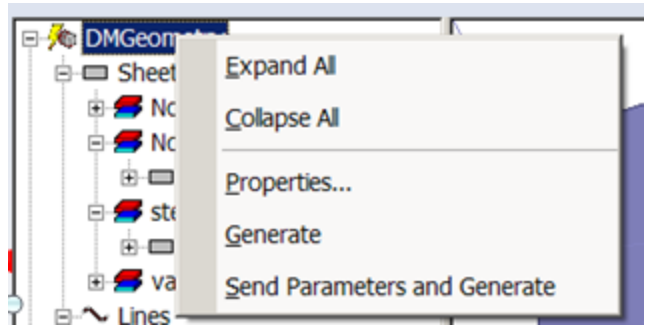
ANSYS Electromagnetics CAD Integrations uses Refresh (or Generate) CAD Model to pass updates.

For example, you make an edit in a CAD application and either run **Refresh** on an ANSYS Electromagnetics Geometry Cell or run **Generate** on the UDM in the ANSYS Electromagnetics window



Refresh pulls the current state of CAD model (geometry, parameters, materials etc) and updates the corresponding data in the ANSYS Electromagnetics application.

If you edit UDM (CAD) parameters in the ANSYS Electromagnetics modeler window you can run the **Send Parameters and Generate** command



The command passes the edited parameters to the linked CAD application and then pulls corresponding CAD geometry.

Related Topics

ANSYS EM CAD Integration Through Workbench

CAD Integration and Geometry Sharing

CAD Integration Model Edits

Multiple Geometry Links for CAD Integration

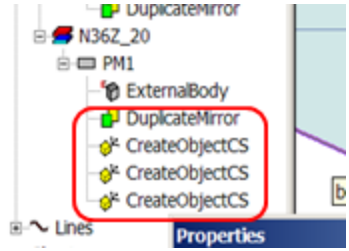
CAD Integration Functionality

Healing with CAD Integration

Important Geometry Options for CAD Integration

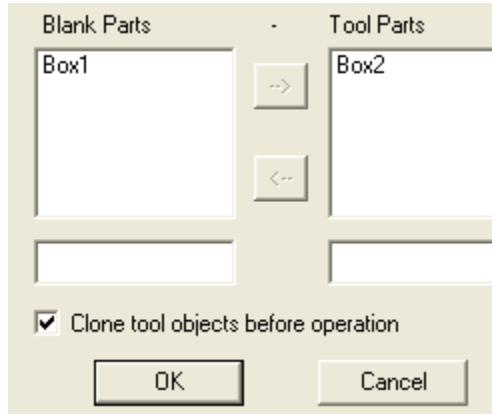
CAD Integration Model Edits

Several modeling operations are allowed on a CAD model in the ANSYS Electromagnetics Modeler window. Operations will included in the History tree and retained during model **Refresh**.



The following operations are not allowed

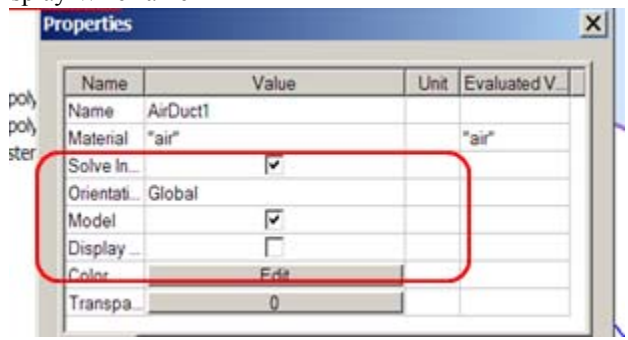
- Non history tree operations like Heal or defeature.
- Operations which use UDM parts as Tool such as Sweep, or Boolean operations like Split or Unite (but allowed when you select the clone tool option).



The following part attributes can be modified for UDM parts.

- Model/Non Model flag
- Solve Inside flag
- Part orientation
- Color

- Display Wireframe



It is not possible to delete individual parts of UDM

Related Topics

ANSYS EM CAD Integration Through Workbench

CAD Integration and Geometry Sharing

Bi-Directional CAD Integration

Multiple Geometry Links for CAD Integration

CAD Integration Functionality

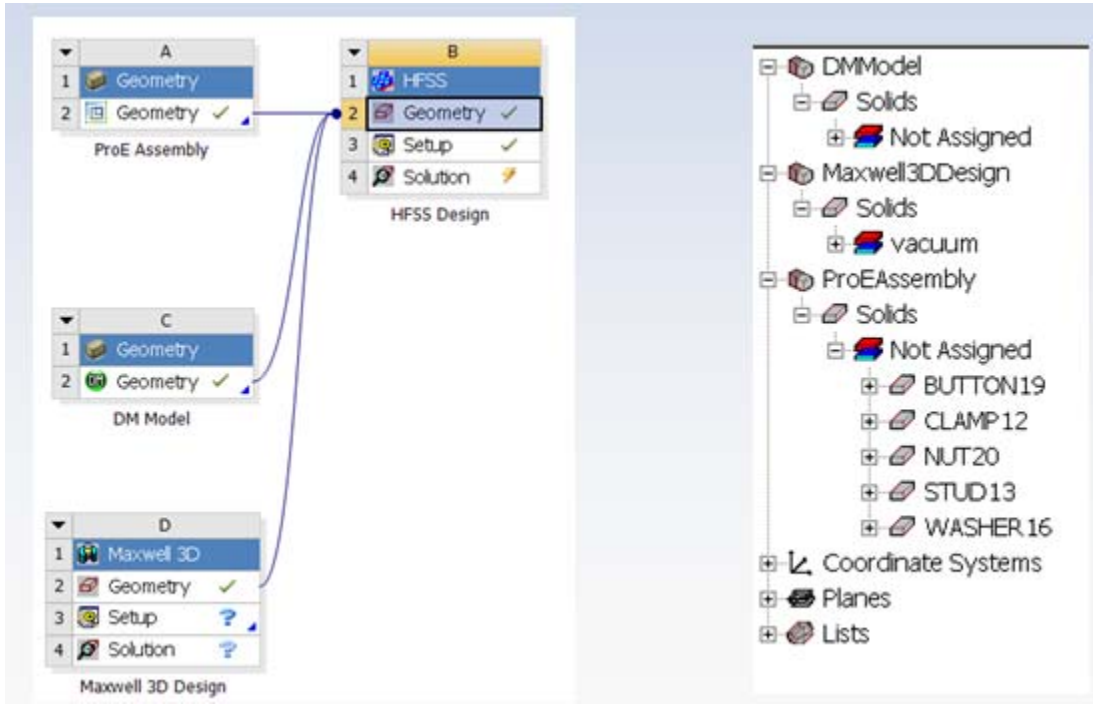
Healing with CAD Integration

Important Geometry Options for CAD Integration

Multiple Geometry Links for CAD Integration

With CAD Integration you can consume geometry from multiple upstream sources. The Source can be any of CAD or ANSYS Electromagnetics products. This creates a UDM for each geometry source.

For example, the following figure shows a DM Model, a Maxwell model, and ProE model linked to HFSS in Workbench, and displayed in the HFSS History tree as three UDMs.



Related Topics

ANSYS EM CAD Integration Through Workbench

CAD Integration and Geometry Sharing

Bi-Directional CAD Integration

CAD Integration Model Edits

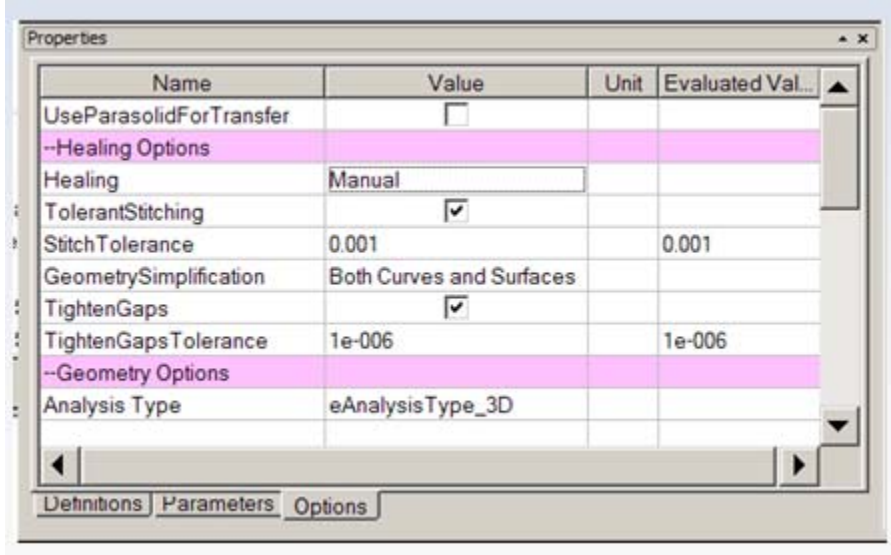
CAD Integration Functionality

Healing with CAD Integration

Important Geometry Options for CAD Integration

Healing with CAD Integration

It is not possible to use the Heal command in ANSYS Electromagnetics Modeler. Instead similar healing options are available under UDM properties option tab.



The Healing options are: None, Auto and Manual. By default healing is off (None) and should be turned on only if required.

Related Topics

ANSYS EM CAD Integration Through Workbench

CAD Integration and Geometry Sharing

Bi-Directional CAD Integration

CAD Integration Model Edits

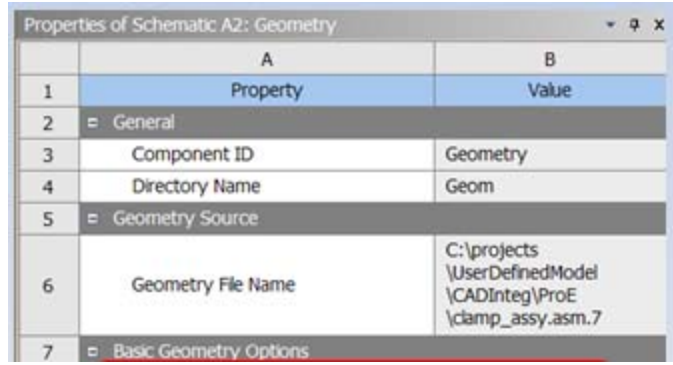
Multiple Geometry Links for CAD Integration

CAD Integration Functionality

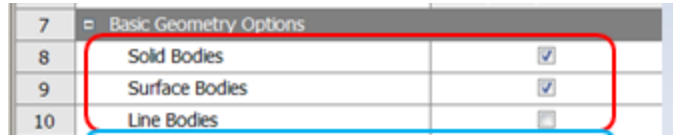
Important Geometry Options for CAD Integration

Important Geometry Options for CAD Integration

Select a Geometry Cell in Workbench to see options in Properties window.



- Control dimension of bodies coming from CAD

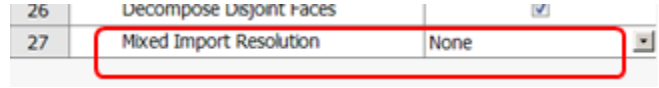


- Make sure parameters is checked and parameter key (filter) is appropriate to bring CAD parameters.

- Attributes key should be empty or Color to bring in CAD Colors

- Material properties must be checked to bring in the material assignment.

- The Mixed import resolution option is used to resolve parts with mixed dimension (typically from Pro/E)

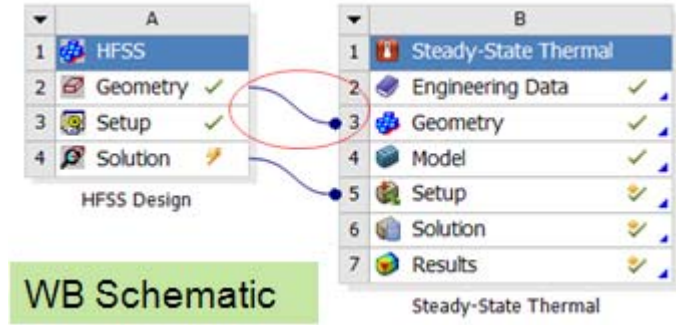


Refer to ANSYS Help for details: Path // CAD Integration // Overview :: 2 // Project Schematic Presence Related to CAD Integration // Geometry Preferences

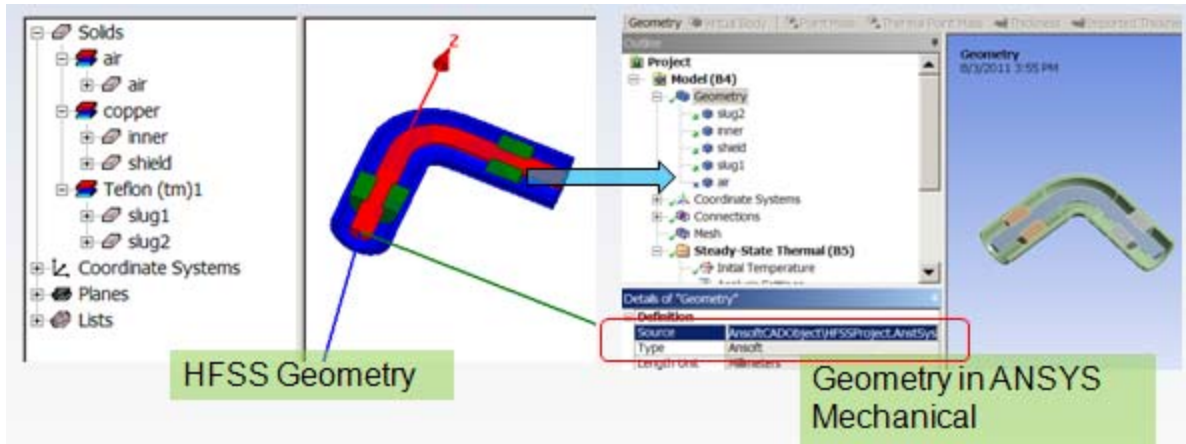
Related Topics

ANSYS EM to ANSYS Geometry Transfer

ANSYS Electromagnetics CAD Integration transfers model information based on Workbench links.

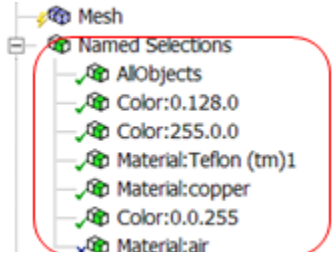


The following figure shows how the information is transferred between simulators.

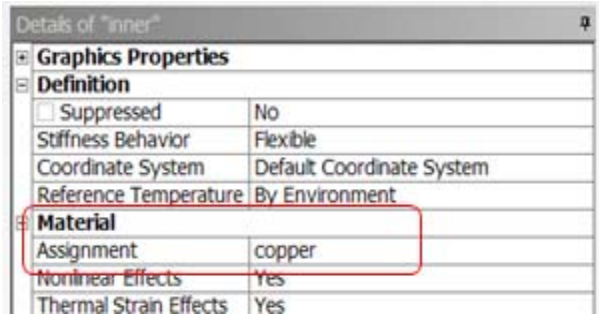


The information transferred includes:

- Geometry
- ANSYS Electromagnetics lists and material assignment as Named Selection



- Material assignment



The CAD Integration geometry link is

- Dynamic because you can get updated geometry from ANSYS EM
- Associative because IDs persist between ANSYS Electromagnetics model and ANSYS model during model refresh.

  Boundary conditions in ANSYS are preserved.

Related Topics

User Defined Model (UDM) for ANSYS WB Integration

Material Assignment Transfer

Geometry Transfer through ANSYS DesignModeler (DM)

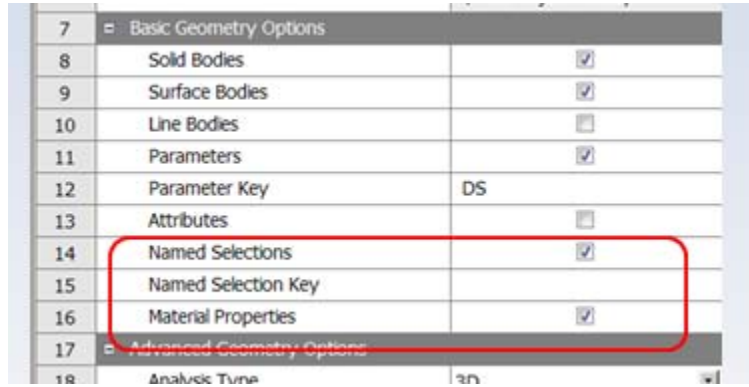
Material Assignment Transfer

Pre-requisite for material assignment transfer

- Engineering Data should have materials used in ANSYS Electromagnetics model and material names should match with case.
- Material Properties in Geometry Options must be checked

Pre-requisite for transfer as named selection

- Named Selection in Geometry Options must be checked



- Named Selection Key should either be empty or include 'Material'

Related Topics

User Defined Model (UDM) for ANSYS WB Integration

ANSYS EM to ANSYS Geometry Transfer

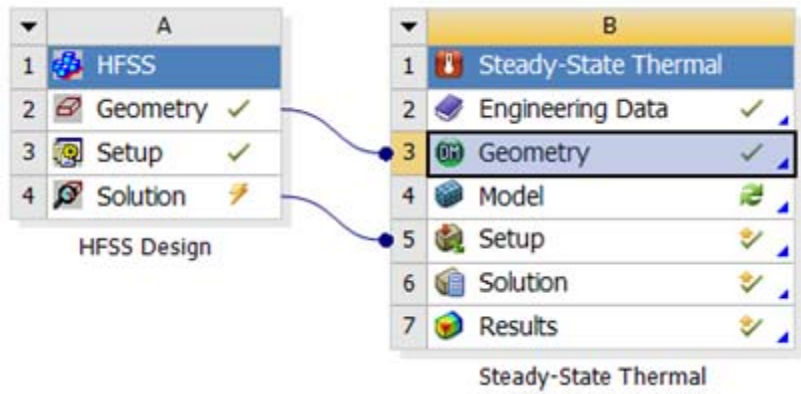
Geometry Transfer through ANSYS DesignModeler (DM)

Geometry Transfer through ANSYS DesignModeler (DM)

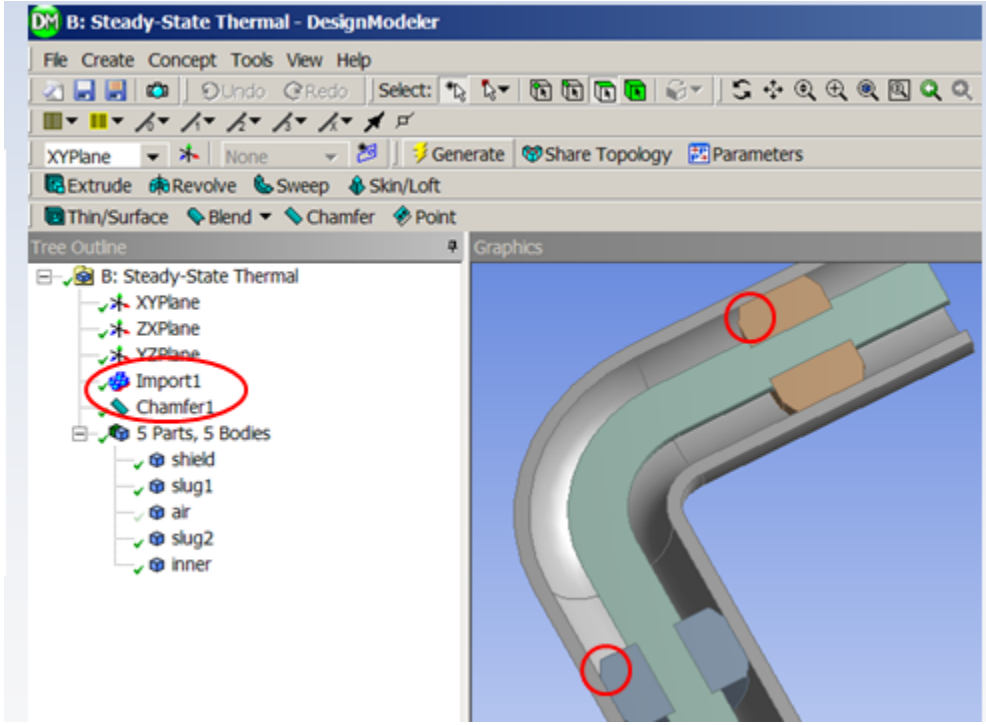
It is possible to edit geometry in ANSYS DesignModeler (DM) before consuming in Mechanical

- Useful when geometry needs preprocessing for ANSYS simulation
- Can take advantage of geometry commands available in DM

For example, consider an HFSS model linked to DM through the Workbench.



In this case, the figure below shows how a chamfer operation on geometries is imported.



