IC-CAP
2014.04
HF3

DynaFET Modeling

KEYSIGHT
TECHNOLOGIES

Notice

Acknowledgments

Qt Version 4.8.4
Qt Notice
The Qt code was modified. Used by permission.
Qt Copyright

Qt Availability http://www.qtsoftware.com/downloads
Patches Applied to Qt can be found in the installation at:
$HPEESOF_DIR/prod/licenses/thirdparty/qt/patches.
You may also contact Brian Buchanan at Keysight Inc. at brian_buchanan@Keysight.com for more information. For details see: http://bmaster.soco.Keysight.com/mw/Qt_License_Information

Python

A. HISTORY OF THE SOFTWARE

==========================

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see http://www.zope.com). In 2001, the Python Software Foundation (PSF, see http://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

===

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT
INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any
third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1
----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

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, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior
permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

bzip2

This copy of Python includes a copy of bzip2, which is licensed under the following terms:

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2007 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``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 AUTHOR 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.

Julian Seward, jseward@bzip.org
bzip2/libbzip2 version 1.0.5 of 10 December 2007

OpenSSL

This copy of Python includes a copy of openssl, which is licensed under the following terms:

LICENSE ISSUES
==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL
please contact openssl-core@openssl.org.

OpenSSL License

Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. 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.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED 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 OpenSSL PROJECT OR ITS 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.
==

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).
The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed.
If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. 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.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
"This product includes cryptographic software written by
Eric Young (eay@cryptsoft.com)"
The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related.
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:
"This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``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 AUTHOR 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.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence including the GNU Public Licence.

Tcl
This copy of Python includes a copy of Tcl, which is licensed under the following terms:

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, ActiveState Corporation and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

Tk
This copy of Python includes a copy of Tk, which is licensed under the following terms:

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

Mersenne Twister
The _random module includes code based on a download from http://www.math.keio.ac.jp/ matumoto /MT2002/emt19937ar.html. The following are the verbatim comments from the original code:

A C-program for MT19937, with initialization improved 2002/1/26. Coded by Takuji Nishimura and Makoto Matsumoto.

Before using, initialize the state by using init_genrand(seed) or init_by_array(init_key, key_length).

Any feedback is very welcome. http://www.math.keio.ac.jp/matumoto/emt.html
email: matumoto@math.keio.ac.jp

Sockets

The socket module uses the functions, getaddrinfo(), and getnameinfo(), which are coded in separate source files from the WIDE Project, http://www.wide.ad.jp/.

Floating point exception control

The source for the fpectl module includes the following notice:

MD5 message digest algorithm

The source code for the md5 module contains the following notice:

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch
ghost@aladdin.com

Independent implementation of MD5 (RFC 1321).

This code implements the MD5 Algorithm defined in RFC 1321, whose text is available at
http://www.ietf.org/rfc/rfc1321.txt
The code is derived from the text of the RFC, including the test suite (section A.5) but excluding the rest of Appendix A. It does not include any code or documentation that is identified in the RFC as being copyrighted.

The original and principal author of md5.h is L. Peter Deutsch <ghost@aladdin.com>. Other authors are noted in the change history that follows (in reverse chronological order):

2002-04-13 lpd Removed support for non-ANSI compilers; removed
references to Ghostscript; clarified derivation from RFC 1321;
now handles byte order either statically or dynamically.
1999-11-04 lpd Edited comments slightly for automatic TOC extraction.
1999-10-18 lpd Fixed typo in header comment (ansi2knr rather than md5);
added conditionalization for C++ compilation from Martin
Purschke <purschke@bnl.gov>.
1999-05-03 lpd Original version.

Asynchronous socket services

The asynchat and asyncore modules contain the following notice:

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, and that the name of Sam
Rushing not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

SAM RUSHING DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL SAM RUSHING BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Cookie management

The Cookie module contains the following notice:

Copyright 2000 by Timothy O'Malley <timo@alum.mit.edu>

All Rights Reserved

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, and that the name of Timothy O'Malley not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

Timothy O'Malley DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL Timothy O'Malley BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Profiling

The profile and pstats modules contain the following notice:

Copyright 1994, by InfoSeek Corporation, all rights reserved.
Written by James Roskind

Permission to use, copy, modify, and distribute this Python software and its associated documentation for any purpose (subject to the restriction in the following sentence) without fee is hereby granted, provided that the above copyright notice appears in all copies, and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of InfoSeek not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. This permission is explicitly restricted to the copying and modification of the software to remain in Python, compiled Python, or other languages (such as C) wherein the modified or derived code is exclusively imported into a Python module.

INFOSEEK CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INFOSEEK CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Execution tracing

The trace module contains the following notice:

portions copyright 2001, Autonomous Zones Industries, Inc., all rights...
err... reserved and offered to the public under the terms of the
Python 2.2 license.
Author: Zooko O'Whielacronx http://zooko.com/ zooko@zooko.com

Copyright 2000, Mojam Media, Inc., all rights reserved.
Author: Skip Montanaro

Copyright 1999, Bioreason, Inc., all rights reserved.
Author: Andrew Dalke

Copyright 1995-1997, Automatrix, Inc., all rights reserved.
Author: Skip Montanaro

Copyright 1991-1995, Stichting Mathematisch Centrum, all rights reserved.

Permission to use, copy, modify, and distribute this Python software and its associated documentation for any purpose without fee is hereby granted, provided that the above copyright notice appears in all copies, and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of neither Automatrix, Bioreason or Mojam Media be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

UUencode and UUdecode functions

The uu module contains the following notice:

Copyright 1994 by Lance Ellinghouse
Cathedral City, California Republic, United States of America.
All Rights Reserved
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, and that the name of Lance Ellinghouse not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
LANCE ELLINGHOUSE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL LANCE ELLINGHOUSE CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Modified by Jack Jansen, CWI, July 1995:

- Use binascii module to do the actual line-by-line conversion between ascii and binary. This results in a 1000-fold speedup. The C version is still 5 times faster, though.
- Arguments more compliant with Python standard

XML Remote Procedure Calls

The xmlrpclib module contains the following notice:

The XML-RPC client interface is

Copyright (c) 1999-2002 by Secret Labs AB
Copyright (c) 1999-2002 by Fredrik Lundh

By obtaining, using, and/or copying this software and/or its associated documentation, you agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its associated documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies, and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Secret Labs AB or the author not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

SECRET LABS AB AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANT-ABILITY AND FITNESS. IN NO EVENT SHALL SECRET LABS AB OR THE AUTHOR BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

test_epoll

The test_epoll contains the following notice:

Copyright (c) 2001-2006 Twisted Matrix Laboratories.