It is possible to edit geometry in ANSYS DesignModeler (DM) before consuming in Mechanical

- Useful when geometry needs preprocessing for ANSYS simulation
- Can take advantage of geometry commands available in DM

Related Topics

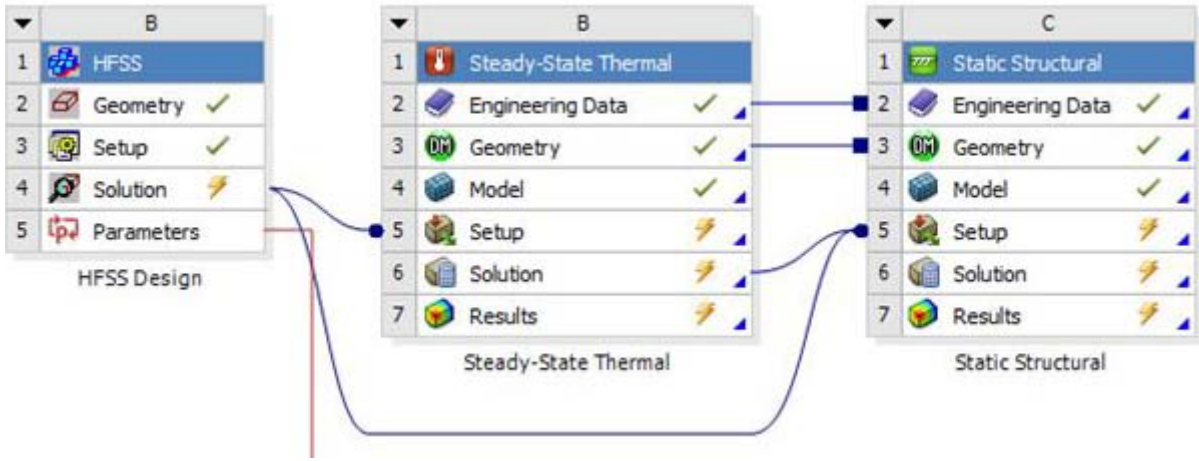
User Defined Model (UDM) for ANSYS WB Integration

ANSYS EM to ANSYS Geometry Transfer

Material Assignment Transfer

Stress Feedback to HFSS using Workbench

Including stress feedback to HFSS involves coupling an HFSS system with both thermal and stress systems. The thermal system serves as an upstream system to the stress system and simulates the effect of heat induced displacement. And through the Workbench link between the Solution cell of HFSS and the setup cell of Mechanical, the displacement can feedback into HFSS



You use a dialog to select one or more objects (geometry) in HFSS upon which displacement data from Mechanical applies. This feature lets you avoid applying displacement to the whole model where only certain parts are sensitive to stress. You must select the same objects in the thermal and stress system to apply the respective imported loads.

The results of the process include deformation feedback applied to selected objects. The Solution Profile includes a report of Solve with thermal/displacement feedback, including information Maximum Delta T and Maximum Delta Displacement. You can create field plots with based on Displacement and Mag_Displacement, and you can scale the deformation displayed. The Fields Calculator also includes these input quantities.

Interface Changes for Stress Feedback from Mechanical to HFSS

Process Flow for Stress Feedback to HFSS

Interface Changes for Stress Feedback from Mechanical to HFSS

When used through the Workbench for stress feedback projects, the HFSS interface contains some commands and information that does not appear in standalone HFSS.

Deformation of Objects dialog

Solution Profile information for Solve with Displacement Feedback

Plot Fields Menu with Displacement and Mag_Displacement

Deformation Scale tab for Modify Plot Attributes

Revert to Zero Displacement command under the **Solve Setup** menus

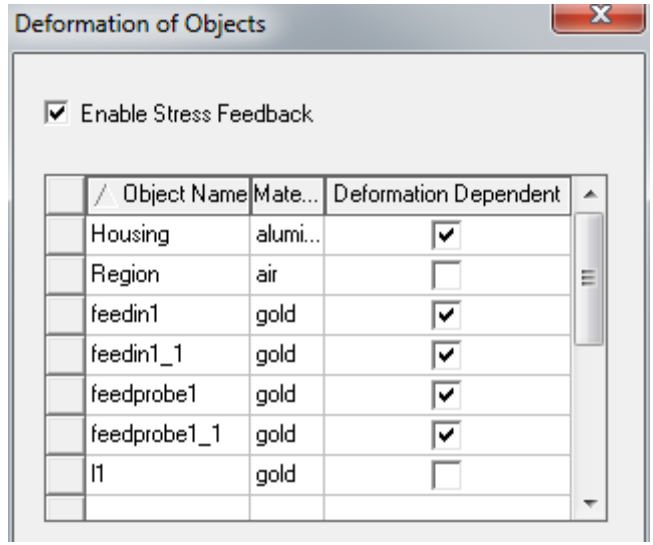
Fields Calculator Field Quantities for Stress Feedback Projects

Plot Fields Context Menu for Stress Feedback Projects

Modify Plot Attributes Dialog for Stress Feedback Projects

Deformation of Objects Dialog

When working with HFSS through the Workbench, the HFSS menu includes a Deformation of objects command that opens this dialog for specifying which objects shall be deformation dependent.



The Enable Stress Feedback checkbox enables object selection. All selections are disabled when this is unchecked. Changing the setting invalidates any existing solutions.

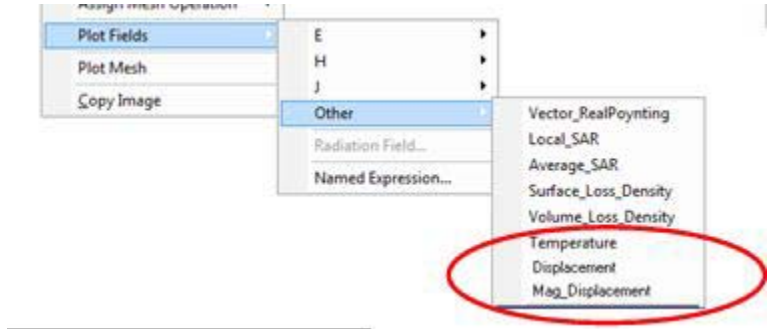
The objects listed do not include objects without materials assigned, non-model objects, or non-3D objects. Objects assigned materials such as air or vacuum are not deformed and don't apply to stress feedback.

Solution Profile with Displacement Feedback

The Solution Profile includes a report of Solve with thermal/displacement feedback, including information Maximum Delta T and Maximum Delta Displacement.

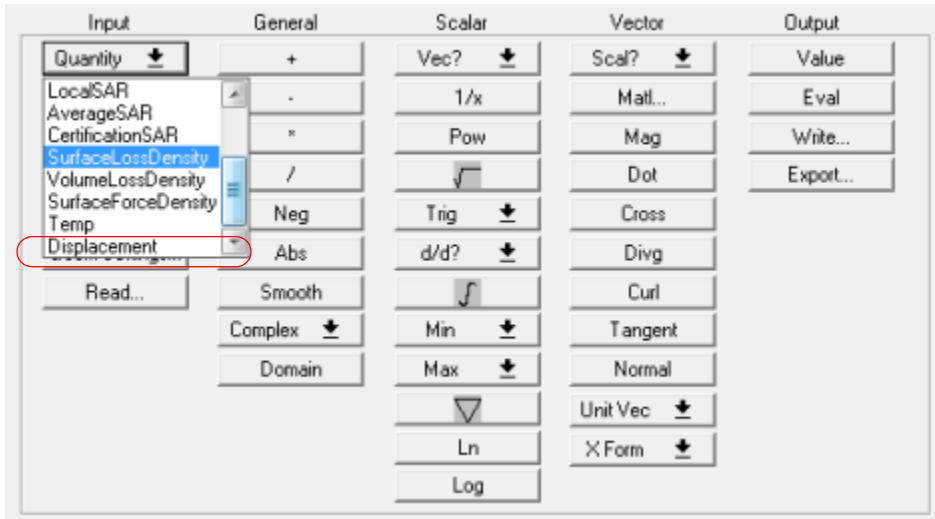
Plot Fields Menu with Displacement and Mag_Displacement

The Plot Fields menu includes entries for displacement and mag_displacement.



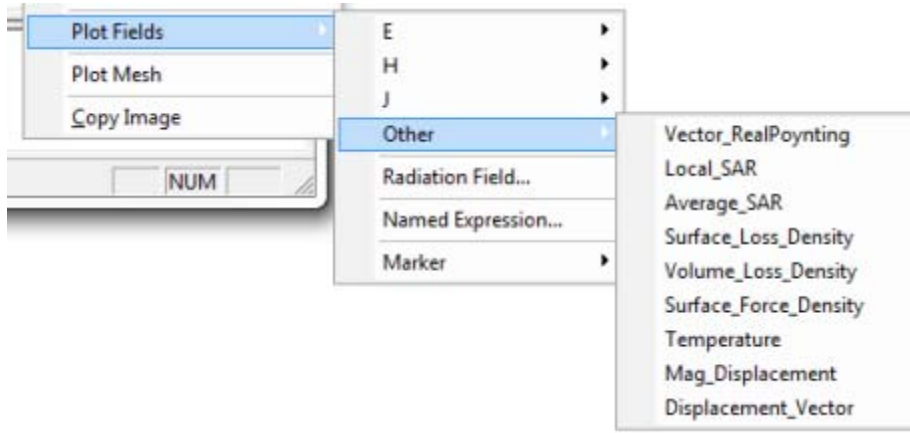
Fields Calculator Field Quantities for Stress Feedback Projects

The Quantity menu for the Fields Calculator lists Displacement for Stress Feedback projects



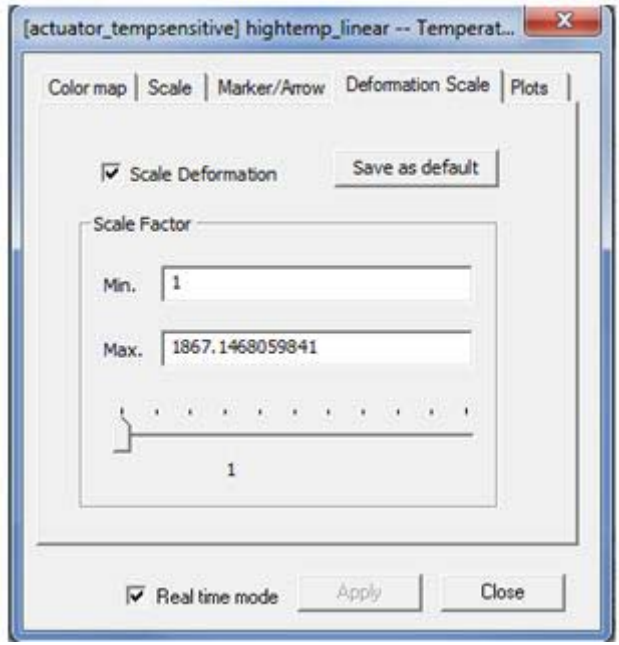
Plot Fields Context Menu for Stress Feedback Projects

The **Plot Fields>Other** menu includes Displacement_vector and Mag_Displacement, a scalar. A field plot remains empty until the displacement data applies to the HFSS solution.



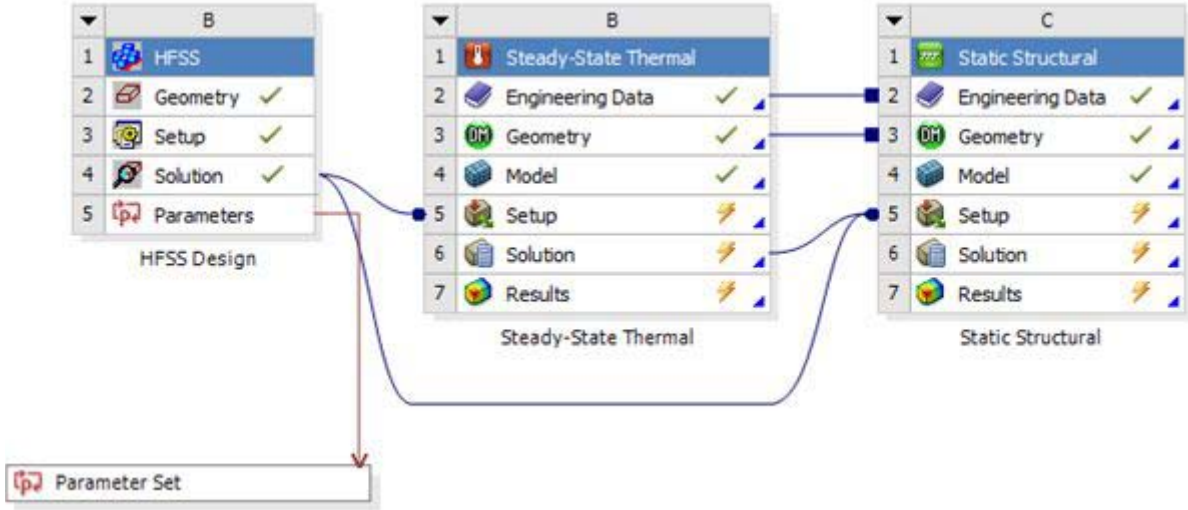
Modify Plot Attributes Dialog for Stress Feedback Projects

When an HFSS plot includes deformation, the **HFSS>Fields>Modify Plot Attributes** dialog includes a tab for scaling the deformation.

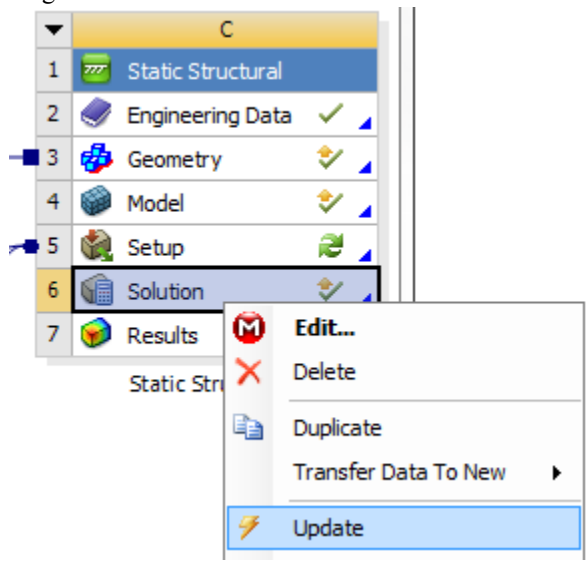


Process Flow for HFSS Workbench for Stress Feedback

To setup the coupling, drag the "Solution" cell of the HFSS system and drop it at the "Setup" cell of a Thermal and Stress system. You also need to also couple the Thermal and Stress system to capture the effect of thermal force. The image below illustrates a coupling setup where both the Thermal and Stress system are ready to be "Updated". The HFSS adaptive solution has converged in 3 passes. The HFSS system has both thermal and stress feedback enabled.



1. Right click on the Solution cell of Stress and select "Update"

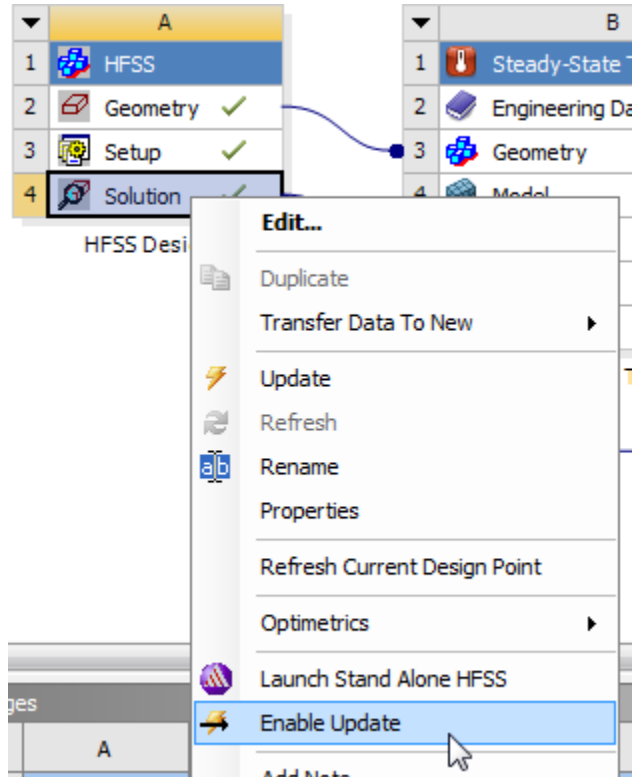


   a. The Setup of Thermal will be "Updated" with em loss from HFSS
   b. The Solution of Thermal will be "Updated" and temperature will be exported to HFSS

c. The Setup of Stress will be "Updated" with thermal force from Thermal and force density (just zeros) from HFSS

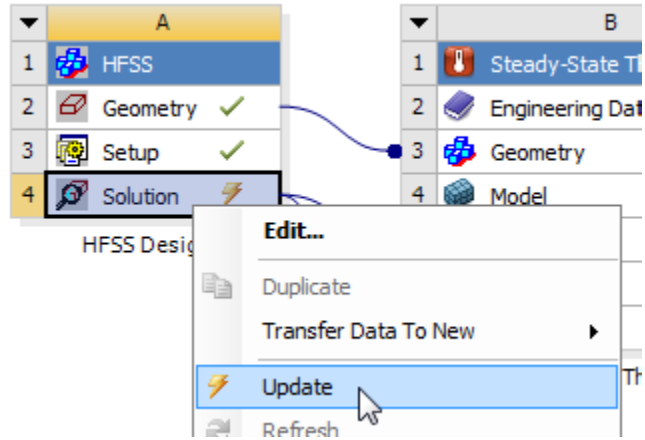
d. Displacement will be exported to HFSS after Stress finishes simulation

2. Right click on the Solution cell of HFSS and select "Enable Update"

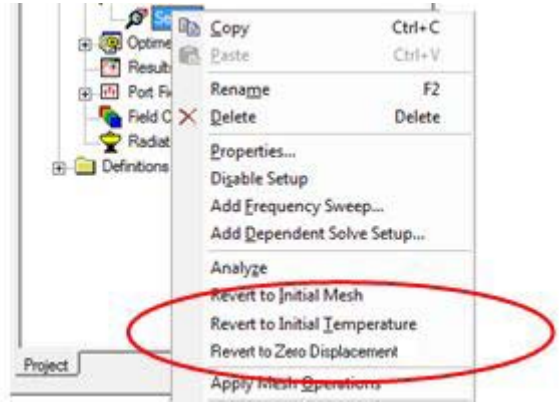


The Solution cell icon changes from a green checkmark to the Update lighting icon.

3. Right click on the Solution cell of HFSS and select "Update"

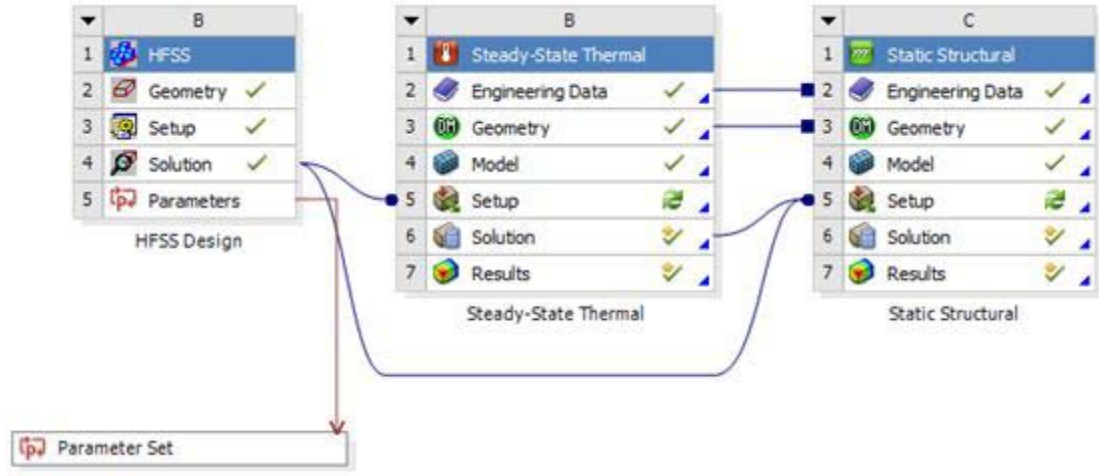


   a. HFSS re-simulate the 3rd pass with its mesh and the temperature/displacement feedback
   b. The profile will show information about the feedback
   c. Both "Revert" menus will be available at the right mouse click menu of the solve setup item.

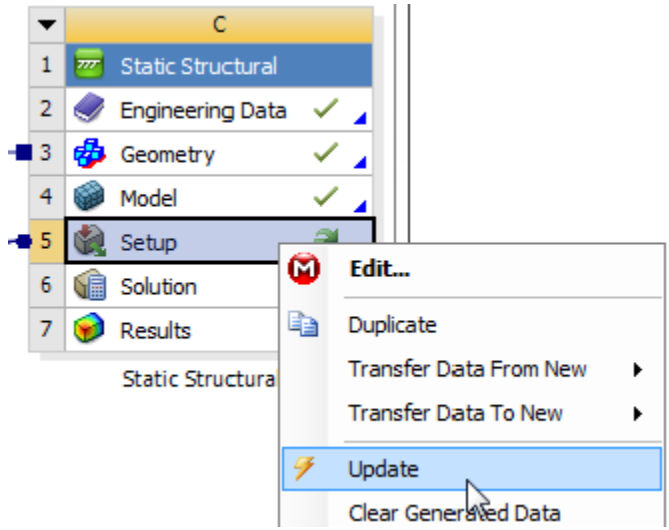


Manual Iteration After First Iteration Completes

The image below illustrates the WB schematic after the previous situation, after the 1st iteration has completed. Notice the changed icons for the status for Setup for Thermal and Structural.

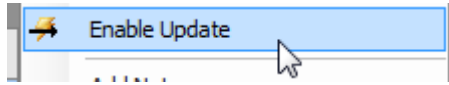


1. Right click on the Solution cell of Stress and select "Update"

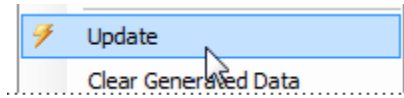


   a. The Setup of Thermal will be "Refreshed" and "Updated" with EM loss from HFSS
   b. The Solution of Thermal will be "Updated" and temperature will be exported to HFSS
   c. The Setup of Stress will be "Refreshed" and "Updated" with thermal force from Thermal and force density from HFSS
   d. Displacement will be exported to HFSS after Stress finishes simulation

2. Right click on the Solution cell of HFSS and select "Enable Update"



3. Right click on the Solution cell of HFSS and select "Update"



   a. HFSS re-simulates the 3rd pass with its mesh and NEW temperature/displacement feedback
   b. The profile will show information about the feedback

Note that the delta temperature and displacement is being reported in the profile

Revert the HFSS Solution For Temperature and Displacement

You can revert the temperature and displacement in HFSS solution separately.

1. Select "Revert to Initial Temperature" in HFSS.

   A warning message notifies you about the invalidation of solution

2. Right click on HFSS solve setup and select "Analyze"

   HFSS re-simulates the 3rd pass with its mesh and displacement that was previously exported from Mechanical, but without temperature.

3. Select "Revert to Zero Displacement" in HFSS, and followed by "Analyze"

   HFSS re-simulates the 3rd pass with its mesh without neither temperature nor displacement

Only Stress Feedback

The coupling setup is the same as in Use Case 1, with either one of the following differences

- The HFSS system is not enabled for thermal feedback. Since HFSS is not enabled to support feedback, the "Export Result" properties will be missing in the thermal system's property window.
- The HFSS system is enabled for thermal feedback, but users select not to "Export Result" in the thermal system. This is to disable the "automatic" feedback after Mechanical finishes its simulation.

1. Right click on the Solution cell of Stress and select "Update"
   a. The Setup of Thermal will be "Updated" with em loss from HFSS
   b. The Solution of Thermal will be "Updated"
   c. The Setup of Stress will be "Updated" with thermal force from Thermal and force density from HFSS
   d. Displacement will be exported to HFSS after Stress finishes simulation
2. Right click on the Solution cell of HFSS and select "Enable Update"
3. Right click on the Solution cell of HFSS and select "Update"
   a. HFSS re-simulate the 3rd pass with its mesh and the displacement feedback
   b. The profile will show information about the feedback and "Revert to Zero Displacement" will be enabled at the RMC menu of the solve setup item

Related Topics

Feedback Iterator

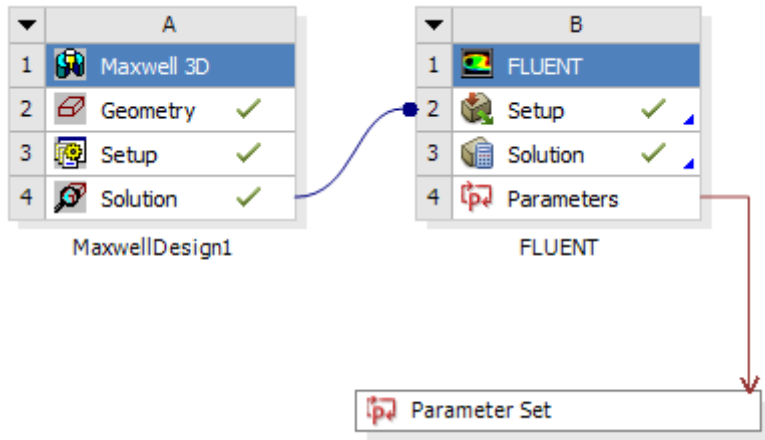
Feedback Iterator

Background Information

Prior to the introduction of the Feedback Iterator, ANSYS Workbench supported a two-way loose-coupling protocol with ANSYS Electromagnetics products.

- SystemCoupling already uses the word coupling to mean low-level solver coupling. Existing Electromagnetics product coupling is loose when compared to SystemCoupling and is limited to file-transfers at the end of a complete solve in a stand-alone system/product. No communication occurs during a solve.
- The coupling is two-way

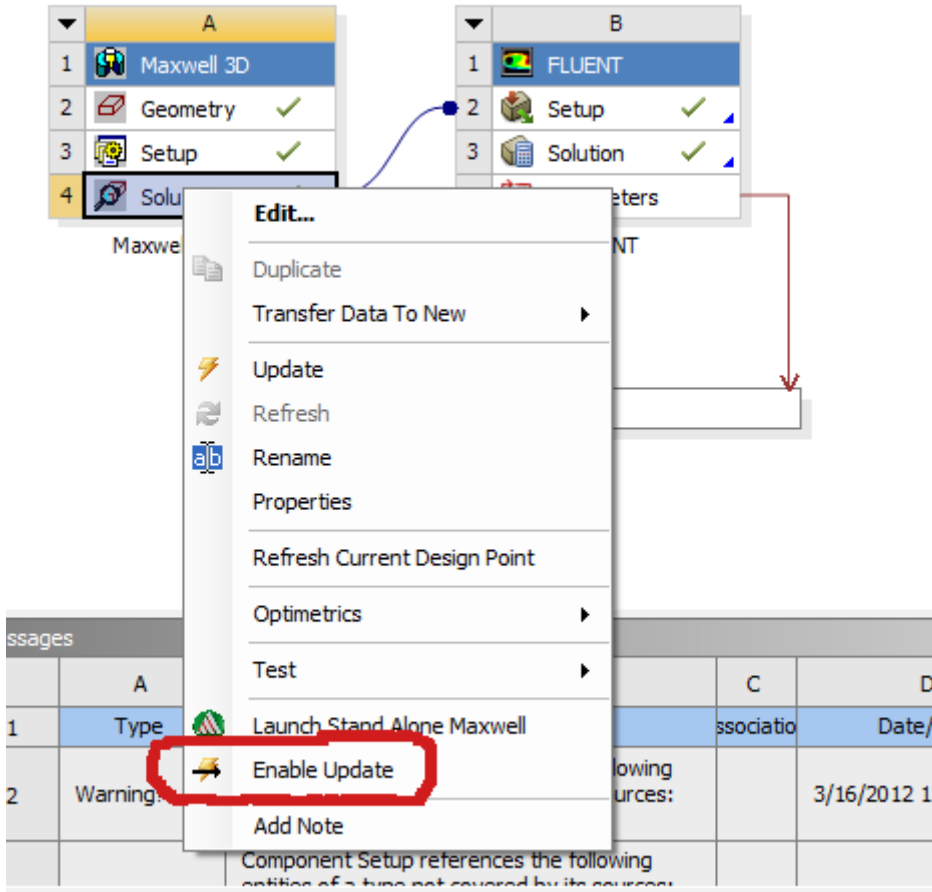
  The one way portion (Upstream source component to Downstream target component) is handled via existing workbench data/transfer connection mechanism.



The round trip is handled by a separate protocol agreed upon by the participating systems when the downstream system exports a set of file to a location specified by the upstream system (via its one way transfer). This exported data is then incorporated by the upstream system in its next update.

If the user chooses to run the next coupled solve iteration, he invoked the Enable Update Gui operation as shown below and then updates all systems involved. These steps (Enable Update,

Update Project) are continued as long as needed

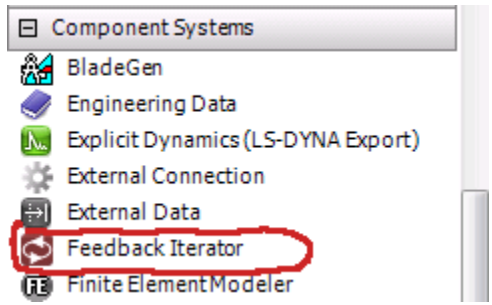


The Feedback Iterator System

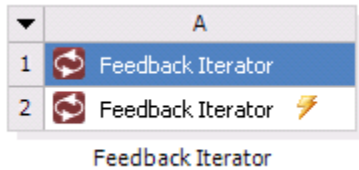
ANSYS Electromagnetics Suite 15.0 provides the Feedback Iterator system for automating the manual steps needed for driving a feedback utilizing system-pair to convergence. In addition to

automating feedback incorporation over a user-specified number of iterations, the Feedback Iterator also allows you to control the number of iterations (decide iteration termination criteria) and set target temperature and displacement convergence criteria.

The Feedback Iterator appears in the Workbench Toolbox user interface under Component Systems.



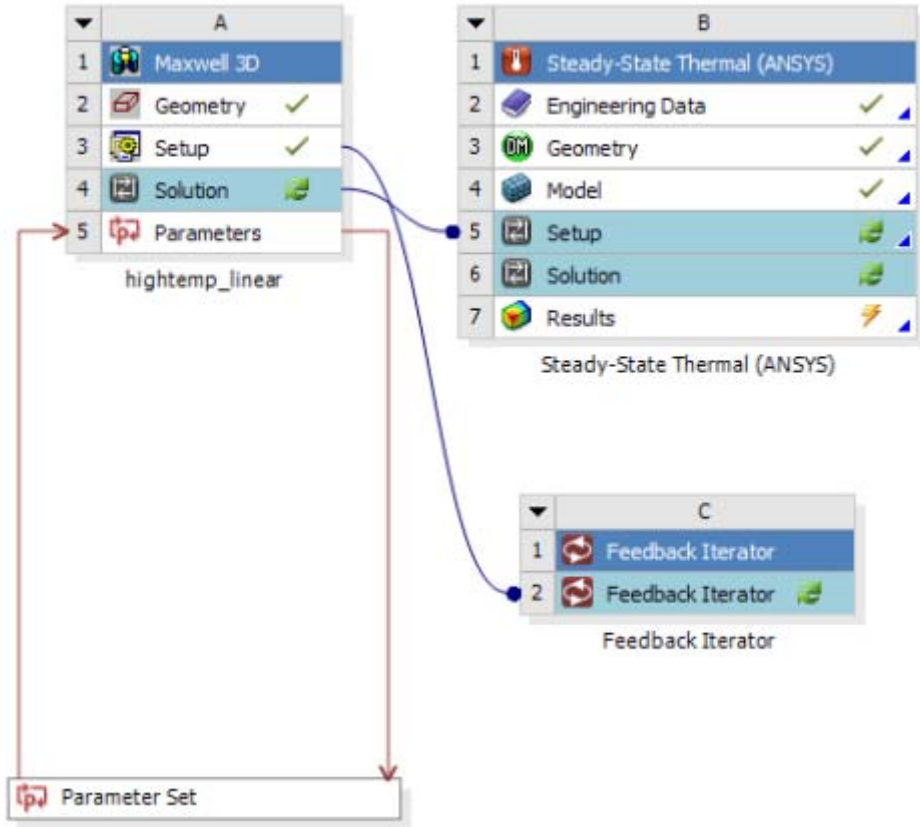
If you drag and drop an instance of the Feedback Iterator into Workbench Schematic, the default system name is Feedback Iterator.



Feedback Iterator

Feedback Iterator in Use

You attach the Feedback Iterator system to the Workbench schematic like any other system. For example, given a Maxwell system providing AnsoftHeatLossData to a fluent system, if you drop the Maxwell setup cell onto the Feedback Iterator component, the following links result..



While the examples deal with Maxwell and Steady-State Thermal, the Feedback Iterator Addin functions the same way with Maxwell or HFSS as sources and Fluent, Thermal or Structural as targets. Essentially all systems that participate in Electromagnetics product two-way loose coupling.

Feedback Iterator Component Properties

The Feedback iterator's primary role is to control iterative solves and its properties target this end.

For Maxwell and HFSS, the Feedback Iterator properties are as shown below.

	A	B
1	Property	Value
2	⊟ General	
3	Component ID	FeedbackIterator
4	Directory Name	FeedbackIteratorComponent
5	⊞ Notes	
7	⊞ Used Licenses	
9	⊟ Iterations	
10	Iterations Completed	0
11	Max Iterations	100
12	⊞ Callback	
14	⊟ Temperature Convergence	
15	Target Delta Temperature %	5
16	Latest Delta Temperature %	Not Available
17	⊟ Displacement Convergence	
18	Target Delta Displacement %	5
19	Latest Delta Displacement %	Not Available

- **Iterations Completed**: displays the number of iterations completed. A “read-only” property.
- **Max Iterations**: sets the maximum number of iterative solve loops to perform before terminating iterations in case the **Target Delta Temperature %** or **Target Delta Displacement %** is not achieved. Default value is 100.
- **Target Delta Temperature %**: specifies the maximum **Target Delta Temperature %** which signifies convergence. Default value is 5%. Value must be 0.01 or greater.

Note

- The delta temperature error is calculated base on the SI unit: Kelvin.
- The Maxwell and HFSS design Profile tab displays absolute/relative delta data, while the delta temperate in Workbench means the relative delta (expressed as %).

- **Target Delta Displacement %**: specifies the maximum **Target Delta Displacement %** which signifies convergence. Default value is 5%. Value must be 0.01 or greater.

The Feedback Iterator properties also display the most recently achieved **Latest Delta Temperature %** and **Latest Delta Displacement %** values, allowing the user to manually abort when suitably satisfied with the achieved deltas.

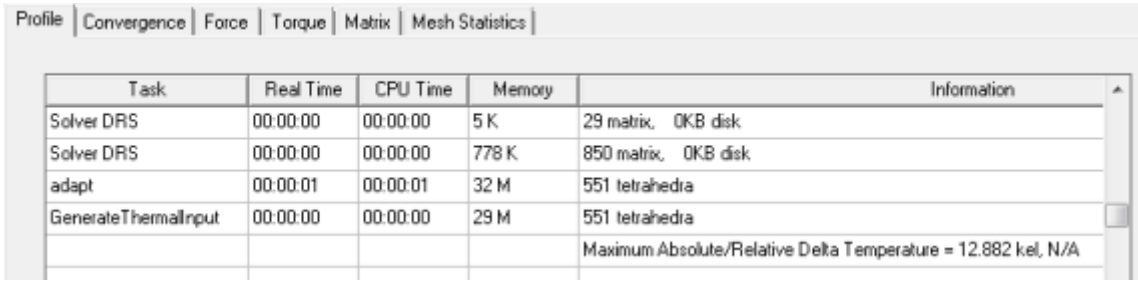
Note Both the Temperature and Displacement Convergence properties sections are always shown irrespective of which feedback types are actually enabled for the project.

When we run a two-way feedback simulation from WB, and keep the Maxwell or HFSS Profile tab open (right-click on Results in the Project Manager and select Solution Data, then select Profile tab), you will see that the 3D solver keeps track of two feedback related quantities:

- Maximum Absolute/Relative Delta Temperature (if temperature feedback is enabled)

Note The delta temperature error is calculated base on the SI unit: Kelvin.

- Maximum Absolute/Relative Delta Displacement (if displacement feedback is enabled)

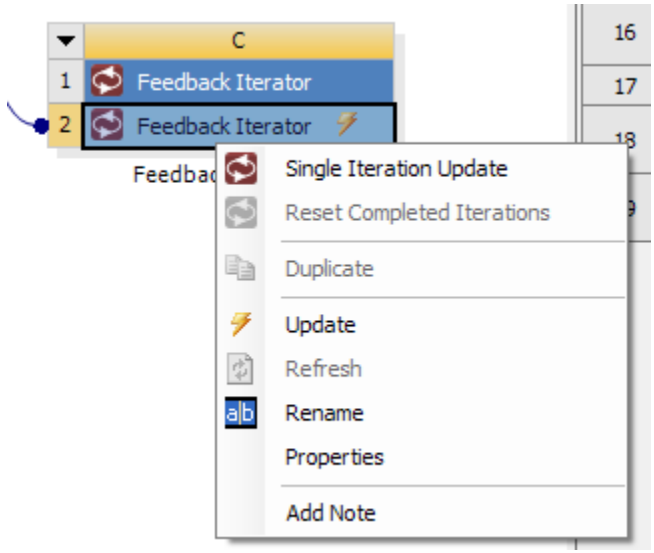


These signify the maximum difference on the solution mesh of the quantity in question from one feedback iteration to the next (thus the value is only available from iteration #2 onward). This data is the basis for the convergence controls in the FeedbackIterator. These controls allow the user to base the termination of the feedback iterations based on a target max delta T and/or max delta displacement. Thus the target max delta forms the convergence condition.

Feedback Iterator GUI Operations

There is currently only one GUI operation, “Single Iteration Update”, which allows you to run a single iteration worth of updates. A Single Iteration Update operates as follows:

- Updates Electromagnetics product (HFSS, Maxwell, Q3D) system (which will incorporate any previous feedback from the downstream system).
- Updates downstream system.
- All coupled systems: Electromagnetics product, Electromagnetics product Downstream and Iterator will be in the UpToDate state.
- Iterator's “Iterations Completed” property will increment by 1.



Right-clicking on the Feedback Iterator component and selecting Update allows you to run iterations automatically. Iterations continue to run either until convergence criteria are met, or until the Max Iterations value is reached.

Callback Interface

The callback interface allows you to react to each step of the iteration process as implemented by the FeedbackIterator component. This is useful for you to implement special iterations or even solve transient simulations (with a limited scope). To this end, there are four main API features provided:

- Callback functions which the FeedbackIterator calls at various points in each iteration
- Utility functions that the Callback functions can use to extra properties of the containers being processed
- Output functions that allow you to debug the script or supply additional messages
- Limited state management allowing the script to store and retrieve state across callback functions and iterations

Callback and State API

Callback and State API

State is managed somewhat simply but in a limited fashion. All the API methods take a final dictionary argument. This dictionary is limited to using **string keys** and **number or string values**. Within this limitation, simply add new keys, read old keys, clear the dictionary etc and it will be persisted across functions calls and across Updates even. When the **Iterations Completed** property is **rest**, the dictionary is also cleared out.

It is advisable for the callback script to initialize the dictionary at Iteration 1.

All the API methods use a subset of the following arguments

iterationNumber	An integer representing the current iteration. This always starts from 1.
ContainerList	A python list of containers (DataContainerReference). This is the entire list of the coupled containers managed by the FedbackIterator. You typically loop through them and using the utility methods listed below, identities them.
Container	A single container that is being processed. This is a DataContainerReference
State	A read/write python dictionary that is used to maintain state across function calls and iterations.

Only return values from BeforeIterationEx are processed. Returns from all other functions are discarded

1. BeforeIteration(IterationNumber, ContainerList, State): This method is called before each iteration. Ideally used to initialize the state dictionary, open editors as required or initialize setups as needed for each iteration.
2. BeforeIterationEx(IterationNumber, ContainerList, State): Similar to the BeforeIteration method except that this allows you to control the number of iterations via the return value
   - Return: "more" to request one more iteration
   - Return "last" to indicate that this is the last iteration
   - Any other return (including none) will be treated as a return of "last" and terminate iterations.
3. AfterIteration(IterationNumber, ContainerList, State): This method is called after each iteration. This can be used to copy result files over, check results, implement any possible convergence calculations, logging of results, etc.
4. BeforeContainerRefresh(IterationNumber, Container, ContainerList, State): Is called before each of the coupled containers is refreshed. The "Container" argument represents the container about to be refreshed.
5. AfterContainerRefresh(IterationNumber, Container, ContainerList, State): Is called after each of the coupled containers is refreshed. The "Container" argument represents the container just refreshed.
6. BeforeContainerUpdate(IterationNumber, Container, ContainerList, State): Is called before each of the coupled containers is Update (after a refresh). The "Container" argument represents the container about to be updated.
7. AfterContainerUpdate(IterationNumber, Container, ContainerList, State): Is called after each of the coupled containers is updated. The "Container" argument represents the container just updated.

If the callback scripts uses other files to send commands to various containers (vb, js, apdl, python etc), all of those files are best saved under the user_files directory. This allows you to use the FBGetUserFilePath(str) to get the absolute path of the file and allows the files to be packaged with any created archive.

Utility Functions

FBSystemForContainer(container)	system	Given a container, returns the system it belongs to
FBSystemDisplayName(system)	string	Given a system, returns its display name on the schematic
FBSystemID(system)	string	Given a system, returns its ID. This is the same as the UniqueDIrectory for the system.
FBGetUserFilesPath(relativePath)		Given a relative path located under the user_files directory, say "hello.py", this returns the absolute path of the file.

Output/Debugging Functions

FBAddInfoMessage(string)	Adds an info messge to the WB message window
FBAddWarningMessage(string)	Adds a warning message to the WB message window
FBAddErrorMessage(string)	Adds an error message to the WB message window
FBMessageBox(string)	Pops up a dialog with the supplied string with an "ok" button.

Example Scenarios for Feedback Iterator

This section describes several scenarios for using the Feedback Iterator.