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 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.

Select kqueue

The select and contains the following notice for the kqueue interface:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. 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.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR 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 AUTHOR 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.

strtod and dtoa

The file Python/dtoa.c, which supplies C functions dtoa and strtod for conversion of C doubles to and from strings, is derived from the file of the same name by David M. Gay, currently available from http://www.netlib.org/fp/. The original file, as retrieved on March 16, 2009, contains the following copyright and licensing notice:

/**

The author of this software is David M. Gay.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

***/

expat

The pyexpat extension is built using an included copy of the expat sources unless the build is configured --with-system-expat:

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd
and Clark Cooper

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 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.

libffi

The _ctypes extension is built using an included copy of the libffi sources unless the build is configured --with-system-libffi:

Copyright (c) 1996-2008 Red Hat, Inc and others.

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 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.

zlib

The zlib extension is built using an included copy of the zlib sources unless the zlib version found on the system is too old to be used for the build:

Copyright (C) 1995-2010 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

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 names of Greg Stein, Mark Hammond nor the name of 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.

adodbapi
----
adodbapi - A python DB API 2.0 (PEP 249) interface to Microsoft ADO

Copyright (C) 2002 Henrik Ekelund, version 2.1 by Vernon Cole

- http://sourceforge.net/projects/pywin32
- http://sourceforge.net/projects/adodbapi

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., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

django adaptations and refactoring by Adam Vandenberg

isapi

Copyright 2002-2003 by Blackdog Software Pty Ltd.

All Rights Reserved

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, and that the name of Blackdog Software not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

BLACKDOG SOFTWARE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL BLACKDOG SOFTWARE BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Scintilla and SciTE

Copyright 1998-2003 by Neil Hodgson <neilh@scintilla.org>

All Rights Reserved

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.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Numpy

Numpy license
Copyright ? 2005-2012, NumPy Developers.
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 NumPy Developers nor the names of any 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 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.

Scipy

Copyright (c) 2001, 2002 Enthought, Inc.
All rights reserved.

Copyright (c) 2003-2012 SciPy Developers.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

a. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
b. 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.
c. Neither the name of Enthought nor the names of the SciPy Developers
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.

Matplotlib

Matplotlib only uses BSD compatible code, and its license is based on the PSF license. See the Open Source Initiative licenses page for details on individual licenses. Non-BSD compatible licenses (eg LGPL) are acceptable in matplotlib toolkits-index. For a discussion of the motivations behind the licencing choice, see Licenses.

License agreement for matplotlib 1.2.0
1. This LICENSE AGREEMENT is between John D. Hunter (“JDH”), and the Individual or Organization (“Licensee”) accessing and otherwise using matplotlib software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, JDH hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use matplotlib 1.2.0 alone or in any derivative version, provided, however, that JDH’s License Agreement and JDH’s notice of copyright, i.e., “Copyright (c) 2002-2009 John D. Hunter; All Rights Reserved” are retained in matplotlib 1.2.0 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates matplotlib 1.2.0 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to matplotlib 1.2.0.

4. JDH is making matplotlib 1.2.0 available to Licensee on an “AS IS” basis. JDH MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, JDH MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF MATPLOTLIB 1.2.0 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. JDH SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF MATPLOTLIB 1.2.0 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING MATPLOTLIB 1.2.0, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between JDH and Licensee. This License Agreement does not grant permission to use JDH trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using matplotlib 1.2.0, Licensee agrees to be bound by the terms and conditions of this License Agreement.

libpng

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

This code is released under the libpng license.

libpng versions 1.2.6, August 15, 2004, through 1.6.2, April 25, 2013, are Copyright (c) 2004, 2006-2012 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.2.5 with the following individual added to the list of Contributing Authors

Cosmin Truta

libpng versions 1.0.7, July 1, 2000, through 1.2.5 - October 3, 2002, are Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors

Simon-Pierre Cadieux
Eric S. Raymond
Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996, 1997 Andreas Dilger
Distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

Table of Contents

The following topics describe the DynaFET Modeling Package.

Watch the recorded webcast to learn more about DynaFET.

Introduction to DynaFET

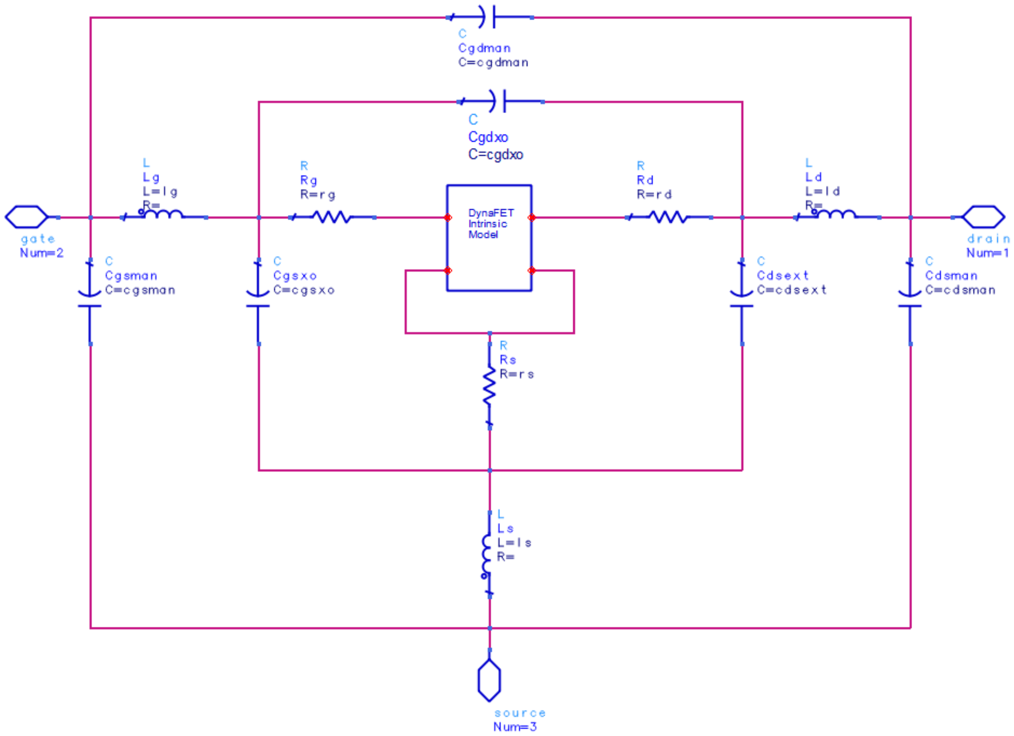
DynaFET is a non-quasi static large-signal FET model. It incorporates self-heating and other multiple timescale dynamics necessary to describe the large-signal behavior of III-V FET technologies including GaAs and GaN. It uses Artificial Neural Networks (ANNs) to model the I-V and Q-V relations in a general way. The ANN is trained on non-linear waveforms measured on a non-linear vector network analyzer (NVNA). The NVNA waveforms are measured over the entire device operation range in such a way that the device integrity is not compromised.