Set up Iteration with Feedback Iterator

User Breaks Iteration Control

Start an Iterative Update

Run a Single Iteration

Interrupt an Iterative Loop

Resume an Interrupted Iterative Loop

Modify any of the Systems Involved in Iteration (coupled clients)

Iterating to Convergence

Set up Iteration with Feedback Iterator

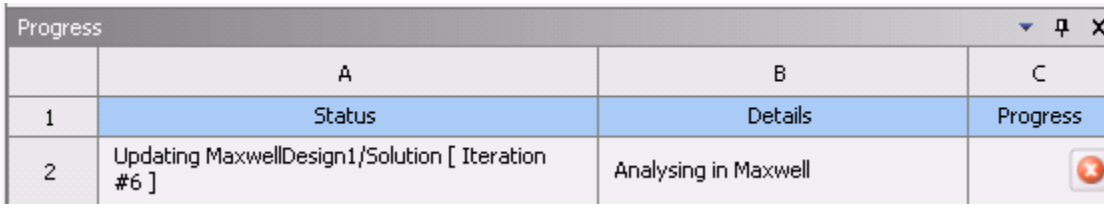
Scenario	Steps to execute	Outcome
Electromagnetics product solution provides solution data to a single downstream setup	• Drop Feedback Iterator system on the setup component of Electromagnetics product OR • Create Feedback iterator system and connect Electromagnetics product setup component to the TwoWay iterator component AND User clicks on the Iterator component and sets the desired Feedback Iterator properties.	Electromagnetics product setup component is connected to the Iterator. Electromagnetics product solution and downstream setup and solution components are coupled as clients to the iterator. The iterator component is the coupled master and the rest are coupled clients. Any coupling changes schematic visuals: • Coupled cells including coupling master are colored differently from normal cells • Coupled cell icons change to reflect the icon of the coupling master • Coupled client cells no longer display the **Update** context menu item and cannot be updated via script commands either. • The cell states of all coupled cells (master and clients) are synchronized to be the most pessimistic state among any of them. eg: after an update, any modification will set all of them to the "Modified" state.
Electromagnetics product solution provides to multiple downstream setup cells	same as above	All the downstream components (setup and solution in each downstream system) are coupled as clients. The update order in this case depends on any additional data-flow connections between the systems that consume the upstream Electromagnetics product solution data.
Electromagnetics product solution cell does not have any downstream targets	same as above	No changes occur beyond what is expected when you create a new connection (no coupling). Coupling only occurs when Electromagnetics product solution cell has a downstream connection. If a downstream connection is added after the link to the iterator, coupling will be performed as described in the first row.

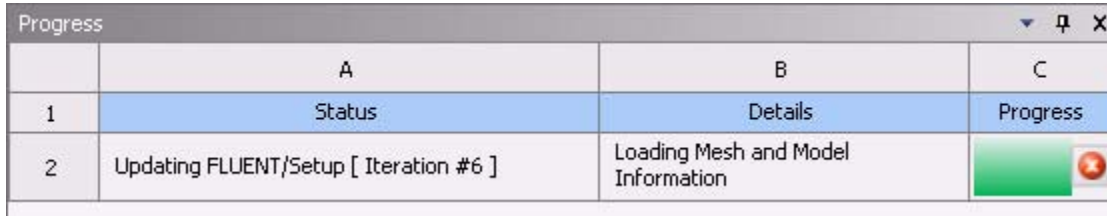
User Breaks Iteration Control

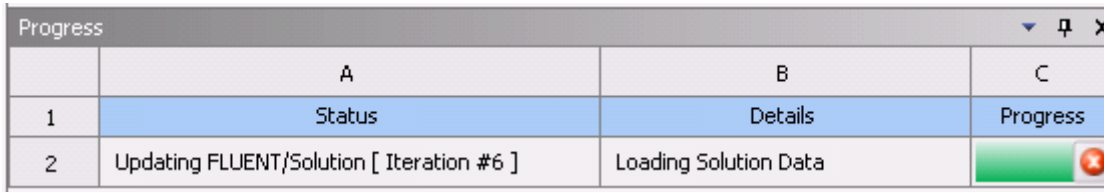
Scenario	Steps to execute	Outcome
Iterator coupled to Electromagnetics product solution cell and downstream setup/ solution cells	• Break Electromagnetics product solution cell's provides link -or- • Break Electromagnetics product setup link to iterator -or- • Delete Electromagnetics product solution consumer system -or- • Delete Iterator system	In addition to what is normally expected from the user actions, all coupled clients are de-coupled (their icons and colors on UI are restored)
Electromagnetics product setup cell is connected to iterator but Electromagnetics product solution does not provide anything. No components are coupled to the iterator	• Break Electromagnetics product setup link to iterator • Delete Iterator system	No coupling exists in this scenario so nothing visible changes beyond what is expected from the user actions

Start an Iterative Update

Scenario	Steps to execute	Outcome
Standard update scenario	• RMB on iterator component and select Update Gui operation -or- • Update Project -or- • Update all design points -or- • DX Update	For each iteration, the Electromagnetics product solution is updated followed by the downstream setup and the downstream solution. Each of the client component updates display their progress monitor and can be aborted (resulting in aborting the current iteration)







Run a Single Iteration

Scenario	Steps to execute	Outcome
NumCompleted < NumIterations	RMB on iterator component and select Single Iteration Update	One iteration is run. "Completed Iterations" property is incremented.
NumCompleted >= NumIterations	same as above	same as above

Interrupt an Iterative Loop

Currently, there is no special progress or interruption control. The progress monitor of the individual components is displayed as they are updated and any control they choose to provide (interrupt, abort or both) are available. If the user chooses to abort any of the client component updates, the current iteration is aborted and the Completed Iterations property remains unchanged from the previous iteration.

Resume an Interrupted Iterative Loop

Scenario	Steps to execute	Outcome
Single step	RMB on iterator component and select Single Iteration Update	One iteration is always run and the Completed iterations property is incremented.
Run till completion	• RMB on iterator component and select Update or- • Select the Project Update menu option from the toolbar etc.	If the user specified number of operations are already completed, nothing is done. Otherwise, the remaining iterations are run.

Modify any of the Systems Involved in Iteration (coupled clients)

When the user modified any of the coupled client components either in Workbench by changing some non-extrinsic property, design point etc or in an external editor, the coupled clients and master will be marked as modified. Additionally, the Completed Iterations property will be set to 0.

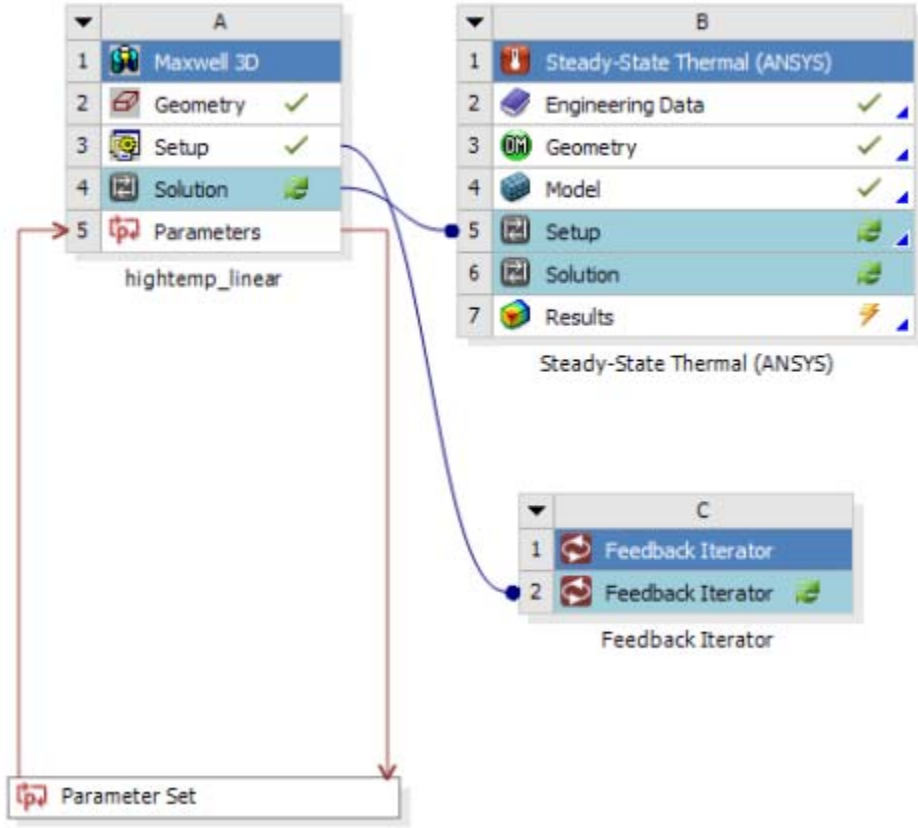
Maxwell in a thermal deformation use case

This is a more complex and involved use case. The feedback iterator works identically. This use case is explained in more detail in Maxwell Thermal Deformation

Iterating to Convergence

Scenario	Steps to execute	Outcome
Standard update scenario	• RMB on iterator component and select Update Gui operation -or- • Update Project -or- • Update all design points -or- • DX Update	For each iteration, the Electromagnetics product solution is updated followed by the downstream setup and the downstream solution. Each of the client component updates display their progress monitor and can be aborted (resulting in aborting the current iteration)

The following temperature convergence example uses a Feedback Iterator with a Maxwell design coupled with a Steady-State Thermal component. Iterating to convergence operates similarly for HFSS projects.



The example uses defaults of 5% for the convergence targets and a value of 100 max iterations. Recall that the max iterations value is set as a "safety" measure to ensure that iterations do not continue indefinitely if the solution does not converge. As we progress through the iterations, we observe the following:

End of Iteration #1

The first iteration, since it has no previous iteration, cannot return a meaningful delta value.

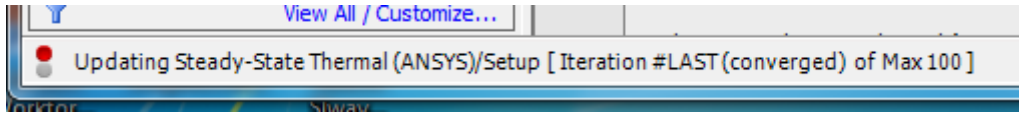
		started. (9:51:41 PM Jun 11, 2013)		
6	Informational	After iteration#: 1, Max temperature delta % = NaN, Max displacement delta % = NaN		6/11/2013 9:51:39 PM

At the end of this iteration, the Latest Delta value will still be listed as Not Available.

	A	B
1	Property	Value
2	General	
3	Component ID	FeedbackIterator
4	Directory Name	FeedbackIteratorComponent
5	Notes	
6	Notes	
7	Used Licenses	
8	Last Update Used Licenses	
9	Iterations	
10	Iterations Completed	1
11	Max Iterations	100
12	Callback	
13	Script	
14	Temperature Convergence	
15	Target Delta Temperature %	5
16	Latest Delta Temperature %	Not Available
17	Displacement Convergence	
18	Target Delta Displacement %	5
19	Latest Delta Displacement %	Not Available

Convergence Achieved

In this example, convergence has been achieved by the third iteration.



Properties of Schematic C2: Feedback Iterator

	A	B
1	Property	Value
2	General	
3	Component ID	FeedbackIterator
4	Directory Name	FeedbackIteratorComponent
5	Notes	
6	Notes	
7	Used Licenses	
8	Last Update Used Licenses	Not Applicable
9	Iterations	
10	Iterations Completed	4
11	Max Iterations	100
12	Callback	
13	Script	
14	Temperature Convergence	
15	Target Delta Temperature %	5
16	Latest Delta Temperature %	2.5
17	Displacement Convergence	
18	Target Delta Displacement %	5
19	Latest Delta Displacement %	2.5

When the convergence criteria are met, the simulation stops. If you attempt to do an update after convergence has been achieved, the simulation will not be launched since the current solutions satisfy the convergence criteria.

You can also monitor the iteration progress by opening the Maxwell editor and from the Results menu, select the Solution data dialog. Choose the Profile tab and keep the dialog open. As you solve each iteration, you can observe the reported Delta-T (with thermal feedback) and/or delta-displacement (with displacement feedback) and abort the iterations (or stop single iteration updates) when the values reach acceptable convergence levels

Profile | Convergence | Force | Torque | Matrix | Mesh Statistics

Task	Real Time	CPU Time	Memory	Information
Solver DRS	00:00:00	00:00:00	5 K	29 matrix, 0KB disk
Solver DRS	00:00:00	00:00:00	778 K	850 matrix, 0KB disk
adapt	00:00:01	00:00:01	32 M	551 tetrahedra
GenerateThermalInput	00:00:00	00:00:00	29 M	551 tetrahedra
				Maximum Absolute/Relative Delta Temperature = 12.882 kel, N/A

Surface Force Density in HFSS

HFSS can calculate surface force density for Work Bench coupling and for post processing within HFSS. The whole approach resembles Maxwell except that HFSS must calculate the surface force density at the post processing stage while solver provide the surface force density in Maxwell.

The surface force density formulation is provided in a technical note here. Surface forces may exist on surfaces where one side is conductor but the other is not, or finite conductivity and layered impedance boundary. The computation of surface force density will be performed at field geometry instance level.

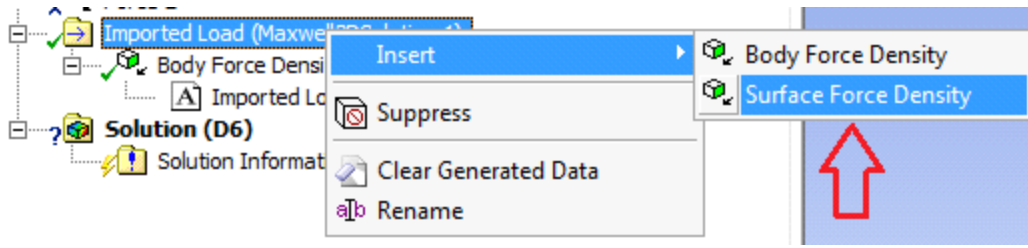
This feature is mainly for the purpose of mapping surface force density in HFSS to Work Bench Mechanical. We will also provide it for post processing within HFSS. This feature should work regardless of HFSS solution types as long as surface forces exist. Surface forces may exist when one side is conductor but the other is not, or finite conductivity and layered impedance boundary.

If an old coupling project happens to request surface force and is solved again, then the surface forces (likely nonzero) will be passed to mechanical side.

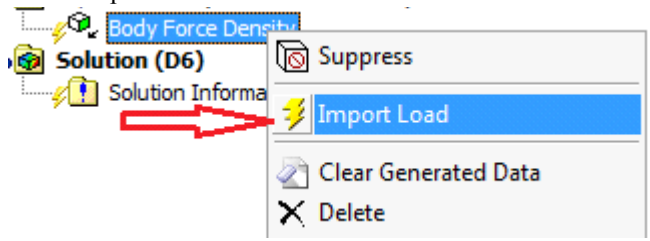
Mapping surface force density from HFSS to Work Bench Mechanical.

The behavior is the same as all current couplings.

1. From the Mechanical side set up a surface force coupling to an HFSS design similar to Maxwell.



2. Import the load.

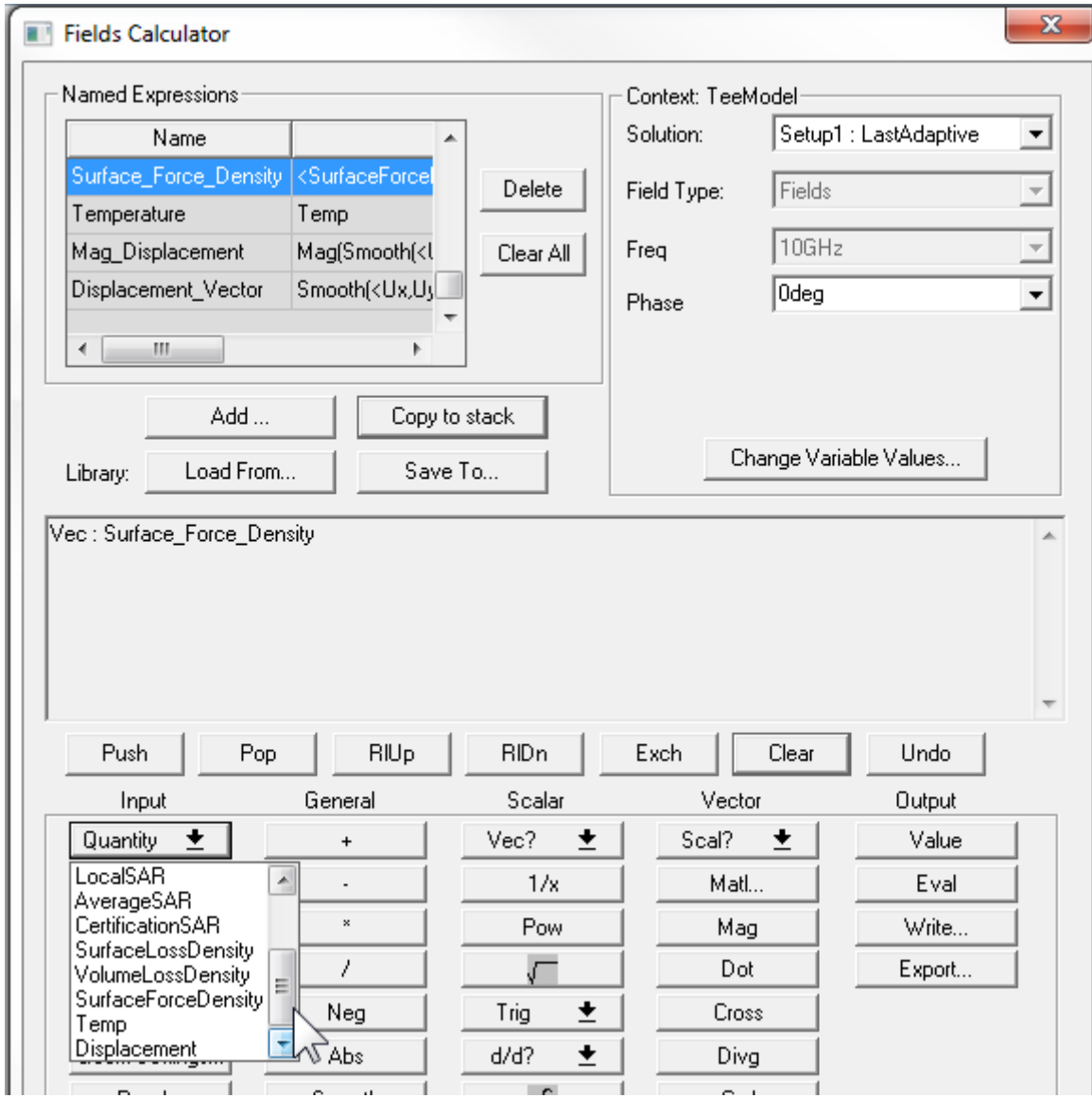


   Behind the scenes Mechanical will pass necessary information to HFSS, points on surface to request surface forces and auxiliary files, HFSS will then calculate the surface force densities and pass them back to Mechanical.

3. Post process surface density within HFSS over any surfaces where surface forces exist.

Surface Force Density will appear in the context menu in the Other category under Fields menu.

In the fields calculator Surface Force Density appears in the Named Expression list and the Quantity pull-down menu. So the user can use it the same way as other vector quantities.



22

Desktop Scripting with IronPython

This document describes IronPython briefly and then goes on to describe the desktop provided IronPython scripting console and scripting with IronPython. You can open an IronPython Command Window by clicking **Tools>Command Window**.

The document assumes that you know how desktop scripting works using VBScript or Javascript.

Introduction to IronPython

IronPython Mini-cookbook

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

Creating User Defined Primitives and User Defined Models in Python Scripts

Related Topics

User Defined Outputs: An Introduction

Introduction to IronPython

IronPython is an implementation of the Python programming language targeting the .NET runtime. What this means in practical terms is that IronPython uses the Python programming language syntax and standard python libraries and can additionally use **.**NET classes and objects to give one the best of both worlds. This usage of .NET classes is fairly seamless in that a class defined in a .NET assembly can be used as a base class of a python class.

Scope

Functioning as a tutorial on Python or IronPython is way out of the scope of this document. There are several excellent resources online that do a very good job in that regard. This document only attempts to provide a limited introduction to IronPython as used to script ANSYS EM products.

This document is also not a tutorial on the scripting of ANSYS EM products. It complements the existing scripting guide (available from a product's Help menu) and provides a pythonic interpretation of that information. The reader might have to refer to either the scripting guide or recorded samples of VBScript to follow some of the sections.

Python compatibility

The version of IronPython in use is **2.6.1** and built on the .NET framework version 2.0: this version targets **Python 2.6** language compatibility. While most python files will execute under IronPython with no changes, python libraries that make use of extensions written in the C programming language (NumPy or SciPy for instance), are not expected to work under IronPython. In such cases, it might be possible to locate .NET implementation of such libraries or explore the use of IronClad.

(http://code.google.com/p/ironclad/).

Advantages of IronPython

The advantages that IronPython use provides are significant:

- Python has a large eco-system with plenty of supporting libraries, Visual IDEs and debuggers. It is actively developed and enhanced.
- IronPython, in addition, has access to the entire .NET eco system. This allows us, for instance, to create a modern GUI using the **System.Windows.Forms** assembly from IronPython code and call any other .NET assembly for that matter.
- The use of IronPython's technologies enables the ability to interactively script Desktop (feature in development). This allows better discovery of the scripting APIs as well as directly programming to the scripting API in python, a language more tractable and platform independent compared with VBScript.
- The Python syntax of dictionaries is somewhat easier to read and write when supplying arguments to the scripting methods.

Related Topics

IronPython Mini-cookbook

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

IronPython Mini-cookbook

While a tutorial on Python syntax is beyond the scope of this document, it will present simple counterparts the VBScript constructs that users tend to regularly use.

Comments

VBScript	IronPython
' Comments start with a single quote ' like this line	# Comments start with a sharp or hash # symbol, like these lines

Assigning/Creating Variables

VBScript	IronPython
' Declare with a Dim Dim oDesktop **'Assignment needs a Set instruction** Set oDesktop = oApp.GetAppDesktop()	# No Set syntax. Simply create and assign oDesktop = oApp.GetAppDesktop()

Create Lists/Arrays

VBScript	IronPython
' Declare as array of String with 11 ' indices from 0 through 10 Dim myArray(0 to 10) as String myArray(0) = "Hello" myArray(1) = "bye" ' **Declare n array with no size** Dim array2() as String ' **Re-Dimension the array once size is** ' **known** ReDim array2(0 to 2) as String array2(0) = "this" array2(1) = "also"	# Declare an empty array myEmptyArray = [] # declare an array and initialize it with 5 ints myInitedArray = [1, 2, 3, 4, 5] # Python lists can have items of any type # and there is no pre-declaration # declare an array and init with mixed types mixed = ["hello", 1 ,2 ["nested"]] # append to an array mixed.append(3.5)

Create Dictionaries/Maps

VBScript	IronPython
' No direct equivalent is available as ' far as the author knows	# an IronPython dictionary is a collection of # name value pairs. Just like arrays, there is # no restriction on the keys or the values. # For purposes of ANSYS EM scripting however, # all keys must be strings # delimiters are curly braces # use a ":" between the key and the value # separate key value pairs with a "," myDict = { "a" : 1, "b":"hello there", "c" :[1, 2, "abc"] }

Boolean Values

VBScript	IronPython
' Boolean literals are **in lower case** true false	# The first letter is capitalized True False

Converting Numbers to Strings and Vice Versa

VBScript	IronPython
' Use **CInt, CDbl, CBool, CLng** ' to convert the string representation ' to **the number representation. Use** ' **IsNumber to check before conversion** Dim nStr = "100" Dim n = CInt(nStr) ' **Use CStr to convert a number to** ' its string representation Dim v, vStr v = 100 vStr = CStr(v)	# use the integer() or float() or double() # functions to cast a string CONTAINING the # string representation of whatever you are # casting to. strInt = "3" intVal = int(strVal) floatVal = float(strVal) # invoke the str() function with the int/float # values as needed. You can alternately use # the string formatting method listed below strVal = str(42) strVal = str(42.345)

String formatting/concatenation

VBScript	IronPython

' string concatenation uses the & ' operator Dim allStr, str1 str1 = " how are you" allStr = "Hello " & " There" & str1 ' there seems to be no direct string ' formatting function in VBScript ' using string concatenation or using ' Replace are the two **builtin options** Dim fmt = "{1} climbs stalk {2}" Dim str = Replace(fmt, "{1}", "jack") str = Replace(str, "{2"}, 10)	# if you have two strings, you can always # concatenate then using the '+' operator str1 = "hello" str2 = "world" str12 = str1 + " " + str2 # if you have different types though, string # and int say, you must use the string # formatting commands. When formatting # multiple arguments, they must be entered # as a tuple (item1, item2,) num = 10 str3 = "%s climbs stalk %d" % ("jack", num) str4 = "%d stalks" % num

Looping over lists

VBScript	IronPython
Dim myArray(0 to 2) as String myArray(0) = "alpha" myArray(1) = "bravo" myArray(2) = "charlie" For Each i in myArray Print i Next	vals = [1, 3, 3.456] def process(val): return 2*val # is of the form # for variable_name in array ':' # < indent> statement1 # <indent> statement2 for i in vals: print i print " -> " process(i)

Looping over a range

VBScript	IronPython
' Loop over a range, specify start, end ' and step For i = 0 To 10 Step 1 Print i Next	# prints out values from 0 through 9 for i in range(0, 10): print i

Related Topics

A Note About Indentation
Obtaining More Information
Discovering Methods
Help on a Method
Introduction to IronPython
Translating Script commands from VBScript to IronPython
Scripting Using Iron Python: Putting it all Together
IronPython Samples

A note about indentation

Python is one of the languages where whitespace (spaces, tabs etc) are syntactically significant. You must understand the basics of indentation before scripting in python.

Any statement that introduces a block of code should be written such that every line of the block has the same indent (leading spaces or tabs) and the indent should be at least one more than the indent of the introducing statement.

```
# define a function that starts at 0 indentation.
def multInt(a,b):
      # every line following the def multInt which is expected
      # to be a part of the function, must have the indent used
      # by the first line of the function (3 spaces)

      # here we introduce one more block, the if condition
      # each line that belongs to the body of this func should
      # have an indent that is more than the indent used by the
   # if statement
   If a%2 == 0:
      # I am using an indent 2 more than the parent. i.e. 5
      # spaces in total
      return (a * b) + 100
   else:
      return (a * b) + 1000
```

Sample Script 1: Describing python indentation

Related Topics

Obtaining More Information

Discovering Methods

Help on a Method

Introduction to IronPython

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

Obtaining more Information on Python

Reading a book and searching online are two of the standard options. There are several very good python tutorials online and the command window is also a great way to quickly execute code and learn by doing.

Much of the official python documentation is embedded into each python distribution and the command window can also be used to get more help. You can open the IronPython Command Window

by clicking Tools>Command Window. You can also aces the command window executable, **Ipy.exe**, from the **IronPython** directory under the ANSYS EM product installation directory. You can use this window to interactively enter python commands (any of the pure python samples in this document)

Related Topics

Discovering Methods

Help on a Method

Introduction to IronPython

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

Discovering Methods

For instance, if you want to list all methods available in the string module, you would type in "import string" to import the module first and then type in "dir(string)". You will get a list of all the methods available (as well as some __somename__ internal names that can be ignored)

```
>>> import string
>>> dir(string)
['Formatter', 'Template', '_TemplateMetaclass', '__builtins__', '__doc__', '__file__', '__name__',
'__package__', '_float', '_idmap', '_idmapL', '_int', '_long', '_multimap', '_re', 'ascii_letters',
'ascii_lowercase', 'ascii_uppercase', 'atof', 'atof_error', 'atoi', 'atoi_error', 'atol',
'atol_error', 'capitalize', 'capwords', 'center', 'count', 'digits', 'expandtabs', 'find',
'hexdigits', 'index', 'index_error', 'join', 'joinfields', 'letters', 'ljust', 'lower', 'lowercase',
'lstrip', 'maketrans', 'octdigits', 'printable', 'punctuation', 'replace', 'rfind', 'rindex', 'rjust',
'rsplit', 'rstrip', 'split', 'splitfields', 'strip', 'swapcase', 'translate', 'upper', 'uppercase',
'whitespace', 'zfill']
```

Related Topics

Obtaining More Information

Help on a Method

Introduction to IronPython

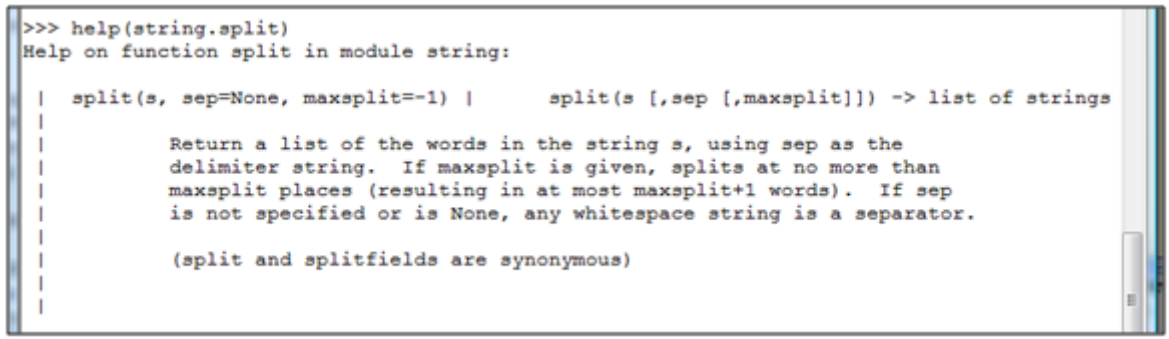
Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

Help on a Method

Once you know a function name (maybe you see something you like from the dir(string) listing above), you can get more help on it using the builtin **help** method. For instance, executing **help(string.split)** in the **IronPython Command Window** displays the following:



Related Topics

Obtaining More Information

Discovering Methods

Introduction to IronPython

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

Translating Script commands from VBScript to IronPython

This chapter briefly describes Desktop scripting methods and arguments via VBScript samples. The distinctions made here are significant and come in use later when translating scripts written in VBScript to IronPython.

Related Topics

Script Method Argument

Script method calls in VBscript are described later in this document, they generally take the form:

```
objectName . methodName ( arg1, arg2, ..)
```

The function call syntax is a standard followed by several programming languages, however, the argument types in VBScript objects, when used for product scripting, are restricted to the following

Primitive Types

Primitive types are the standard **bool, int, float, double** and **string**

Named Array

Named arrays are a special construct used very commonly and can be found in practically every recorded script sample.

It starts with **Array("NAME:someName**" and is followed by a collection of comma separated values which can be:

- A primitive value
- An array of primitive values
- Another named array
- A key, of the form "**keyName:="** followed by
  - A primitive value
  - A function (described next)

Named Function

Named functions are arrays which start with **Array(** and do not have a leading "NAME:name" item. **They are always introduced by a key** and can contain comma separated values of the following type:

- A primitive value
- A key (of the form "**keyName:=**") followed by
  - A primitive value
  - Another function (nested function)

Related Topics

Translating Script commands from VBScript to IronPython

VBscript Method Call Types

VBScript method calls fall into two categories and the distinction between the two results in syntax differences. These syntax differences are significant when converting VBScript to IronPython.

VBScript Functions

In VBScript terminology functions return values. The syntax for this is the one shared with practically all programming languages.

```
Set oDesktop = oAnsoftApp.GetAppDesktop()
Set oProject = oDesktop.NewProject
```

Sample Script 2: VBScript function call sample

Note that the significant item here is that the method name is **always** followed by an argument list enclosed in parentheses **if there are arguments**. If the argument list is empty as shown above for the *NewProject* call, the parentheses can be omitted.

VBScript Sub-Routines

VBScript Sub-Routines are those that do not have any return value. VBScript allows these to be written without any parentheses even if they have a non-empty argument list.

```
oModule.CreateReport "XY Plot1", "Standard", "XY Plot", "optimtee
: optimtee", _
   Array("Domain:=",  "Sweep"), Array("Freq:=", Array("All"),
   "offset:=",_
   Array("0uin")), Array("X Component:=",  "Freq", "Y Compo-
   nent:=", _
   Array("dB20(S(1,1))", "dB20(S(1,2))", "dB20(S(1,3))",  _
               "dB20(S(2,1))", "dB20(S(2,2))", "dB20(S(2,3))",
```

```
"dB20(S(3,1))",
        "dB20(S(3,2))",    "dB20(S(3,3))")), Array()
```

Sample Script 3: VBScript Sub-Routine sample

Related Topics

Translating Script commands from VBScript to IronPython

Converting VBScript Function calls to IronPython Syntax

When converting functions, the important point to remember is that IronPython function names, when used for scripting, are **always** followed by parentheses. So:

- If you see a VBScript snippet that looks like a VBScript Sub-Routine, remember to add parentheses.
- If you see a VBScript function that has no arguments and no parenthesis, remember to add them around an empty argument list.

The parentheses change is the only one to keep in mind when converting VBScript function calls syntax to IronPython.

Return Values

Primitive Method Arguments

Named Array Argument

Named Array Values with All Key Value Pairs

Named Arrays with Nested Named Arrays

Function Blocks

Return Values

VBscript return values are sometimes assigned via the **Set** declaration. IronPython return values are simple assignment (see the cookbook chapter)

Related Topics

Converting VBScript Function calls to IronPython Syntax

Primitive Method Arguments

Replace each VBScript primitive with an equivalent IronPython primitive. The main thing to notice here is that Boolean values in IronPython have their first letter capitalized.

```
True instead of true and False instead of false
```

Related Topics

Converting VBScript Function calls to IronPython Syntax

Named Array Argument

The recommended approach here is to simply replace a VBScript array with a python array. The mapping is quite simple:

- Change **Array(** to **[** and close with a **]** instead of the **)**
- Remove the line continuation symbols: **_**
- Map Boolean values correctly

oEditor.CreateCone Array("NAME:ConeParameters", "XCenter:=", "0mm", _
"YCenter:=", "0mm", "ZCenter:=", "0mm", "WhichAxis:=", "Z", "Height:=", "2mm",_ "Bottom-Radius:=", "1.56204993518133mm", "TopRadius:=", "0mm"), Array("NAME:Attributes", "Name:=", "Cone1", "Flags:=", "", "Color:=", _
"(132 132 193)", "Transparency:=", 0, "PartCoordinateSystem:=", _
"Global", "UDMId:=", "", "MaterialValue:=", _
"" & Chr(34) & "vacuum" & Chr(34) & "", "SolveInside:=", true)

Sample Script 4: Create cone recorded snippet

For instance, method and named VBScript arrays in the snippet above are translated to

```
oEditor.CreateCone(
    [
        "NAME: ConeParameters",
        "XCenter:=", "0mm",
        "YCenter:=", "0mm",
        "ZCenter:=", "0mm",
        "WhichAxis:=", "Z",
        "Height:=", "2mm",
        "BottomRadius:=", "1.56204993518133mm",
        "TopRadius", "0mm"
    ],
    [
        "NAME:Attributes",
        "Name:="  , "Cone1",
        "Flags:="    , "",
        "Color:="   ,  "(132 132 193)",
        "Transparency:=" , 0,
        "PartCoordinateSystem:=" , "Global",
        "UDMId:=" , "",
        "MaterialValue:=" , "\"vacuum\"",
```

```
        "SolveInside:="       , True
    ])
```

Sample Script 5: Create a cone in IronPython

Note that the formatting (which helps readability immensely) is not really needed. All that **had** to be done was

- Add the parentheses since the VBScript subroutine omits it
- Replace the **Array( )** delimiters with **[ ].**
- Remove the **Char(34)** function (which introduced a double quote) and replace it with the escaped double quote \" literal.
- Replace **true** with **True**
- Remove the line continuation symbol, _

Related Topics

Converting VBScript Function calls to IronPython Syntax

Named Array Values with All Key Value Pairs

While it is in general not allowed to replace the arrays and nested arrays with python dictionaries, in the case where the named array consists entirely of key value pairs (like the sample above), one can use a dictionary and **avoid typing the trailing ":=" symbols after the keys**. This further aids readability of the script.

- If all key value pairs
- Remove the trailing ":=" after each key
- Replace the "," after the key with a ":"
- If the named array is the top level argument, ensure that the "NAME:name" is present and is split into "NAME" : "name" as a key value pair
- Enclose the converted array in a { } pair to declare the dictionary.

```
oEditor.CreateCone(
    {
    "NAME" : "ConeParameters",
    "XCenter": "0mm",
    "YCenter": "0mm",
    "ZCenter": "0mm",
    "WhichAxis": "Z",
    "Height": "2mm",
    "BottomRadius": "1.56204993518133mm",
    "TopRadius": "0mm"
```

```
},
{
"NAME": "Attributes",
"Name": "Cone1",
"Flags": "",
"Color":  "(132 132 193)",
"Transparency": 0,
"PartCoordinateSystem": "Global",
"UDMId": "",
"MaterialValue" : "\"vacuum\"",
"SolveInside": True
}
```

Sample Script 6: CreateCone in IronPython using dictionary syntax

Related Topics

Converting VBScript Function calls to IronPython Syntax

Named Arrays with Nested Named Arrays

- Split the "NAME:name" field into a key value pair
- Translate array key value pair to a dictionary key value pair.
- Create a new key with the name of the nested array and keep the nested array (as an array or as a dictionary) as its value. If the nested array is being retained **as an array**, **the**"NAME**:name**" **field should be retained in the array**. If the nested array is being converted to a dictionary, the name is optional: if also retained in the nested array, it must match the outer key.

```
[ "NAME:name",
     "key1:=", 1,
   "key2:=", 2,
   ["NAME:name2", "R:=", 255]
]
```

Sample Script 7: Named array with nested named array in array syntax

The above named array with a nested named array (after conversion to IronPython as named array) can be converted to a dictionary as well. The dictionary can take any of the following forms

```
{ "NAME": "name",
     "key1": 1,
     "key2": 2,
```

```
        "name2": ["NAME:name2", "R:=", 255]
]
```

Sample Script 8: Named array with nested named array as mixed dictionary + array

```
{ "NAME": "name",
  "key1": 1,
  "key2": 2,
   "name2": {"R" : 255}
}
```

Sample Script 9: Named array with nested named array in all dictionary syntax

```
{ "NAME": "name",
  "key1": 1,
  "key2": 2,
   "name2": {
       "NAME" : "name2",
        "R"           : 255
     }
}
```

Sample Script 10: Nested named array with optional "NAME:" field

Related Topics

Converting VBScript Function calls to IronPython Syntax

Function Blocks

Function blocks in VBScript argument syntax are represented as arrays without the "NAME:.." field. However, functions are always introduced by a key in a parent structure. Function blocks can therefore never exist as a top-level argument. They are only found as the value pairs inside a named array or inside another function block.

Note Function blocks and their items cannot be converted to dictionaries even though they might be composed entirely of key value pairs.

The reason for this is the need to main the user-entered order. Every item in a function block is expect to be transmitted to the script method in exactly the same order as typed out and this is

impossible to achieve when a dictionary is used (as the keys get reordered according to the dictionary's internal tree/key sorting scheme).

Note When you see a function block, simply replace the Array( ) delimiters with the python array delimiters [ ]

Related Topics

Converting VBScript Function calls to IronPython Syntax

Scripting Using Iron Python: Putting it all Together

If you have existing VBScript/Javascript scripts use existing scripts them as much as possible by either embedding the test into the IronPython script or invoking them via the Run methods described later in this chapter.

Translating a script in VBScript to IronPython

Read the chapter on translation and study the samples in that chapter as well as those in the appendix. For python syntax and the differences, the mini-cookbook chapter will also be useful.

Writing an IronPython script from scratch

Read through the scripting guide available from the product's help menu and translate the VBScript methods described to IronPython using the information provided in the chapter on translation. Studying the samples in the document will also prove helpful.

For python syntax and the differences, the mini-cookbook chapter will also be useful.

IronPython Script Execution Environment

Scripting using Embedded VBScript or JavaScript

Scripting with IronPython

Related Topics

Introduction to IronPython

IronPython Mini-cookbook

Translating Script commands from VBScript to IronPython

Appendix: IronPython Samples

IronPython Script Execution Environment

Scripts written in IronPython are executed by desktop in four different ways:

- **Tools>Command Window**, to open the **IronPython Command Window:**

- **Tools > Run Script** menu item, select "IronPython" from the file type drop down
- Launch the product with a script argument, e.g., **HFSS -runscript someScript.py**
- Register an IronPython script as an external tool using the **Tools > External Tools** menu item.

When desktop executes a script, it does so in an execution environment setup with predefined variables and functions. These predefined variables and functions are how the script communicates with the desktop and they come in three flavors.

Script Argument for IronPython

Script Objects for IronPython

Methods for IronPython

Script Argument for IronPython

When scripts are launched using the **Tools > Run Script** menu item, the dialog that pops up allows the user to specify arguments.

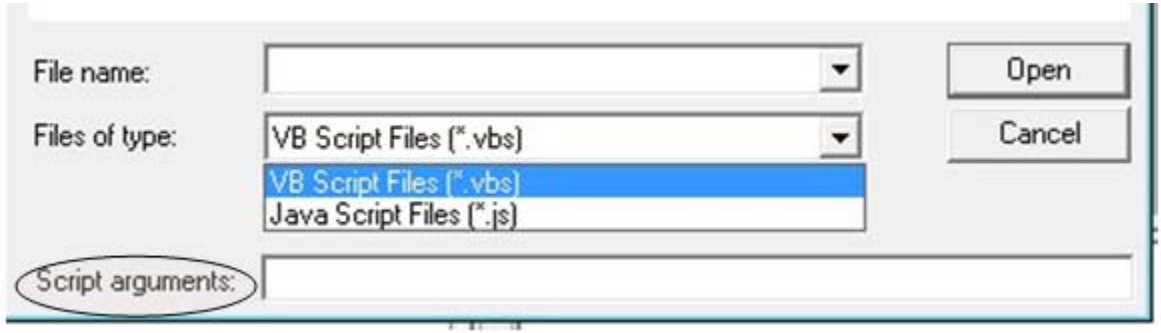


Figure 1: Run Script dialog and script arguments

Any argument specified here is communicated to the script being executed as the predefined variable **ScriptArgument**.