The model supports geometrical scaling rule(s) and so measurement of just one device can be scaled to other gate widths.

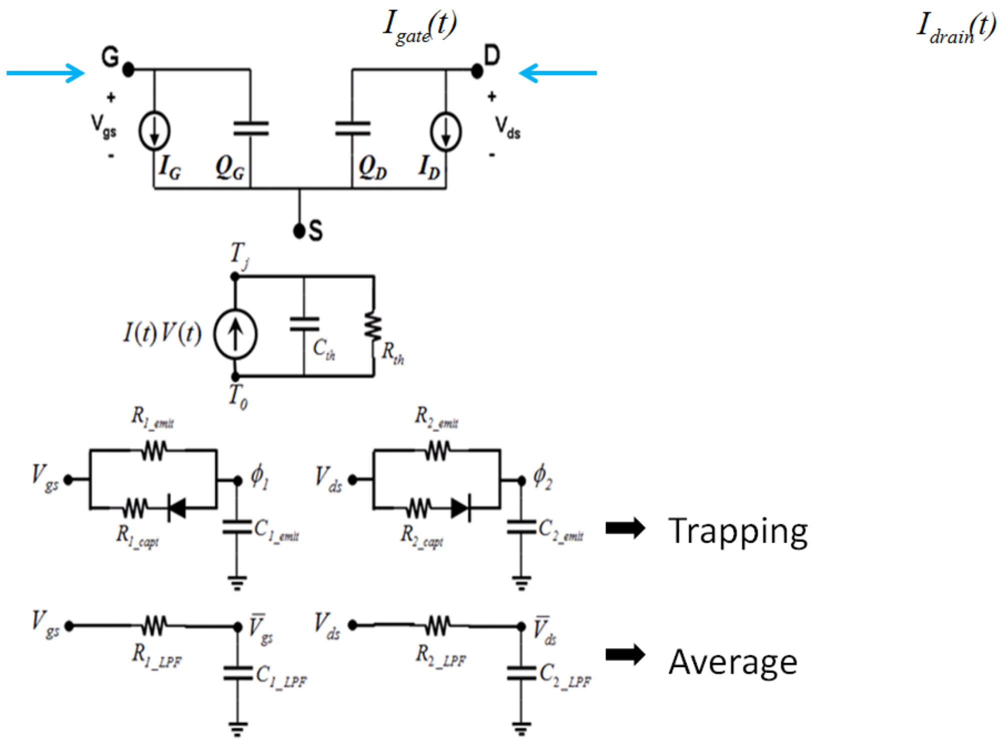
The key features and benefits of the DynaFET are listed in the table below.

Feature	Benefit
Dynamic self-heating and ambient temperature-dependence	Accuracy versus temperature; thermal memory modeled
Gate-lag and drain-lag independently modeled (knee walk-out, power slump)	Power slump, bias current versus Pin, and PAE well modeled
Aritificial neural networks (ANNs) for very general, smooth I-V / Q-V relations	Details non-linear coupling of trap states to current accurately and smoothly modeled. Model does not need to know detailed physics-ideal for new and changing processes
NVNA non-linear data for model generation and independent non-linear validation	Data over entire operating range while maintaining device safety; NVNA data closer to actual size condition
Simple geometrical scaling	Same as standard expectations for a copact time-domain transistor model

The DynaFET model is comprised of a conventional parasitic network surrounding the intrinsic model. The topology of the model is depicted in the following figure.

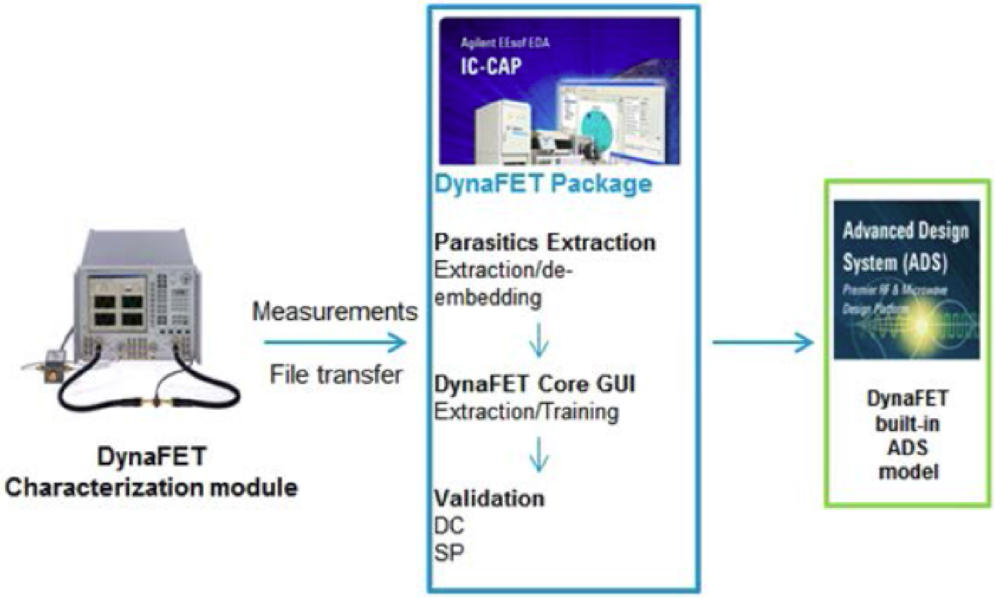


The intrinsic model contains the ANN and a self-heating circuit, and a gate and drain circuits to represent the gate and drain traps. These circuits provide inputs to calculate auxiliary variables. The auxiliary variables, along with the input non-linear waveforms are used to train the ANN. The following figure depicts the Auxiliary circuits used in the intrinsic DynaFET model.



DynaFET Extraction Flow

The following figure depicts the functional relationship of measurements and model.

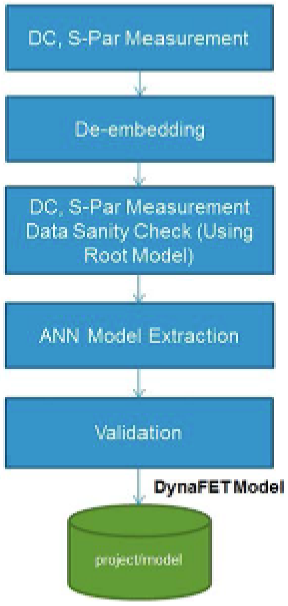


The IC-CAP DynaFET module manages the data measured by the non-linear vector network analyzer as well as data from measurements it made. The IC-CAP module also trains the ANN and outputs the necessary files for the ADS implementation to reference.

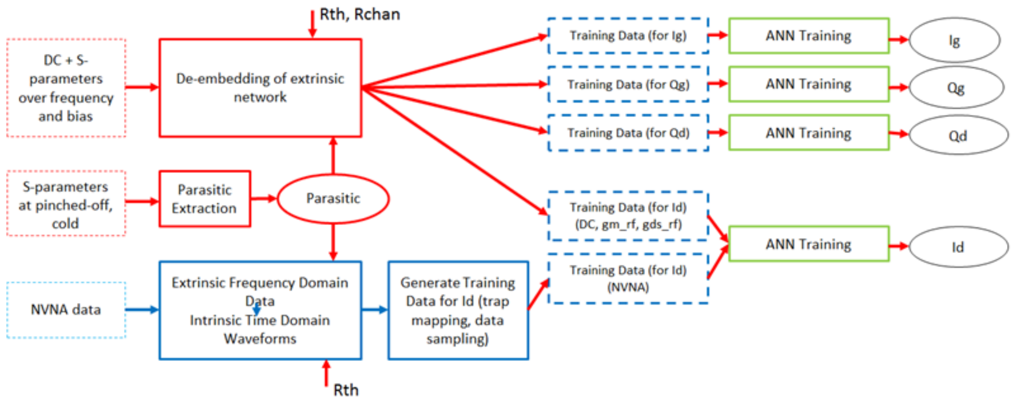
The DynaFET Extraction Package generates a model from the DC, S-Parameters, and large-signal waveform data. The DynaFET Package works with the software on the PNA-X to acquire non-linear waveform data. It then extracts the parasitic network elements, trains the Artificial Neural Network, then presents the results for validation against other measurements. Finally, it generates a model file that is compatible with the DynaFET implementation in ADS (versions ADS2014 and later).

The package comprises the key procedural steps in the DynaFET extraction flow, as depicted in the following figure:

- **Data (Daq) Acquisition**: Measures DC, S Parameters and copies the NVNA data file to the project directory.
- **De-embedding**: Executes de-embedding with flexible configuration.
- **Extraction of Extrinsic Elements**: Extracts extrinsic linear elements.
- **Intermediate Review**: Performs data sanity check by generating a Root model.
- **ANN Model Generation**: Extracts the core model Artificial Neural Network.
- **Model validation**: Models the simulated results against measured data that was not part of the model fitting data set.



Each step is expected to be executed sequentially. The output of each step are data files, stored locally in the project, which are used by subsequent steps. If you return to a previous step, it is mandatory to repeat the steps that follow as well. A detailed DynaFET mode extraction flow is depicted in the following figure.



In the above figure, the red lines indicate the data measured by the IC-CAP DynaFET program.

The IC-CAP DynaFET module manages the complex data generated by the non-linear vector network analyzer, measures linear DC and S-Parameters, and then, using additional information you have supplied, trains the ANN. The model output is an encrypted file that is used by the ADS DynaFET model.

Project Structure

The following table displays the project structure (created when the DynaFET example is copied to your working directory).

/$project_root	
/config	hardware setting, model card for simulation
/daq	data files generated by data acquisition

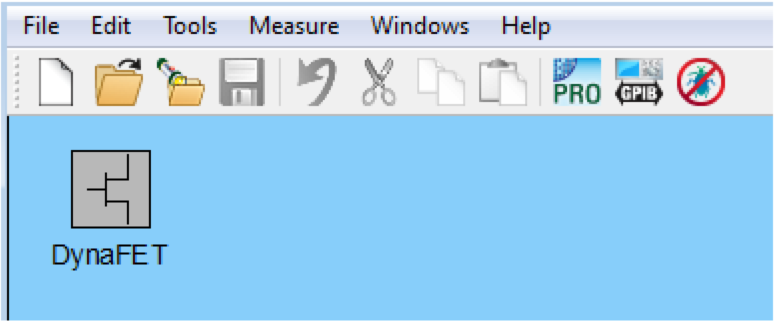
/deem	data files generated by de-embedding
/extraction	intrinsic data
/review	root model data
/ann	ann training data
/qg	
/model	ann weights for model generation of Qg
/test	test ann weights
/sweep	sweep ann weights
/qd	
/model	ann weights for model generation of Qd
/test	test ann weights
/sweep	sweep ann weights
/ig	
/model	ann weights for model generation of Ig
/test	test ann weights
/sweep	sweep ann weights
/id	
/model	ann weights for model generation of Id
/test	test ann weights
/sweep	sweep ann weights
/validation	DynaFET model card