Related Topics

IronPython Script Execution Environment

Script Objects for IronPython

The Desktop script objects are available to every IronPython script being run. The names used follow the same convention as with VBScript.

- **oAnsoftApplication** is the script object representing the entire application.
- **oDesktop** is the script object representing the desktop. While this can be obtained from the oAnsoftApplication using *oAnsoftApplication.GetAppDesktop()*, since this is a very commonly used script object, it is always made available.

The use of these objects is done as per the individual product scripting guides with some syntax adjustments made for the parameters to account for the IronPython differences.

Related Topics

IronPython Script Execution Environment

Methods for IronPython

A collection of methods is made available to all python scripts to enable the use of existing scripts written in VBScript or Javascript. Additional utility methods to add messages to the desktop message window are also present.

- **AddErrorMessage(str), AddWarningMessage(str), AddInfoMessage(str) and AddFatalMessage(str)** this family of methods allows a script to add messages to the product's message window. All methods take a python string.
- **RunScriptFile(filename)** runs the contents of the supplied file as a script. The type of the script (Javascript/VBScript) is deduced from the file extension.
- **SetScriptingLanguageToJavascript(), SetScriptingLanguageToVBScript()** allows the script to change the syntax of the language being used in the *Run*Command* commands listed below. Once set, the language choice remains for the rest of the script execution or till it is changed. The default is *VBScript*.
- **RunScriptCommand(*CommandString*)** runs the supplied script text. The language of the text is controlled by a previous call to *SetScriptingLanguageToJavascript()* or *SetScriptingLanguageToVBScript()*
- **RunDesktopCOmmand**(***CommandString***) runs the supplied script text. The text is expected to generally be a single line and be a method call on a oDesktop object. The oDesktop object is initialized based on the supplied optional ProjectName Pparameter. If the ProjectName parameter is not supplied, the active project is used.
- **RunProjectCommand(*CommandString,ProjectName=None, ProjectName=None*)** runs the supplied script text. The oDesign object is initialized based on the supplied optional parameters. If the ProjectName is not supplied, the active project is used to resolve the project. If the DesignName is not supplied, the project's active design or first design will be used to resolve the oDesign object within the project.
- **RunModuleCommand(*CommandString,ModuleName, DesignName=None, ProjectName=None*)** runs the supplied script text. The text is expected to be a single line and be a method call on a oModule object. The design targeted is resolved using the supplied optional DesignName and ProjectName parameters and the oModule object is initialized using the supplied Module name on that resolved design.
- **RunEditorCommand(*CommandString, EditorName,DesignName=None, ProjectName=None*)** runs the supplied script text. The text is expected to be a single line and be a method call on a oEditor object. The design targeted is resolved using the supplied optional DesignName and ProjectName parameters and the oEditor object is initialized using the supplied Editor name on that resolved design.

- **RunDefinitionManagerCommand(*CommandString, ProjectName=None*)** runs the supplies script text. The text is expected to be a single line and be a mthod call on a oDefinitionManager object. The oDefinitionManager object is created for the project specifie4d by the optional ProjectName parameter.

Related Topics

IronPython Script Execution Environment

Scripting using Embedded VBScript or JavaScript

Since script recording is still done in VBScript and users are expected to have a significant collection of VBScript or JavaScript assets, it is useful to continue to use existing script files and snippets even when scripting in IronPython. The various **Run<*>Command** methods have been designed for this purpose.

For instance: one can create a parameterized cone in HFSS by executing the following IronPython script from the **Tools>Run Script** menu.

```
# assign the VBScript snippet obtained from a script recording from
HFSS to
# coneScript and replace the BottomRadius recorded value with
botRadius
coneScript = """Dim oAnsoftApp
Dim oDesktop
Dim oProject
Dim oDesign
Dim oEditor
Dim oModule
Set oAnsoftApp = CreateObject("AnsoftHfss.HfssScriptInterface")
Set oDesktop = oAnsoftApp.GetAppDesktop()
oDesktop.RestoreWindow
Set oProject = oDesktop.GetActiveProject()
oProject.InsertDesign "HFSS", "HFSSPyTestDesign", "DrivenModal",
""
Set oDesign = oProject.SetActiveDesign("HFSSPyTestDesign")
Set oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateCone Array("NAME:ConeParameters", _
  "XCenter:=", "0mm", "YCenter:=", "0mm", "ZCenter:=", "0mm",_
  "WhichAxis:=", "Z", "Height:=", "2mm",  _
  "BottomRadius:=",  "3mm", _
 "TopRadius:=", "0mm"), Array("NAME:Attributes", "Name:=",  _
```

```
  "Cone1", "Flags:=", "", "Color:=", "(132 132 193)", "Transpar-
ency:=", 0, _
  "PartCoordinateSystem:=",  "Global", "UDMId:=", "", "Material-
Value:=", _
  "" & Chr(34) & "vacuum" & Chr(34) & "", "SolveInside:=",  _
  true)
"""

SetScriptingLanguageToVBScript()
RunScriptCommand(coneScript)
```

Sample Script 11: Hybrid VBScript + IronPython scripting: parameterized Cone Creation

Even though recorded VBScript is used for scripting, the incremental functionality that is provided using IronPython is the ability to write a GUI using IronPython/.NET, collect information from the user and then modify or generate the VBScript commands to actually script the ANSYS EM desktop. This GUI functionality is cross platform and a significant positive. The following example demonstrates a contrived use of a .NET window form to display the argument supplied to the IronPython script (via the ***ScriptArgument*** variable).

```
#import the CLR references
import clr
clr.AddReference("System.Windows.Forms")

from System.Windows.Forms import Application, Form, Label,
Button,  DockStyle

# the GUI form to show some text
# the class below derives from From (System.Windows.Forms.Form)
# imported above from the .NET assembly.
class ShowPropertiesForm(Form):
   def __init__(self, name, text):
      self.Name = name
      self._label = Label()
      self._label.Text = text
      self._label.Dock = DockStyle.Fill

      _button = Button()
```

```
            _button.Text = "Close"
            _button.Dock = DockStyle.Bottom
            _button.Click += self._buttonPressed

            self.Controls.Add(self._label)
            self.Controls.Add(_button)

        def _buttonPressed(self,sender,args):
            self.Close()

#-----------------------------------------------------
# Main script code
#-----------------------------------------------------
#display the ScriptArgument variable as the text label
# in the form.
gui = ShowPropertiesForm("Sample Form", ScriptArgument)

# This makes it a modal dialog.
gui.ShowDialog()

# the following will make it a non-modal dialog
#Application.Run(gui)
```

Sample Script 12: Demonstrates the use of a .NET form from IronPython

While creating cross platform user interfaces from scripts is one of the main motivations driving the adoption of IronPython, any .NET assembly can be used with the caveat that Linux use requires Mono compatibility of any used assemblies.

While this hybrid approach is useful when you have existing VBScript commands that you want to reuse or when you want to quickly parameterize a recorded sample, the one significant limitation of this approach is the inability to capture return values from VBScript or JavaScript calls that do return something. Full two way communication with the product requires the use of pure IronPython to directly invoke the script objects as described below.

Related Topics

IronPython Script Execution Environment

Scripting with IronPython

While this section talks about directly interacting with the script objects, note that you can execute VBScript or Javascript at any point using any of the available Run*Command functions. using your

existing script assets in this fashion and mixing with IronPython code for new functionality as needed is a viable and option.

Access to the application scripting objects is provided via the predefined **oDesktop** object (as listed in Script Objects). Interacting with the script objects is very natural, method calls are made just like in VBScript except that the argument syntax is somewhat simplified to follow natural Python syntax. All primitive types (string, integer, double) map to the natural primitive types in python. The only differences from the VBScript syntax are seen when specifying array type arguments. The differences are described in earlier chapters.

Note The typical VBScript calls to obtain the registered COM scripting interface via CreateObject calls and then obtain the oDesktop object from it using the GetAppDesktop() is not needed (or even supported on all platforms). Since all scripting occurs in the context of a running workbench, the available Desktop object is always provided and expected to be used directly.

Scripting using the IronPython scripting API is very much like scripting with VBScript except that

- Any argument is supplied via the built in **ScriptArgument** variable
- The **oDesktop** object is always available
- The scripting method names are identical to the ones used with VBScript
- Method calls, while the name is the same have to adhere to the rule of ensuring trailing parentheses irrespective of whether the function returns anything or has any arguments.
- Any compound/block arguments should be translated to the appropriate IronPython array or dictionary syntax.

The samples section lists a collection of pure IronPython snippets: these, along with the various script snippets listed in this document should serve as a guide and reference.

Related Topics

IronPython Script Execution Environment

IronPython Samples

Change property

The following snippets show how a change property command (in this case, to change the color of a cone) looks in VBScript and its two possible IronPython variants.

```
oEditor.ChangeProperty Array("NAME:AllTabs",
Array("NAME:Geometry3DAttributeTab",_
   Array("NAME:PropServers",   "Cone1"), _
   Array("NAME:ChangedProps", _
   Array("NAME:Color", "R:=", 255, "G:=", 255, "B:=",   0))))
```

Sample Script 13: ChangeProperty command to change color of a cone in VBScript

```
oEditor.ChangeProperty(
        ["NAME:AllTabs",
   ["NAME:Geometry3DAttributeTab",
   ["NAME:PropServers",  "Cone1"],
   ["NAME:ChangedProps",
   ["NAME:Color", "R:=", 0, "G:=", 0, "B:=", 64]
   ]
]
      ])
```

Sample Script 14: ChangeProperty command to change color of cone using Python arrays

Any time there are named arrays composed purely of key-value pairs, they can always be represented using a Python dictionary, irrespective of the nesting of said named array.

```
oEditor.ChangeProperty(
   ["NAME:AllTabs",
   ["NAME:Geometry3DAttributeTab",
      ["NAME:PropServers",  "Cone1"],
      ["NAME:ChangedProps",
      {
        "NAME":"Color",
         "R"  : 0,
           "G"  : 64,
                 "B"  : 0
```

```
    }]]
])
```

Sample Script 15: ChangeProperty command to change the color of a cone using Python arrays and dictionaries

Create a Cone using IronPython

Most scripting tasks using IronPython are expected to be formatted as the example below. One starts with the predefined **oDesktop** object and drills down to the design, editors, modules etc and issues any required commands on the object while formatting the script command arguments in natural python syntax.

```
oProject = oDesktop.GetActiveProject()
oDesign =
oProject.InsertDesign("HFSS","Random","DrivenModal","")
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateCone(
{
   "NAME" : "ConeParameters",
   "XCenter": "0mm",
   "YCenter": "0mm",
   "ZCenter": "0mm",
   "WhichAxis": "Z",
   "Height": "2mm",
   "BottomRadius": "1.56204993518133mm",
   "TopRadius": "0mm"
   },
   {
   "NAME": "Attributes",
   "Name": "Cone1",
   "Flags": "",
   "Color": "(132 132 193)",
   "Transparency": 0,
   "PartCoordinateSystem": "Global",
   "UDMId": "",
   "MaterialValue" : "\"vacuum\"",
   "SolveInside": True
   }
```

```
    )
```

Sample Script 16: IronPython script to create a cone

Create geometry and then create a grid from it using copy/paste/move

The following script demonstrates slightly more advanced use of scripting and the use of return values from script methods. It creates a 5x5 grid of cones and also demonstrates the adding of information messages to the application's message window.

```
oProject = oDesktop.GetActiveProject()
oDesign = oProject.InsertDesign("HFSS","Hersheys
Kisses","DrivenModal","")
oEditor = oDesign.SetActiveEditor("3D Modeler")



# create the first cone
AddInfoMessage("Creating first cone")
firstConeName = "firstCone"
coneBotRad = "1.5mm"
oEditor.CreateCone(
   {
   "NAME" : "ConeParameters",
   "XCenter": "0mm",
   "YCenter": "0mm",
   "ZCenter": "0mm",
   "WhichAxis": "Z",
   "Height": "2mm",
   "BottomRadius": coneBotRad,
   "TopRadius": "0mm"
   },
   {
   "NAME": "Attributes",
   "Name": firstConeName,
   "Flags": "",
   "Color":  "(132 132 193)",
   "Transparency": 0,
   "PartCoordinateSystem": "Global",
   "UDMId": "",
```

```
   "MaterialValue" : "\"vacuum\"",
   "SolveInside": True
   }
   )

# Now replicate this a few times and create an array out of it
AddInfoMessage("Replicating it 24 times")
for x in range(5):
   for y in range(5):
         # leave the first one alone in it's created
         # position
         if x == 0 and y == 0:
            continue

# all other grid positions, replicate from the
# first one

# copy first
oEditor.Copy(
   {
      "NAME": "Selections",
      "Selections": firstConeName
   }
)

# paste it and capture the pasted name
# the pasted names come in an array as we could
# be pasting a selection cmposed of multiple objects
pasteName = oEditor.Paste()[0]

# now move the pasted item to it's final position
oEditor.Move(
   {
      "NAME": "Selections",
      "Selections": pasteName
```

```
        },
        {
            "NAME": "TransalateParameters",
            "CoordinateSystemID": -1,
            "TranslateVectorX": "%d * 3 * %s" % (x, coneBotRad),
            "TranslateVectorY": "%d * 3 * %s" % (y, coneBotRad),
            "TranslateVectorZ": "0mm"
        }
    )


    # Now fit the display to the created grid
    oEditor.FitAll()
```

Sample Script 17: Sample script to create a cone and then use copy/paste/move to replicate it.

Related Topics

Introduction to IronPython

IronPython Mini-cookbook

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

Creating User Defined Primitives and User Defined Models in Python Scripts

You can create User Defined Primitives and User Defined Models in Python scripts (based on the IronPython implementation).

Advantages Compared to C++

- No need to create and build project; all you need to do is create a Python script
- Python script is platform independent
- Scripts can inherit functionality from existing scripts
- Garbage collector - no need to free memory
- Easy debugging

Changes compared to C

Though methods, constants and structures are kept as close to the C implementation as possible, some changes had to be made to make code Python-compatible.

Structures

- Structures have the same names as in C implementation.
- Structures fields names are capitalized.
- Arrays in structures become lists in Python (Technically a.NET IList container)
- Structure instances are created using the supplied constructors and members are accessed using the provided access methods.

For a complete list of structures and examples please see UDP/UDM Structures.

Constants

Enumeration/Enum constants have almost the same names as in C but the enum must be qualified by the type. Additionally, redundant "UDP", "UDM" or type prefixes have been removed. This allows for better human-readability.

```
# Example of specifying the LengthUnit enum by qualifying it
# with the type of the enum: UnitType
unitType = UnitType.LengthUnit
```

For a complete list of enum constants please see UDP/UDM Constants.

Methods

Methods are described in IUDPExtension methods, IUDMExtension methods, UDPFunctionLibrary and UDMFunctionLibrary listed further in this document.

The main differences in functions parameters (from C implementation):

- functions names in UDPFunctionLibrary and UDMFunctionLibrary are capitalized
- arrays become a python list of objects
- `void * callback parameter` is dropped from the parameter list

- output parameters (pointer types that are filled during the function call) usually become return values
- 'list size' parameter usually will be omitted as redundant

Output Parameters

The rule for the output parameters is as follows:

- If the function has one output parameter variable and no return value, the variable will become function's return value.The same will happen if the return value is a 'success/failure' boolean ('None' will be returned on failure and parameter variable - on success).
- If the function has one output parameter and a return value, the function will return a Python tuple where function return value will be the first one in the tuple.
- If there is more than one out variable, the function will return a Python tuple with all output parameters in the specified order. If function has a return value, it must always be the first in the tuple.

```
# one output parameter; return value is ignored
udmDefinition = udmFunctionLibrary.GetDefinition()

# one output parameter; return value must be preserved. return
# and output values are packed into the return tupe, in order
(lRet, partIdsList) = udpFunctionLibrary.DetachFaces(nPartIds,
faceIdsList)

# Two output parameter; return value must be preserved
# the return tuple is (returnVal, output1, output2)
(bRet, udpPositionLow, udpPositionHigh) =
udmFunctionLibrary.GetBoundingBox(partId,exact);
```

Comparison with C function:

C	Python
bool getDefinition(UDMDefinition* udmDefinition, void* callbackData); where udmDefinition is an output parameter	udmDefinition = udmFunctionLibrary.GetDefinition() (Note: callbackData is omitted in py interface)
long detachlFaces(int nFacesAndPartIds, long* faceIds, long* partIds, void* callbackData); where partIds is an output parameter	(bRet, partIds) = udmFunctionLibrary.DetachlFaces(nFacesAndPartIds, faceIds) (Note: callbackData is omitted in py interface)

'List Size' Parameters

The rule for the 'list size' is as follows:

- If function has input 'List' parameter and input 'list size' parameter, 'list size' parameter will be omitted.
- If function has output 'List' parameter and output 'list size' parameter, 'list size' parameter will be omitted.
- If function has output 'List' parameter and input 'list size' parameter, 'list size' parameter won't be omitted as it's needed for memory allocation in the corresponding C++ function from the UDP/UDM function library.

Example:

```
# input list, input list size
lret = udpFunctionLibrary.Unite(objectIds)

# output list, output list size
faceIdList = udmFunctionLibrary.GetAllFaces(PartId)

# output list, input list size
(lret, partIdList) = udpFunctionLibrary.DetachFaces(listSize,
faceIdList)
```

Comparison with C function:

C	Python
bool getAllFaces(long partId, long* numFaces, long** faceIds, void* callbackData); where numFaces and faceIds are output parameters and numFaces is the size of faceId.	faceIds = udmFunctionLibrary.GetAllFaces(partId) (ignore numFaces as redundant: folded into faceIds, return value is omitted: folded into the faceIds is None check callbackData is omitted)
long unite(long numObjects, long* objectIds, void* callbackData); where numObjects and objectIds are input parameters and numObjects is the size of objectIds.	lret = udpFunctionLibrary.Unite(objectIds) (ignore numObjects as redundant: folded into objectIds callbackData is omitted)
long detachFaces(long nSize, long* faceIds, long* partIds, void* callbackData); where partIds is and output list and nSize is an input parameters and nSize is the size of partIds.	(lret, partIdList) = udpFunctionLibrary.DetachFaces(nSize, faceIds) (nSize is not ignored, callbackData is omitted)

Added Parameters

There is a special case in UDPFunctionLibrary: two functions - DuplicateAlongLine and DuplicateAroundAxis - have new integer listSize parameter added to their signatures.

This parameter defines the size of the output List. This is done for compliance with C++ geometry library as the size of the List must be predefined and this size is different from the existing parameter's values.

Example:

```
(ret, cloneIDs) = funcLib.DuplicateAlongLine(partID, transVec,
numCubes, cloneIdsSize)
(ret, cloneIDs) = funcLib.DuplicateAroundAxis(partID, axis,
angle, nClones, cloneIdsSize)
```

Here cloneIdsSize is a new integer parameter.

Comparison with C function:

C	Python
long duplicateAlongLine(long partId, UDPVector transVector, int nClones, long* nClones, void* callbackData);	(lret, cloneIds) = udmFunctionLibrary.DuplicateAlongLine(partId, transVec, nClones, cloneIdsSize) (callbackData is omitted cloneIdsSize is a new parameter)
long duplicateAroundAxis(long partId, UDPCoordinateSystemAxis axis, double angle, int nClones, long* nClones, void* callbackData);	(lret, cloneIds) = udmFunctionLibrary.DuplicateAroundAxis(partId, axis, angle, nClones, cloneIdsSize) (callbackData is omitted cloneIdsSize is a new parameter)

Developing a UDM/UDP

Creation

To create a User Defined Primitive in Python you write a Python script that implements UDPExtension class. To create a User Defined Model in Python you write a Python script that implements UDMExtension class (see links for full description).

Location

The scripts are located the same way the C based UDM/UDP are. They are expected to be under the UserDefinedParts or UserDefinedModels sub-directories of one of the library folders (SysLib, UserLib or PersonalLib). They will then appear under the appropriate menu items: **Draw>User Defined Primitives for UDP** or **Draw>User Defined Model for UDM**.

The sub-directories structure created in one of the specified directory will be displayed in the UDP/UDM menu.

Keep in mind that there is no difference between the menu display for C and Python implementations of UDM or UDP - only the file names without extensions are displayed

Organize

"Lib" sub-directory is a special directory. The contents of this directory is not shown in the menu. In the "Lib" directory you can create Python scripts with base classes and utilities to be used in UDP/UDM Python scripts. All the Lib directories upstream of a script (till the UserDefinedModels or UserDefinedPrimitives) are included in the Python search path and this allows for easy import of helper modules in such directories.

To use UDM data structures, constants, and/or classes in your Lib sub-directory scripts you have to add import statement to the scripts:

For UDM:extension:

```
from UDM import *
```

For UDP:extension:

```
from UDP import *
```

Edit/Reload

Python is a scripting language, so if you have errors in your script, you will see them at the time you try to run the script. The errors will be displayed in the Message Manager Window. If you need more information, you might be able to get it from log files. See Logging section for more details.