Using the DynaFET Modeling Package

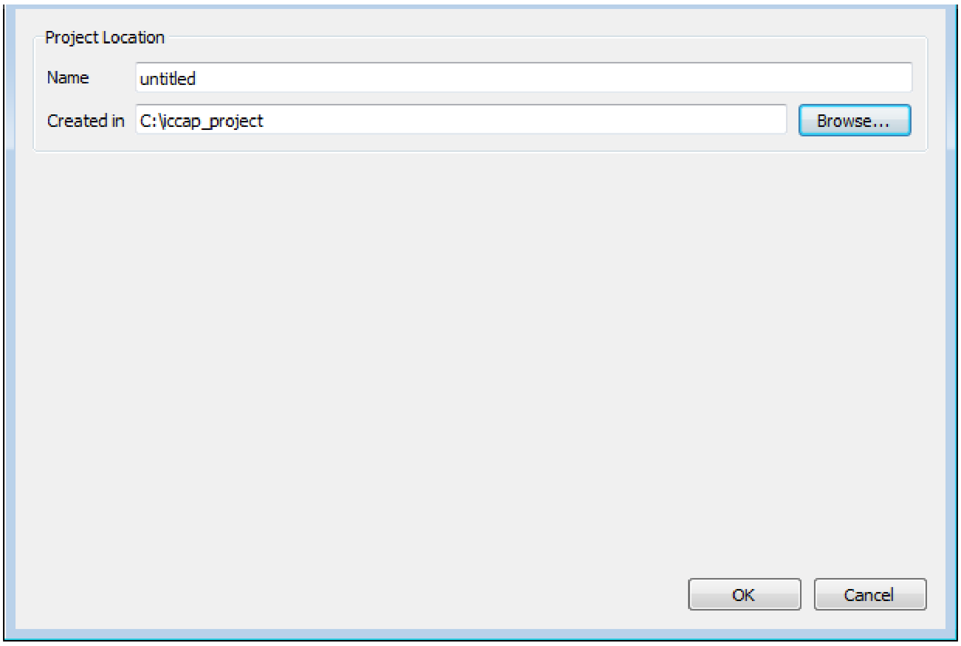
A detailed description of using the DynaFET modeling package is found in "STEP-BY-STEP PROCEDURE FOR MEASURING AND EXTRACTING A DYNAFET MODEL".

To launch the DynaFET modeling package:

1. Select **File > Examples** from the IC-CAP main window.
   The File Open dialog box is displayed.
2. Select the **DynaFET.mdl** model file available under **model_files > mesfet > dynafet** folder.
3. Click **Open**.
   An icon of the DynaFET.mdl model file is displayed in the IC-CAP main window.

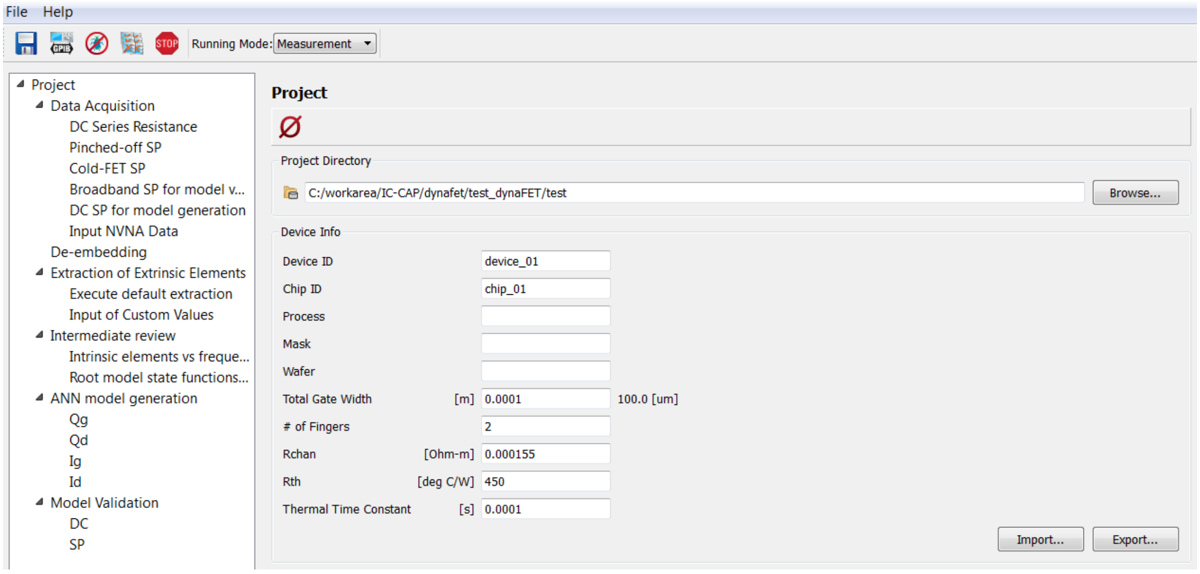


4. Double-click the DynaFET.mdl icon.
   The Copy DynaFET Example Project dialog box is displayed.



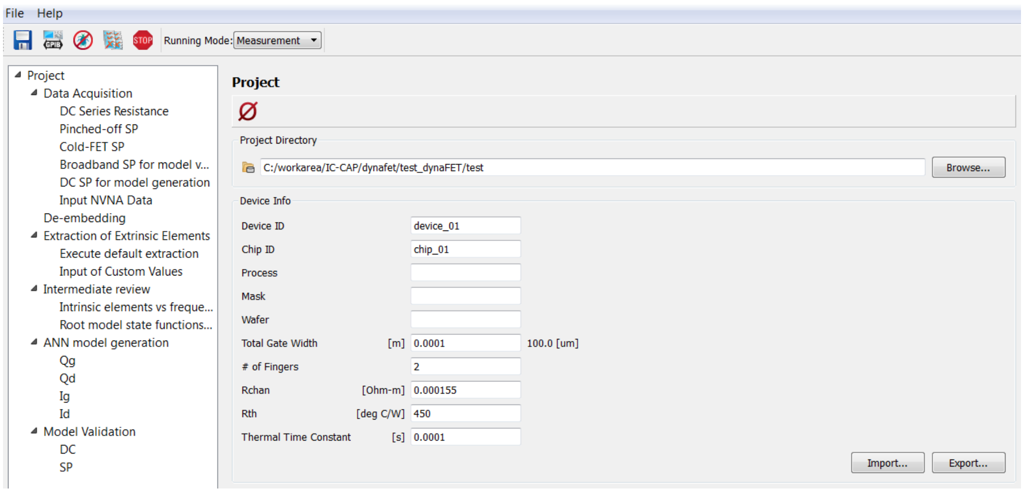
5. Specify the project **Name** and click **Browse** to select the project location.

6. Click **OK** to copy the example to your local working directory.
   The DynaFET Extraction Package window is displayed. For information on the extraction window, see Understanding the DynaFET Extraction Package Window.



Understanding the DynaFET Extraction Package Window

The DynaFET Project window allows you to save the project configuration and perform various project settings for a DynaFET model. The following figure depicts the DynaFET Extraction Package window that is displayed after entering the project name and directory.



The DynaFET Extarction Package window mainly consists of the following parts:

1. **DynaFET Menu Bar**: Provides options to create, open, save a DynaFET model file, and to access documentation.
2. **DynaFET Toolbar**: Provides shortcuts for commonly used commands, such as opening the hardware window, saving a project configuration, and plot options.
3. **Navigation View**: Displays project options as a hierarchical list of task flow in a project.
4. **Workspace Window**: A context dependent area, which changes according to the project action selected from the Navigation View.

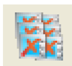
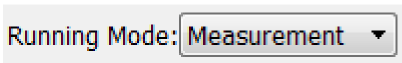
DynaFET Menu Bar

The following menus are available in the DynaFET Extraction Package window.

- **File:** Allows you to create a new project, open an existing project, save the project, or copy an example under a working directory.
- **Help:** Allows you the access the DynaFET Modeling documentation.

DynaFET Toolbar

The following table describes the main DynaFET toolbar options. You can view or hide the toolbars toggle by right-clicking on the toolbar and selecting the **DynaFET ToolBar** option.

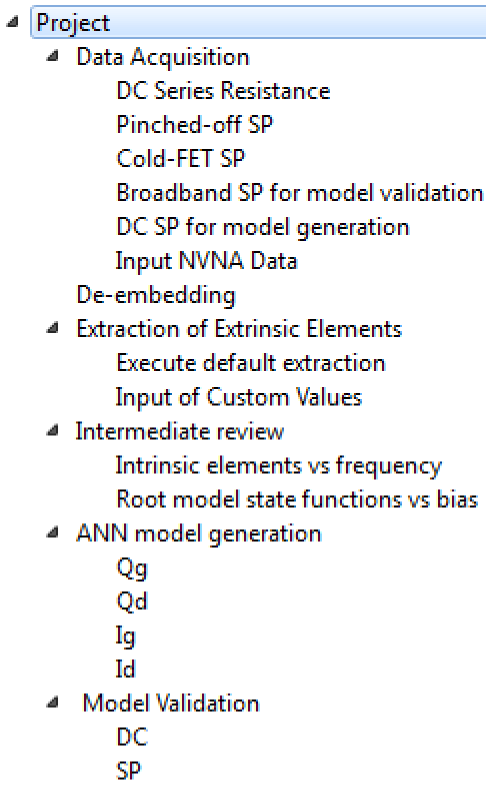
Option	**Icon**	**Description**
Save		Saves a project configuration
Open Hardware Window		Opens the hardware setup window
Open Simulation Debugger		Opens the simulation debugger window
Close All Plots		Closes all the plots for a project
Stop All ANN		Stops all the artificial neural networks for a project
Running Mode		• Measurement • Simulation

Navigation View

DynaFET Modeling User Interface provides a tree-structured representation of the project with various components like data acquisition, de-embedding, etc. You can use the navigation view to perform various project settings. Typically, the parent page either provides some general information about the step, or settings that apply to all the sub-steps in that main step.

Note that if you go back to a previous step and re-execute any function or change a value, you must re-execute the steps following that step. Going to a prior step and re-displaying information does not require re-executing the other steps.

The following figure displays the Navigation View.



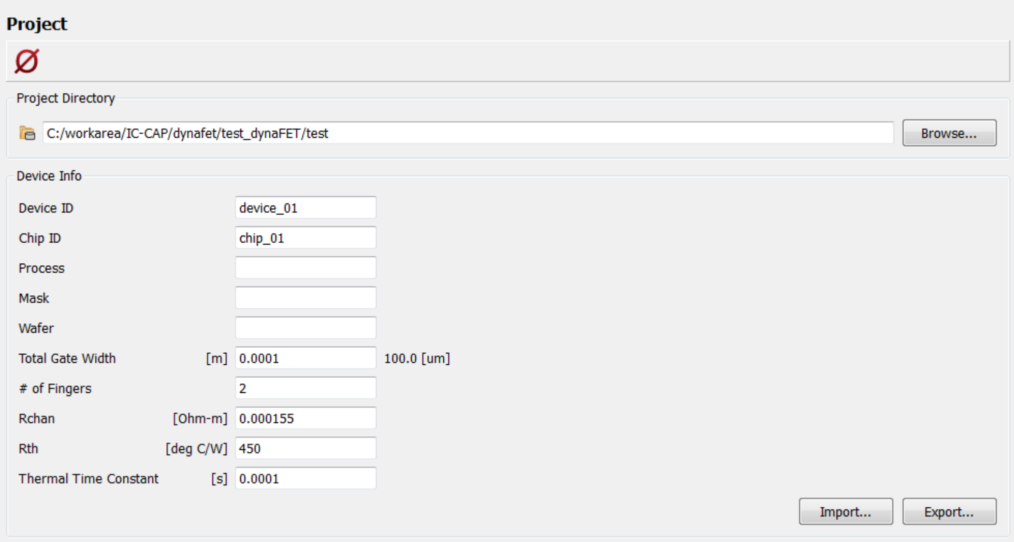
The Navigation view consist of the following:

- Data Acquisition
- De-embedding
- Extraction of Extrinsic Elements
- Intermediate Review
- ANN Model Generation
- Model Validation

Workspace Window

The workspace window provides a display and entry fields that are appropriate for the particular step that is active in the Navigation Window.

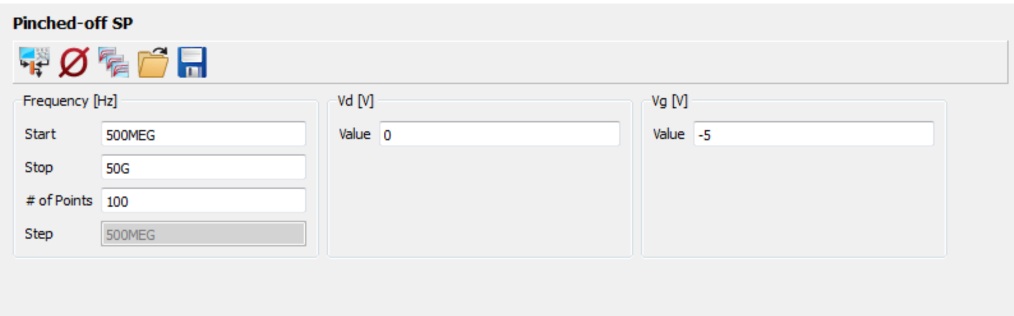
The following figure displays the Workspace Window.