You can always change your script, call **Update Menu** command from **Draw>User Defined Model> menu** or **Draw>User Defined Primitives> menu** and run the script again. If you delete script you might want to restart the application instead of calling **Update Menu**.

UDPExtension

Import

You do not have to add import statements for the predefined classes, structures, and constants - it is done for you and all data types described in this document can be used in your Python script.

However you have to add import statements to your helper scripts in your Lib sub-directory.

```
from UDP import *
```

Main class: UDPExtension

You must write a class derived from IUDPExtension with a mandatory name UDPExtension:

```
class UDPExtension(IUDPExtension):
```

The class should implement IUDPextension methods described below.

IUDPExtension methods

All methods are same as the methods in the C UDP implementation. The changes to the methods signatures are just to conform to the Python style.

Mandatory methods.

These methods must be implemented in the UDP Python script as methods of UDPExtension class.

GetLengthParameterUnits()

- returns string.

GetPrimitiveTypeInfo()

- returns UDPPrimitiveTypeInfo.

GetPrimitiveParametersDefinition2()

- returns a list of UDPPrimitiveParameterDefinition2 or None on failure

AreParameterValuesValid2(errorMsg, udpParams)

- errorMsg is a c# list of strings
- udpParams is a c# list of UDPParam
- returns True if udpParams are valid, False otherwise.

CreatePrimitive2(funcLib, udpParams)

- funcLib is UDPFunction library
- udpParams is a c# list of UDPParam
- returns True on success, False on failure.

Optional methods

These methods, which have default implementations, can be implemented as methods of UDPExtension class as needed. Default methods will return NULL or FALSE depending on the return type.

GetPrimitiveParameters()

- returns Python list of strings or NULL

GetRegisteredFaceNames()

- returns Python list of strings or NULL

GetRegisteredEdgeNames()

- returns Python list of strings or NULL

GetRegisteredVertexNames()

- returns Python list of strings or NULL

ProjectParametersOnToValidPlane2(currentUDPParams, projectedUDPParams)

- currentUDPParams is a list of UDPParam
- projectedUDPParams is a list of UDPParam
- returns True on success, False on failure.

MapParametersDefinitionVersions2(oldVersion, oldUDPParams)

- oldVersion is a string
- oldUDPParamsis a list of UDPParam
- returns Python list of UDPParam or NULL

GetOldPrimitiveParametersDefinition2(version)

- version is a string
- returns a list of UDPPrimitiveParameterDefinition2 or None on failure.

Example UDP

```
import sys
class UDPExtension(IUDPExtension):
```

```
def GetLengthParameterUnits(self):
     return "mm"

def GetPrimitiveTypeInfo(self)
     typeInfo = UDPPrimitiveTypeInfo(
           name = "SampleUDP",
           purpose = "example",
           company="ANSYS",
           date="12.21.12",
           version = "1.0")

      return typeInfo
...
...
```

For the full example, see Example Scripts for Python UDP and UDM.

UDPFunctionLibrary

UDPFunctionLibrary implements IUDPFunctionLib interface: The IUDPFunctionLib object is passed as a parameter to Python script in functions **CreatePrimitive2**. You can call any of the functions from the functions list (shown below).

```
udpFunctionLib.AddMessage(MessageSeverity.WarningMessage ,"New warning");
```

Considering that **out** variables are returned as members of the return tuple, sample code that calls DetachFaces in Python script can look like this:

```
intVal = 10
inList = [..., ...]
(lret, outList) = udpFunctionLibrary.DetachFaces(intVal, inList)
```

As you can see *outList* output parameter is defined in the call to DetachFaces function. Unlike in C, there is no need to define it before the function call.

Functions list:

1. ***bool:*** bret = **AddMessage**(***MessageSeverity:*** messageSeverity, ***string:*** message)
2. ***int:*** iret = **NameAFace**(***UDPPosition:*** pointOnFace, ***string:*** faceName)
3. ***int:*** iret = **NameAEdge**(***UDPPosition:*** pointOnEdge, ***string:*** edgeName)

4. ***int:*** iret = **NameAVertex**(***UDPPosition:*** pointOnVertex, ***string:*** vertexName)
5. ***int:*** iret = **GetFaceIDFromPosition**(***UDPPosition:*** pointOnFace)
6. ***int:*** iret = **GetEdgeIDFromPosition**(***UDPPosition:*** pointOnEdge)
7. ***int:*** iret = **CreatePolyline**(**UDPPolylineDefinition:** polylineDefinition)
8. ***int:*** iret = **CreateRectangle**(***CoordinateSystemPlane:*** whichPlane, ***UDPPosition:*** centerPoint, ***List_of_double:*** widthAndHeight, ***int:*** isCovered)
9. ***int:*** iret = **CreateArc**(***CoordinateSystemPlane:*** whichPlane, ***UDPPosition:*** centerPoint, ***UDPPosition:*** startPoint, ***double:*** fAngle)
10. ***int:*** iret = **CreateCircle**(***CoordinateSystemPlane:*** whichPlane, ***UDPPosition:*** centerPoint, ***double:*** fRadius, ***int:*** isCovered)
11. ***int:*** iret = **CreateEllipse**(***CoordinateSystemPlane:*** whichPlane, ***UDPPosition:*** centerPoint, ***double:*** fMajorRadius, ***double:*** fRadiusRatio, ***int:*** isCovered)
12. ***int:*** iret = **CreateRegularPolygon**(***CoordinateSystemPlane:*** whichPlane, ***UDPPosition:*** centerPoint, ***UDPPosition:*** startPoint, ***int:*** numOfSides, ***int:*** isCovered)
13. ***int:*** iret = **CreateEquationBasedCurve**(***UDPEquationBasedCurveDefinition:*** curveDefinition)
14. ***int:*** iret = **CreateEquationBasedSurface**(***UDPEquationBasedSurfaceDefinition:*** surfaceDefinition)
15. ***int:*** iret = **CreateSpiral**(***UDPSpiralDefinition:*** spiralDefinition)
16. ***int:*** iret = **CreateBox**(***UDPPosition:*** startPoint, ***List_of_double:*** boxXYZsize)
17. ***int:*** iret = **CreateSphere**(***UDPPosition:*** centerPoint, ***double:*** fRadius)
18. ***int:*** iret = **CreateCylinder**(***CoordinateSystemAxis:*** whichAxis, ***UDPPosition:*** centerPoint, ***double:*** fRadius, ***double:*** fHeight)
19. ***int:*** iret = **CreateCone**(***CoordinateSystemAxis:*** whichAxis, ***UDPPosition:*** centerPoint, ***double:*** fBottomRadius, ***double:*** fTopRadius, ***double:*** fHeight)
20. ***int:*** iret = **CreateTorus**(***CoordinateSystemAxis:*** whichAxis, ***UDPPosition:*** centerPoint, ***double:*** fMajorRadius, ***double:*** fMinorRadius)
21. ***int:*** iret = **CreatePolyhedron**(***CoordinateSystemAxis:*** whichAxis, ***UDPPosition:*** centerPoint, ***UDPPosition:*** startPosition, ***int:*** numOfSides, ***double:*** fHeight)
22. ***int:*** iret = **CreateHelix**(**UDPHelixDefinition:** helixDefinition)
23. ***int:*** iret = **Unite**(***List_of_int:*** pObjectIDArray)
24. ***int:*** iret = **Subtract**(***List_of_int:*** pBlankObjectIDArray, ***List_of_int:*** pToolObjectIDArray)
25. ***int:*** iret = **Intersect**(***List_of_int:*** pObjectIDArray)
26. ***int:*** iret = **Imprint**(***List_of_int:*** pBlankObjectIDArray, ***List_of_int:*** pToolObjectIDArray)
27. ***int:*** iret = **SweepAlongVector**(***int:*** profileID, ***UDPVector:*** sweepVector, ***UDPSweepOptions:*** sweepOptions)
28. ***int:*** iret = **SweepAroundAxis**(***int:*** profileID, ***CoordinateSystemAxis:*** whichAxis, ***double:*** sweepAngle, **UDPSweepOptions:** sweepOptions)

29. *int:* iret = **SweepAlongPath**(*int:* profileID, *int:* pathID, *UDPSweepOptions:* sweepOptions)
30. *int:* iret = **Translate**(*int:* partID, *UDPVector:* translateVector)
31. *int:* iret = **Rotate**(*int:* partID, *CoordinateSystemAxis:* whichAxis, *double:* rotateAngle)
32. *int:* iret = **Mirror**(*int:* partID, *UDPPosition:* mirrorPlaneBasePosition, *UDPVector:* mirrorPlaneNormalVector)
33. *int:* iret = **Transform**(*int:* partID, *List_of_double:* rotationMatrix, *UDPVector:* translateVector)
34. *int:* iret = **Scale**(*int:* partID, *double:* xScale, *double:* yScale, *double:* zScale)
35. (*int:* lret, *List_of_double:* cloneIDs) = **DuplicateAlongLine**(*int:* partID, *UDPVector:* translateVector, *int:* numTotalObjs, *int:* cloneIDsListSize)
36. (*int:* lret, *List_of_double:* cloneIDs) = **DuplicateAroundAxis**(*int:* partID, *CoordinateSystemAxis:* whichAxis, *double:* rotateAngle, *int:* numTotalObjs, *int:* cloneIDsListSize)
37. *int:* iret = **DuplicateAndMirror**(*int:* partID, *UDPPosition:* mirrorPlaneBasePosition, *UDPVector:* mirrorPlaneNormalVector)
38. *int:* iret = **Connect**(*List_of_int:* objectIDArray)
39. *int:* iret = **Offset**(*int:* partID, *double:* offsetDistance)
40. *int:* iret = **Section**(*int:* partID, *CoordinateSystemPlane:* sectionPlane)
41. (*int:* iret , *int:* newPartID) = **Split**(*int:* partID, *CoordinateSystemPlane:* splitPlane, **SplitWhichSideToKeep:** whichSideToKeep, *bool:* bSplitCrossingObjectsOnly)
42. (*int:* iret , *List_of_int:* importedObjectIDs) = **ImportNativeBody2**(*string:* fileNameWithFullPath)
43. (*int:* lret, *List_of_int:* importedObjectIDs) = **ImportAnsoftGeometry**(*string:* fileNameWithFullPath, *List_of_string:* overridingParamsNameArray, *List_of_UDPParam:* overridingParamsArray)
44. *int:* iret = **Clone**(*int:* partID)
45. *int:* iret = **DeletePart**(*int:* partID)
46. *int:* iret = **CreateObjectFromFace**(*int:* faceID)
47. *int:* iret = **Fillet**(*UDPBLNDElements:* entitiesToFillet, *UDPBLNDFilletOptions:* filletOptions)
48. *int:* iret = **Chamfer**(*UDPBLNDElements:* entitiesToChamfer, *UDPBLNDChamferOptions:* chamferOptions)
49. (*int:* iret , *List_of_int:* newPartIDArray) = **DetachFaces**(*int:* newPartIDArraySize, *List_of_int:* faceIDArray)
50. (*int:* iret , *List_of_int:* newPartIDArray) = **DetachEdges**(*int:* newPartIDArraySize, *List_of_int:* edgeIDArray)
51. *int:* iret = **CreateObjectFromEdge**(*int:* edgeID)
52. *int:* iret = **SheetThicken**(*int:* partID, *double:* fThickness, *bool:* bThickenBothSides)
53. (*int:* iret , *List_of_int:* newPartIDArray) = **SweepFaceAlongNormal**(*int:* newPartIDArray-

Size, ***List_of_int:*** faceIDArray, ***double:*** sweepLength)

54. ***int:*** iret = **CoverLine**(***int:*** partID)
55. ***int:*** iret = **CoverSurface**(***int:*** partID)
56. ***int:*** iret = **UncoverFaces**(***List_of_int:*** faceIDArray)
57. (***int:*** iret , ***int:*** numPartsCreated, List_of_int>: faceIDArray) = **SeparateBodies**(***int:*** partID, ***int:*** numPartsCreated)
58. ***int:*** iret = **MoveFaces**(***List_of_int:*** faceIDArray, ***bool:*** bMoveAlongNormal, ***double:*** fOffset-Distance, ***UDPVector:*** moveVector)
59. ***int:*** iret = **WrapSheet**(***int:*** sheetBodyID, ***int:*** targetBodyID)
60. ***int:*** iret = **ImprintProjection**(***int:*** blankBodyID, ***List_of_int:*** toolBodyIDArray, ***bool:*** bNormalProjection, ***UDPVector:*** projectDirection, ***double:*** projectDistance)
61. ***string:*** str = **GetTempDirPath**()
62. ***string:*** str = **GetSysLibDirPath**()
63. ***string:*** str = **GetUserLibDirPath**()
64. ***string:*** str = **GetPersonalLibDirPath**()
65. ***string:*** str = **GetInstallDirPath**()
66. ***string:*** str = **GetProjectPath**()
67. (***bool:*** bret, ***bool:*** abort) = **SetProgress**(***UDPProgress:*** progress)

UDMExtension

Import

You do not have to add import statements for the predefined classes and structures - it is done for you and all data types described in this document can be used in your Python script.

However you have to add import statements to your helper scripts in your Lib sun-directory.

```
from UDM import *
```

Main class: UDMExtension

You must write a class derived from IUDMExtension with a mandatory name UDMExtension:

```
class UDMExtension(IUDMExtension):
```

The class should implement IUDMExtension methods described below.

IUDMExtension methods

All methods are the same as the methods in the C UDM implementation. The changes to the methods signatures are just to conform to the Python style.

Mandatory methods.

These methods must be implemented in the UDM Python script as methods of UDMExtension class.

GetInfo()

- returns UDMInfo object populated with appropriate UDM information.

IsAttachedToExternalEditor()

- returns True if UDM dll is attached to external editor.
- In case of python UDMs, this should typically return False

CreateInstance(funcLib)

- funcLib is UDMFunctionLibrary
- returns UDMParameters.

GetUnits(instanceId)

- instanceId is a long
- returns string containing units for the instance.

Refresh(funcLib, udmInParams, updatedParams, refreshModifiedPartsOnly, nonEditedPartRefIds)

This Method is called every time a UDM is refreshed. Geometry creation/refresh should happen in this method.

- funcLib is UDMFunctionLibrary
- udmInParams is a list of UDMParameters that comes from desktop
- updatedParams: UDM script can change the UDM parameters it receives. Updated parameters need to be sent back to desktop. If the UDM script is not going to change any of the parameters that it received, it needs to copy udmInParams to updatedParams.
- refreshModifiedPartsOnly is a boolean

  Supporting this flag is optional. For UDMs where the refresh performance is not an issue, it is recommended to ignore this flag and update all parts every time.

  This flag can be used to optimize performance of Refresh method when the model created by UDM is large. If the UDM consists of multiple parts, and new parameters change only a few parts amongst them, UDM script can only modify parts that are changed by the new parameters.
- nonEditedPartRefIds: If RefreshModifiedPartsOnly is true and the UDM script supports partial update, Refresh method needs to return ids of parts that are unchanged.

  returns True on success, False on failure.

ReleaseInstance(instanceId)

- instanceId is a long
- This should release any resources assigned to this particular instance of UDM.
- returns True on success, False on failure.

GetAttribNameForEntityId()

- Returns string that acts as a the name of the attribute containing entity IDs.

- For example, it can return a unique string such as "ATTRIB_XACIS_ID"
- Python UDMs should implement this method.

GetAttribNameForPartId()

- Returns string that acts as a the name of the attribute containing entity IDs.
- For example, it can return a unique string such as "ATTRIB_XACIS_ID" (Can be same as GetAttribNameForEntityId())
- Python UDMs should implement this method.

Optional methods

These methods have default implementations (default is to return NULL or FALSE depending on the return type) but can be overridden by the user as needed as methods of UDMExtension class.

DialogForDefinitionOptionsAndParams(self, defData, optData, params):

Replaces the old UDMDialogForDefinitionAndOptions method, which is still supported, but users are urged to use UDMDialogForDefinitionOptionsAndParams. If both methods are present, application will use UDMDialogForDefinitionOptionsAndParams.

- UDM can pop up dialog for UDM definition, options, parameters in this method. Definition, options, and parameters are set/modified by user and returned to application. Dll can also just give default definition, options and parameters.
- Returns two booleans and a string
- First boolean returns whether the method was successful or not.
- Second boolean returns whether the application should popup a dialog. If it is True, application will populate a dialog with definition, options, parameters that are returned.
- String returned contains length units for parameters.

DialogForDefinitionAndOptions(self, defData, optData) [Deprecated]

UDM can pop up dialog for UDM definition and options in this method. Definition, and options are set/modified by user and returned to application. Dll can also just give default definition and options.

- Returns two booleans.
- First boolean provides whether the call to this method was successful or not.
- Second boolean determines whether the application should pop up a dialog. If this is true, application will populate the dialog with the definitions and options that are returned. As no parameters are returned, no parameters are shown in this dialog.

GetInstanceSourceInfo(instanceId)

- instanceId is a long
- returns string containing source information of UDM instance. It is used to create initial name for UDM instance.

ShouldAttachDefinitionFilesToProject()

- returns true if any of definition files needs to be attached to project
- returns python list of string containing definition names of files or NULL

Example UDM

```
class UDMExtension(IUDMExtension):

     def IsAttachedToExternalEditor(self):
          return False

     def GetInfo(self)
          udmInfo = UDMInfo(
                name = "SampleUDM",
                purpose = "udm example",
                company="ANSYS",
                date="12.21.12",
                version = "1.0")

           return udmInfo
     ...
     ...
```

UDMFunctionLibrary

UDMFunctionLibrary implements IUDMFunctionLib interface. The IUDMFunctionLib object is passed as a parameter to Python script in the following functions

- CreateInstance
- Refresh

You can call any of the functions from the functions list (shown below).

```
partRefId = udmFunctionLib.GetPartRefId(partId)
```

For example sample code that calls GetBoundingBox in Python script can look like this:

```
partId      = 10
exact       = True
udpPosition = UDPPosition(0,0,0)
```

```
(bret, udpPositionLow, udpPositionHigh) =
udmFunctionLibrary.GetBoundingBox(partId,
exact);


if bret:
   udpPosition.X = udpPositionLow.X
```

As you can see udpPositionLow and udpPositionHigh output parameters are defined in the call to GetBoundingBox function. There is no need to define them before the function call.

Functions list:

1. ***List_of_UDMDefinition:*** udmDefinitionList = **GetDefinition**()
2. ***List_of_UDMOption:*** udmOptionList = **GetOptions**()
3. ***bool:*** bret = **SetMaterialName**(***string:*** matName, ***int:*** partId)
4. ***bool:*** bret = **SetMaterialName2**(***string:*** matName, ***string:*** partName)
5. ***bool:*** bret = **SetPartName**(***string:*** partName, ***int:*** partId)
6. ***int:*** iret = **GetInstanceId**()
7. ***string:*** str = **GetPartRefId**(***int:*** partId)
8. ***bool:*** bret = **SetPartRefId**(***int:*** partId, ***string:*** refId)
9. ***List_of_int:*** faceIds = **GetAllFaces**(***int:*** partId)
10. ***List_of_int:*** edgeIds = **GetAllEdges**(***int:*** partId)
11. ***List_of_int:*** vertexIds = **GetAllVertices**(***int:*** partId)
12. ***bool:*** bret = **SetFaceAttribs**(***List_of_int:*** faceIds, ***List_of_string:*** attribs)
13. ***bool:*** bret = **SetEdgeAttribs**(***List_of_int:*** edgeIds, ***List_of_string:*** attribs)
14. ***bool:*** bret = **SetVertexAttribs**(***List_of_int:*** vertexIds, ***List_of_string:*** attribs)
15. ***string:*** str = **GetModelerUnit**()
16. ***string:*** str = **GetCacheFileForUDMResume**()
17. ***bool:*** bret = **SetPartColor**(***int:*** partId, ***int:*** nColor)
18. ***bool:*** bret = **SetPartFlags**(***int:*** partId, ***int:*** nFlags)
19. (***bool:*** bret, ***UDPPosition:*** low, ***UDPPosition:*** high) = **GetBoundingBox**(***int:*** partId, ***bool:*** exact)
20. ***bool:*** bret = **IsParametricUpdate**()
21. ***bool:*** bret = **SetMaterialNameByRefId**(***string:*** partRefID, ***string:*** matName)
22. ***bool:*** bret = **SetPartNameByRefId**(***string:*** partRefId, ***string:*** partName)
23. ***bool:*** bret = **SetPartColorByRefId**(***string:*** partRefId, ***int:*** nColor)
24. ***bool:*** bret = **SetPartFlagsByRefId**(***string:*** partRefId, ***int:*** nFlags)

In addition to the above functions all functions defined in the UDPFunctionLib are available in the IUDMFunctionLib and can be called directly exactly the same way as the IUDMFunctionLib functions.