The following table describes the options available in the workspace window.

Option	Description
Project Directory	Displays the example project location.
Device Info	Displays the device information.

Each window contain icons to execute various tasks and these icons vary depending on the selection . For example,  icon is available in the **Pinched-off SP** workspace window (depicted below) and is used to clear the data values for a particular step.

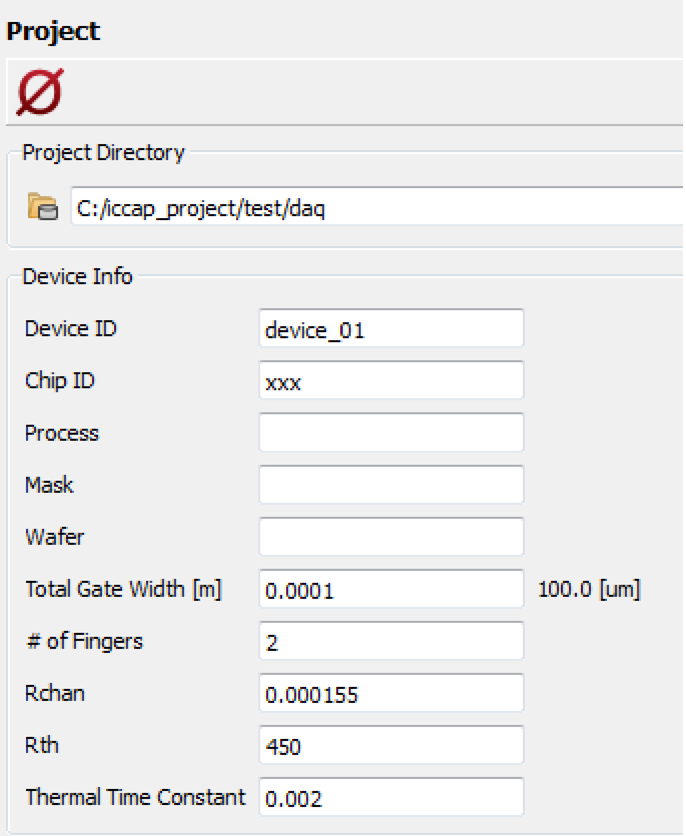


NOTE

The workspace window contain few grayed out fields that are uneditable.

Project Definition

The project step provides an interface, **Project** workspace, to enter information about the device, as depicted in the following figure.



The Device ID and Chip ID are arbitrary strings, but must be defined as they are used in the file names of various data files.

Field	Description
Device ID	An arbitrary string, typically a descriptive name of the device
Chip ID	An arbitrary string used to describe a particular chip, typically an alphanumeric identifier
Process	An arbitrary string, typically a description of the process used to fabricate the device
Mask	An arbitrary string, typically indicates a mask set so as to specify geometry
Total Gate Width	The total width of the device, in meters, used by the model to scale the device
# of Fingers	The number of gate fingers in the device, used by the model to scale parasitic element values
Rchan	The channel resistance, in Ohm-mm, used by the extraction algorithms to partition access resistances

Field	Description
Rth	Thermal resistance in Deg C /-W that describes the self-heating of the device.
Thermal Time Constant	The time constant, in seconds, (Rth*Cth) that describes the thermal behavior of the device.

Project property settings are used by the program for file names and geometry calculations, but also provide a way to annotate the device.

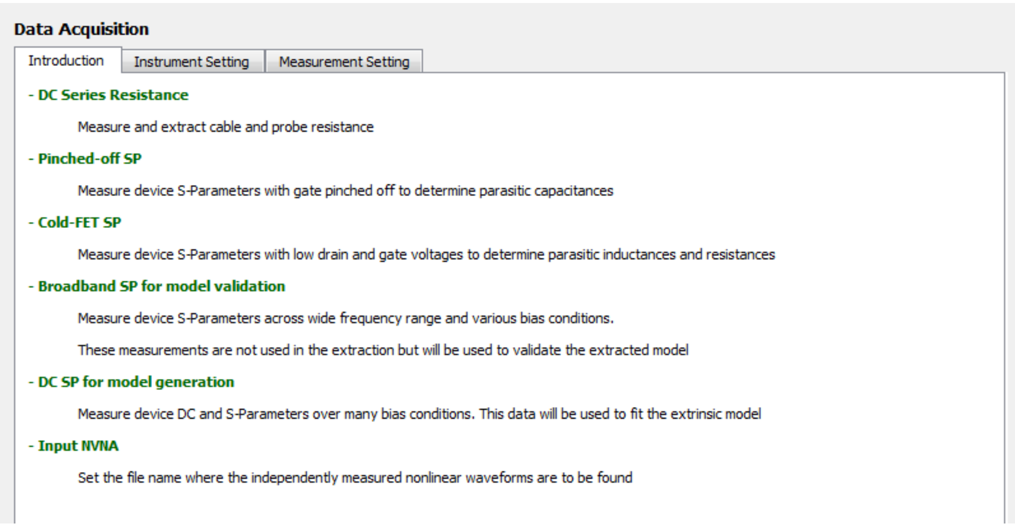
The Process, Mask, and Wafer fields can be used to optionally store information about the device within the project.

The Total Gate Width and # of Fingers must be entered and are used during the extraction to set geometric scaling rules.

Rchan, the channel resistance, is used to help partition the Rd and Rg resistances. The thermal properties of the device are defined by Rth and the overall thermal time constant of the device. This value must be measured independently.

Data Acquisition

The data acquisition step covers all the DC and S-Parameter measurements made by IC-CAP as well as identifying and copying the non-linear waveform data files made by the PNA-X outside of the IC-CAP program. Selecting the ‘Data Acquisition’ page shows three tabs - Introduction, Instrument Setting, and Measurement Setting of the data acquisition step. The following figure depicts the Data Acquisition workspace displaying the short descriptions for various steps.



Instrument Setting

The Instrument Setting tab allows you to add hardware and perform the settings for the connected instrument and then load the saved settings. For information on instrument options, see Supported Instruments. For information on setting up the hardware and connecting the instruments to be measured, see Configuring Hardware and Performing Measurement.

NOTE

For most devices, it is safe to switch the gate bias before the drain bias. Therefore, it is recommended that the SMU connected to the drain be located in a lower slot number.