Example:

```
udmFunctionLib.CreateCircle(center,radius,ratio,isCovered)
```

UDP/UDM Structures and constants

The following sections describe:

- UDP/UDM Structures
- UDP/UDM Constants

UDP/UDM Structures

Differences compared to C API

- **UDMDefinition**
- **UDMOptions**
- **UDMParameters**

Instead of containing arrays of data, the structures contain single fields where each field corresponds to an item in a different array from the original C API. The structure objects thus constructed are added to the Python list. Alternately the Python list can be initialized using the structure objects.

Example (creating UDMParameter list):

```
udmParamList  = [UDMParameter("cubeSizeName",
UnitType.LengthUnit,
         UDPParam(ParamDataType.Double, cubeSize),
         ParamPropType.Value,
         ParamPropFlag.MustBeReal),
         UDMParameter("cubeDistanceName", UnitType.LengthUnit,
   UDPParam(ParamDataType.Double, cubeDistance),
         ParamPropType.Value,
         ParamPropFlag.MustBeReal),
   UDMParameter("numCubesName", UnitType.LengthUnit,
         UDPParam(ParamDataType.Int, numCubes),
         ParamPropType.Number,
         ParamPropFlag.MustBeInt)]
```

- **UDPParam**
- **UDPParamData**

Data field in UDPParam is now an object - the same for all types of data used - as Python can work with any type of data.

UDPParamData is obsolete, thus not implemented. Be sure to set proper data type to UDPParam.DataType when setting UDPParam.Data.

Example:

```
nCubesParam = UDPParam(ParamDataType.Int, numCubes)
nCubes = nCubesParam.Data

distanceParam = UDPParam()
distanceParam.setDouble(10.5)
doubleDistance = distanceParam.Data * 2
```

- **UDP3x3Matrix**

The structure is not implemented. Use size 9 Python List of doubles instead.

Example:

```
rotationMatrix =[0,0,1, 1,0,0, 0,0,1]
udpFunctionLib.Transform(partId, rotationMatrix,
translationVector)
```

List of structures

You can use constructors to create a structure. You can also modify fields - directly or by provided methods.

Example:

```
pos1 = UDPPosition(1,2,3)
pos2 = UDPPosition(x=1,y=10,z=0)
pos2.Z = pos1.Z

udpParam = UDPParam(ParamDataType.Double,1)
value = udpParam.Data
```

Structure	Construction	Members
UDPPrimitiveTypeInfo	UDPPrimitiveTypeInfo(string name, string purpose, string company, string date, string version)	string Name string Purpose string Company string Date string Version
UDPPrimitiveParameterDefinition	UDPPrimitiveParameterDefinition(string name, string description, UnitType unitType, double defaultValue)	string Name string Description UnitType UnitType double DefaultValue
UDPParam	UDPParam() UDPParam(ParamDataType dataType, object data) object can be int, double, string, bool or UDPPosition **methods:** setInt(int val) setBool(bool val) setString(string val) setDouble(double val) setPosition(UDPPosition val)	ParamDataType DataType object Data object can be int, double , string, bool or UDPPosition

Structure	Construction	Members
UDPPrimitiveParameterDefinition2	UDPPrimitiveParameterDefinition2(string name, string description, UnitType unitType, ParamPropType propType, ParamPropFlag propFlag, UDPParam defaultValue)	string Name string Description UnitType UnitType ParamPropType PropType ParamPropFlag PropFlag UDPParam DefaultValue
UDPPosition	UDPPosition(double x, double y, double z)	double X double Y double Z
UDPVector	UDPVector(double x, double y, double z)	double X double Y double Z
UDPSweepOptions	UDPSweepOptions(SweepDraftType draftType, double draftAngle, double twistAngle)	SweepDraftType DraftType double DraftAngle double TwistAngle
UDPPolylineSegmentDefinition	UDPPolylineSegmentDefinition(PolylineSegmentType segmentType, int segmentStartIndex, int numberOfPoints, double angle, UDPPosition centerPoint, CoordinateSystemPlane arcPlane)	PolylineSegmentType SegmentType int segmentStartIndex, int numberOfPoints, double angle, UDPPosition centerPoint, CoordinateSystemPlane arcPlane)

Structure	Construction	Members
UDPPolylineDefinition	UDPPolylineDefinition() UDPPolylineDefinition(List_of_UDPPosition positions, List_of_UDPPolylineSegmentDefinition segDefs, int closed, int covered)	int IsClosed int IsCovered List_of_UDPPosition ArrayOfPosition List_of_UDPPolylineSegmentDefinition ArrayOfSegmentDefinition
UDPEquationBasedCurveDefinition	UDPEquationBasedCurveDefinition(string functionXt, string functionYt, string functionZt, double tStart, double tEnd, int numOfPointsOnCurve)	string FunctionXt string FunctionYt string FunctionZt double TStart double TEnd int NumOfPointsOnCurve
UDPEquationBasedSurfaceDefinition	UDPEquationBasedSurfaceDefinition(string functionXuv, string functionYuv, string functionZuv, double uStart, double uEnd, double vStart, double vEnd)	string FunctionXuv string FunctionYuv string FunctionZuv double UStart double UEnd double VStart double VEnd
UDPHelixDefinition	UDPHelixDefinition(int profileID, UDPPosition ptOnAxis, UDPPosition axisDir, double noOfTurns, bool isRightHanded, double radiusChangePerTurn, double pitch)	int ProfileID UDPPosition PtOnAxis UDPPosition AxisDir double NoOfTurns bool IsRightHanded double RadiusChangePerTurn double Pitch

Structure	Construction	Members
UDPSpiralDefinition	UDPSpiralDefinition(int profileID, UDPPosition ptOnAxis, UDPPosition axisDir, double noOfTurns, bool isRightHanded, double radiusChangePerTurn)	int ProfileID UDPPosition PtOnAxis UDPPosition AxisDir double NoOfTurns bool IsRightHanded double RadiusChangePerTurn
UDPBLNDElements	UDPBLNDElements(int partID)	int PartID List_of_int ListOfEdges List_of_int ListOfFaces
UDPBLNDFilletOptions	UDPBLNDFilletOptions(bool supressFillet, BLNDFilletRadiusLaw filletRadiusLaw, double filletStartRadius, double filletEndRadius, bool followSmoothEdgeSequence, BLNDFilletType filletType, double setbackDistance, double bulgeFactor)	bool SupressFillet BLNDFilletRadiusLaw FilletRadiusLaw double FilletStartRadius double FilletEndRadius bool FollowSmoothEdgeSequence BLNDFilletType FilletType double SetbackDistance double BulgeFactor
UDPBLNDChamferOptions	UDPBLNDChamferOptions(bool supressChamfer, BLNDChamferRangeLaw chamferRangeLaw, double chamferLeftRange, double chamferRightRange)	bool SupressChamfer BLNDChamferRangeLaw ChamferRangeLaw double ChamferLeftRange double ChamferRightRange
UDPProgress	UDPProgress(int prog, int subProg, string mesg, string subMesg)	int Prog int SubProg string Mesg string SubMesg

Structure	Construction	Members
UDMInfo	UDMInfo(string name, string purpose, string company, string date, string version)	string Name string Purpose string Company string Date string Version
UDMDefinition	UDMDefinition() UDMDefinition(string name, UDParam value, ParamPropType propType, ParamPropFlag propFlag)	string DefName UDPParam DefValue ParamPropType PropType ParamPropFlag PropFlag
UDMOption	UDMOption() UDMOption(string name, UDParam value, ParamPropType propType, ParamPropFlag propFlag)	string OptName UDPParam OptValue ParamPropType PropType ParamPropFlag PropFlag
UDMParameter	UDMParameter() UDMParameter(string name, UDParam value, UnitType unitType, ParamPropType propType, ParamPropFlag propFlag)	string ParamName UDPParam ParamValue UnitType UnitType ParamPropType PropType ParamPropFlag PropFlag

UDP/UDM Constants

Full names of enum constants must be used in scripts.

Example:

```
unitType = UnitType.LengthUnit
```

```
dataType = ParamDataType.Int
```

Enum constants:

enum Constant	Parameters
UnitType	NoUnit LengthUnit AngleUnit
ParamDataType	Int Double String Bool Position Unknown
ParamPropType	Text Menu Number Value FileName Checkbox Position Unknown
ParamPropFlag	NoFlag ReadOnly MustBeInt MustBeReal Hidden Unknown
CoordinateSystemAxis	XAxis YAxis ZAxis
CoordinateSystemPlane	XYPlane YZPlane ZXPlane

enum Constant	Parameters
SweepDraftType	ExtendedDraft RoundDraft NaturalDraft MixedDraft
SplitWhichSideToKeep	SplitKeepBoth SplitKeepPositiveOnly SplitKeepNegativeOnly
PolylineSegmentType	LineSegment ArcSegment SplineSegment AngularArcSegment
MessageSeverity	WarningMessage ErrorMessage InfoMessage IncompleteMessage FatalMessage
BLNDFilletRadiusLaw	BLNDConstantRadius BLNDVariableRadius
BLNDFilletType	BLNDRound BLNDMitered
BLNDChamferRangeLaw	BLNDConstantRange BLNDVariableRange
PartPropertyFlags	PropNonModel PropDisplayWireFrame PropReadOnly PostprocessingGeometry PropInvisible PropShowDirection PropDummy

Logging and Desktop Messaging

There is error logging and displaying script errors in Desktop Message Manager for UDP/UDM Python scripts.

You can also add messages to Message Manager using the UDPFunctionLibrary AddMessage method, but no logging is done in this case.

Collecting EBU Logs

The instructions below are for Windows, but they work similarly for Linux. On Linux, determine your shell (execute echo $SHELL) and, based on that, use setenv or export to set the environment variables.

1. Create a directory that serves as a folder for the logs. Make sure that this directory is empty: clean it out if it already has files. For this example, the name of the directory is c:\ansdir\ansdebug.
2. Start a new shell to set the environment. (You can set it permanently, but this is not recommended.)
3. Set the following env variables in this shell:
   - Set ANSOFT_DEBUG_MODE to 2. Set it to a higher number such as 4 or 5 if the developer requests it.
   - Set ANSOFT_DEBUG_LOG to a file in the previously created ansdebug directory: for example, set ANSOFT_DEBUG_LOG=c:\ansdir\ansdebug\ansdebug.log.
   - Set ANSOFT_DEBUG_LOG_SEPARATE to 1.
4. From the above shell, start the ANSYS Electromagnetics product and exercise the problem scenario.
5. Zip all the log files in the c:\ansdir\ansdebug directory, and send them to the developer or support person requesting them.

Collecting WB Logs

1. Under WB, execute the Tools/Options menu option, and under the Journals And Logs section, ensure the Write Workbench Log Files is checked.

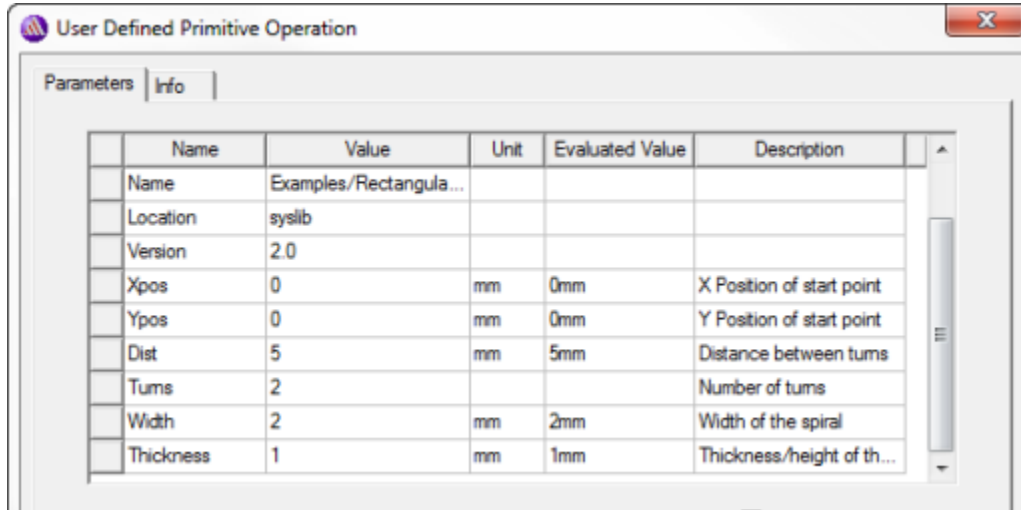
   Note the location of the log files. (On Windows this is typically %TEMP%\WorkbenchLogs.)

   Navigate to that directory and remove all files.
2. Set up the EBU env vars if you need to collect those as well, following the instructions above.
3. Exercise the problem scenario.
4. Collect the WB log files and optionally the ANSYS Electromagnetics logs and, send them to the developer or support person requesting them.

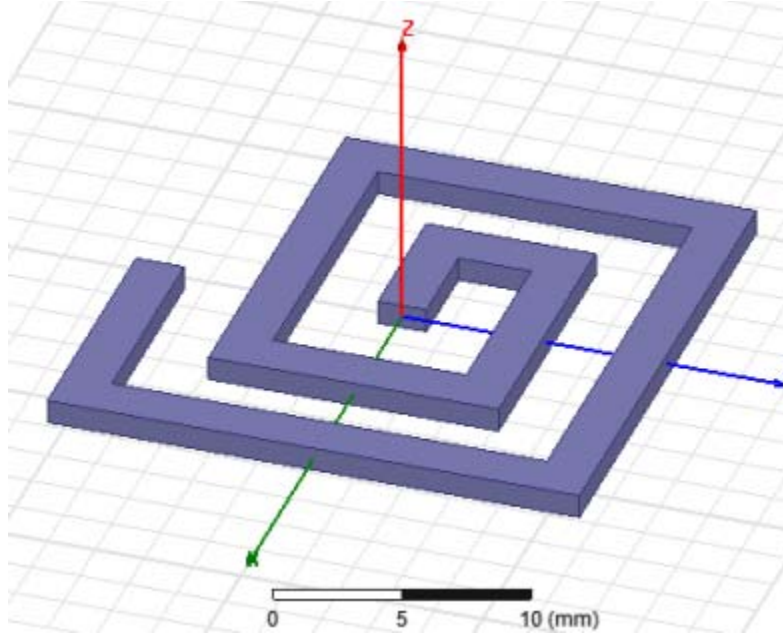
Example Scripts for Python UDP and UDM

The UDP example for Python creates a rectangular spiral.

1. Use **Draw>User Defined Primitive>Examples>Rectangular Spiral (Python)**. This opens a dialog in which you can set parameters such as Xpos, Ypos, distance between turns, and Number of turns.

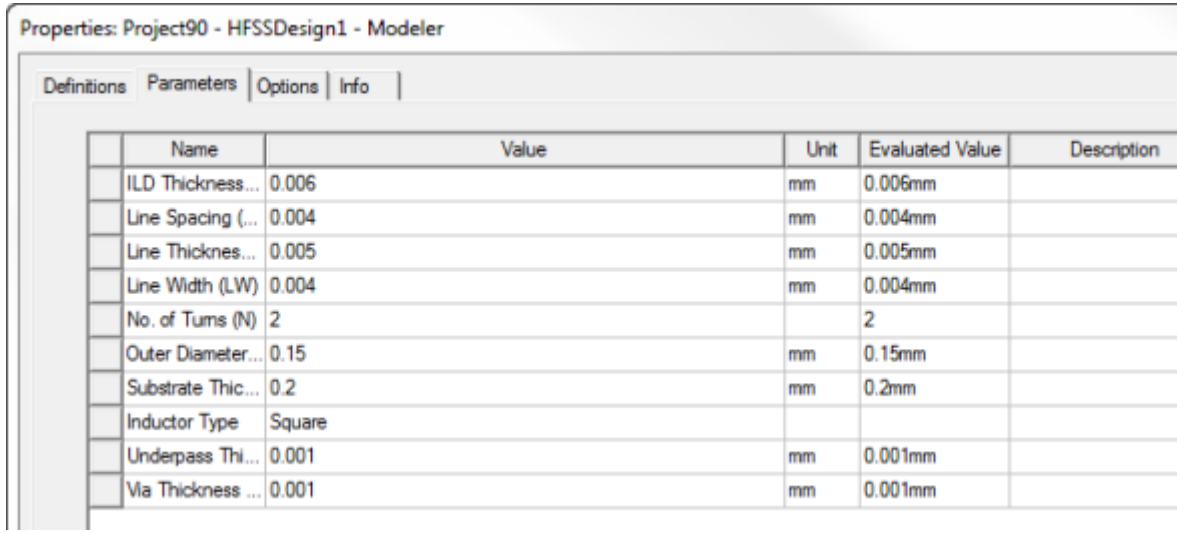


Click OK to create the spiral.



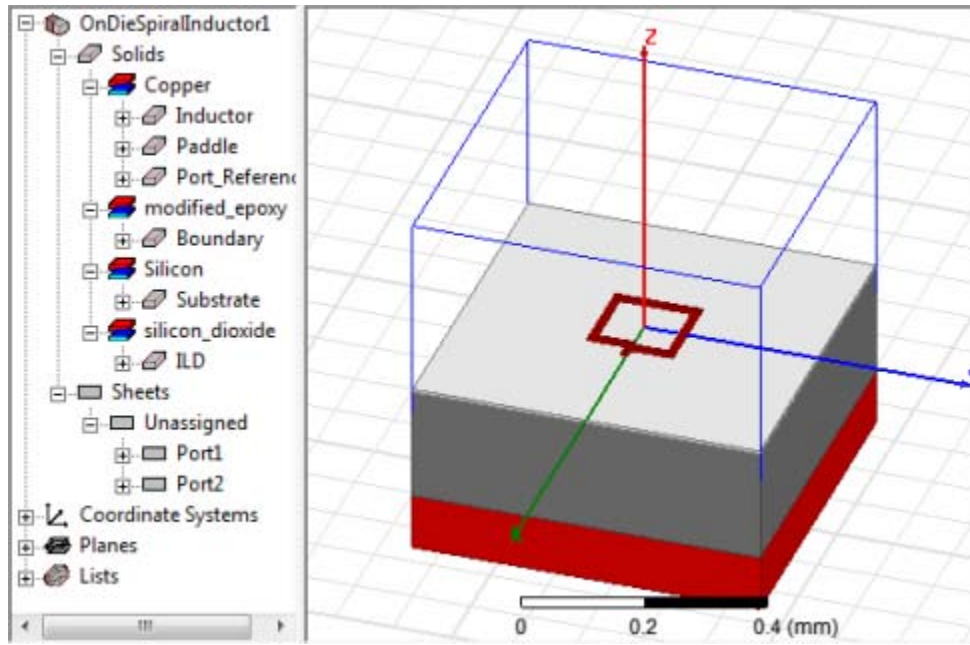
The UDM example for Python creates an on die spiral inductor.

1. Use **Draw>User Defined Model>OnDieSpiralInductor**.
   This opens a Properties dialog for which you can set parameters.



	Name	Value	Unit	Evaluated Value	Description
	ILD Thickness...	0.006	mm	0.006mm	
	Line Spacing (...	0.004	mm	0.004mm	
	Line Thicknes...	0.005	mm	0.005mm	
	Line Width (LW)	0.004	mm	0.004mm	
	No. of Turns (N)	2		2	
	Outer Diameter...	0.15	mm	0.15mm	
	Substrate Thic...	0.2	mm	0.2mm	
	Inductor Type	Square			
	Underpass Thi...	0.001	mm	0.001mm	
	Via Thickness ...	0.001	mm	0.001mm	

2. Set the parameters and click OK create the UDM.



23

Glossaries

Glossary of Terms

cost function In an optimization setup, a cost function is based on goal values specified for at least one solution quantity. Optimetrics changes the design parameter values to fulfill the cost function. The cost function can be based on any solution quantity that HFSS can compute, such as field values, S-parameters, and eigenmode data.

design variation A single combination of variable values that is solved during a parametric or optimization setup.

Euler Angles Euler angles are used in Ansoft software to carry out a coordinate transformation from one coordinate system to another. The Swiss mathematician and physicist Leonhard Euler first developed the classical rotation theorem to describe rotations in 3D space. The angles used are Euler angles and can be used to describe any 3D rotation. These angles, given by (ö, è, ø) represent a series of sequential rotations about two axis of the coordinate system. The first rotation (ö) represents a rotation about the Z-axis of the source coordinate system (X, Y, Z) which results in an intermediate coordinate system denoted by (X", Y", Z"). The second rotation (è) represents a rotation of the intermediate coordinate system about the X"-axis, again resulting in an intermediate coordinate system denoted by (X', Y', Z'). The third and final rotation (ø) represents a rotation about the Z'-axis of the intermediate coordinate system. The final rotation completes the rotation and results in the "target" coordinate system denoted (X, Y, Z).

For further information see, Eric W. Weisstein, "Euler Angles." From *MathWorld* - A Wolfram Web Resource.

http://mathworld.wolfram.com/EulerAngles.html .

goal	In an optimization setup, a goal is the value of a solution quantity that you want to be achieved during the optimization. A goal is represented as one row in the cost function table. Each cost function defined in an optimization setup must include at least one goal.
nominal design	The original model on which Optimetrics analyses are based.
sweep definition	See *variable sweep definition.*
variable sweep definition	A set of variable values within a range that Optimetrics drives HFSS to solve when a parametric setup is analyzed. A parametric setup can include one or more sweep definitions.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z