Instrument Setting Toolbar

The following table describes the main DynaFET toolbar options.

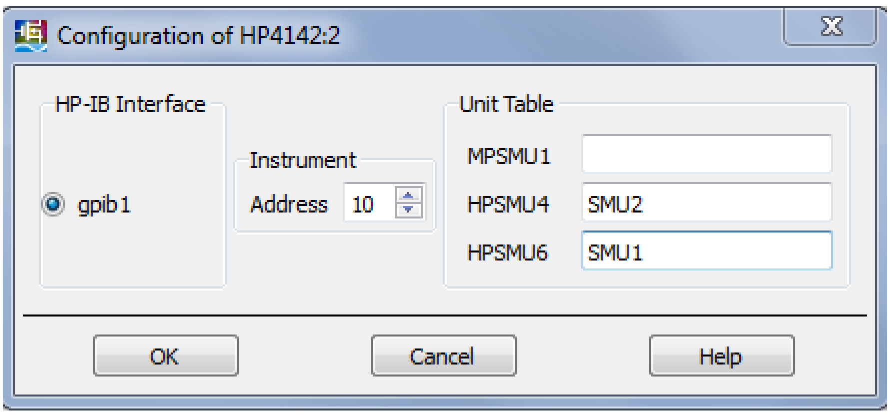
Option	Icon	Description
Open Hardware Window		Opens the hardware setup window
Load instrument Setting		Loads the instrument setting
Save Instrument Options		Saves instrument options

Setting up the Hardware and Instrument Settings

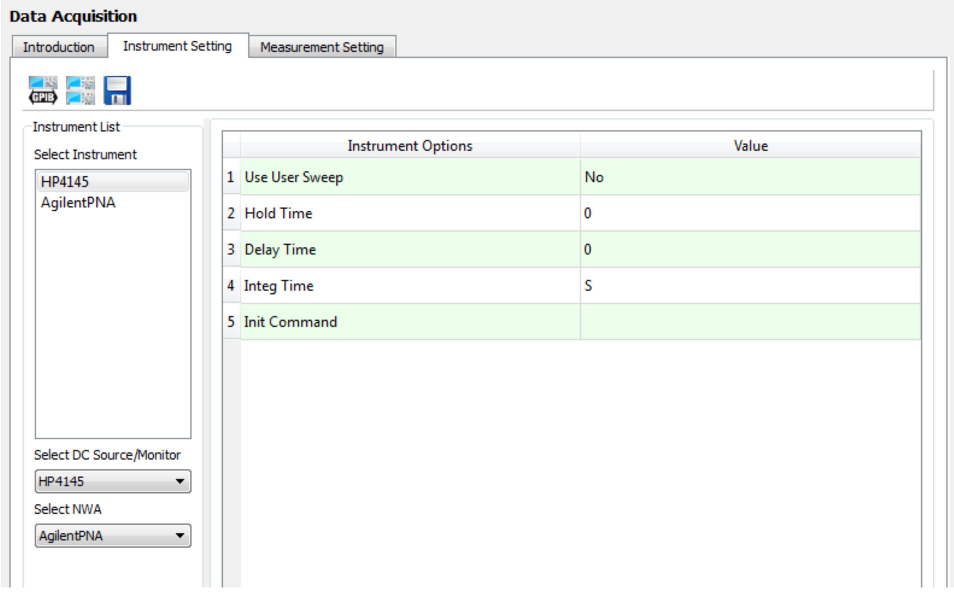
To set up the hardware and load the settings, do the following:

1. Click **Open Hardware Window** ().
   The IC-CAP/Hardware Setup window is displayed using which you can add, build, or configure the connected instrument. For information on the Hardware Setup window, see IC-CAP Design Environment.
2. Add an interface (if required) and rebuild the instrument list using the conventional IC-CAP procedures.
3. Click **Configure...** and set SMU1, SMU2, SMU3 for DC measurement and NWA for SP measurement.

NOTE Drain SMU (SMU2) occurs before the Gate SMU (SMU1).



4. Click **Load Instrument Setting** (), to load the instrument information. The instrument information is displayed under the Instrument List page.
5. Select **DC Source /Monitor** and **NWA** from the available instrument list.

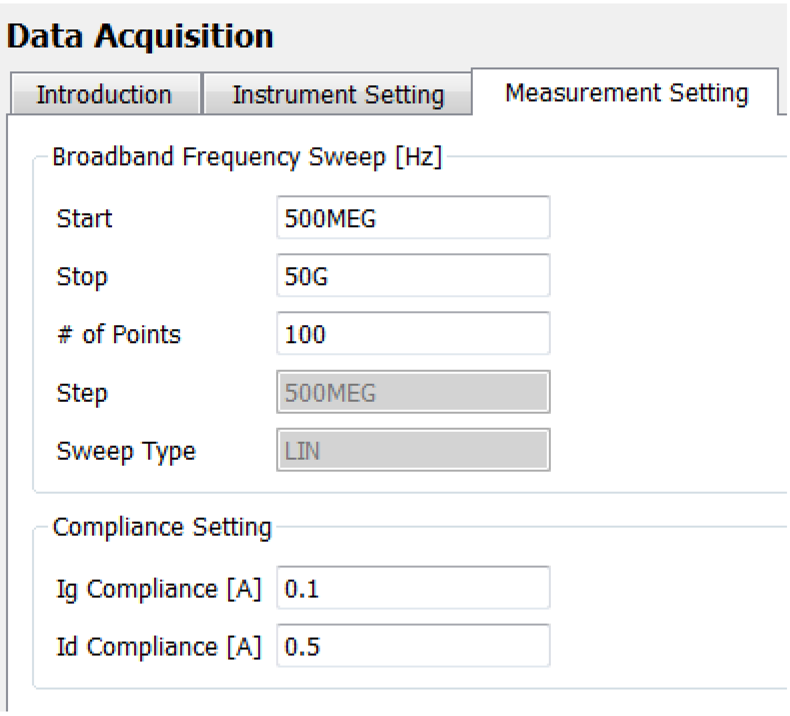


6. Set instrument options for DC and SP measurements, using the DC Source / Monitor and NWA list available on the left side of the window. Then, modify each property value for a particular instrument.

NOTE You can also save these settings in a configuration file for future reference, however, this is optional depending on your requirement.

Measurement Setting

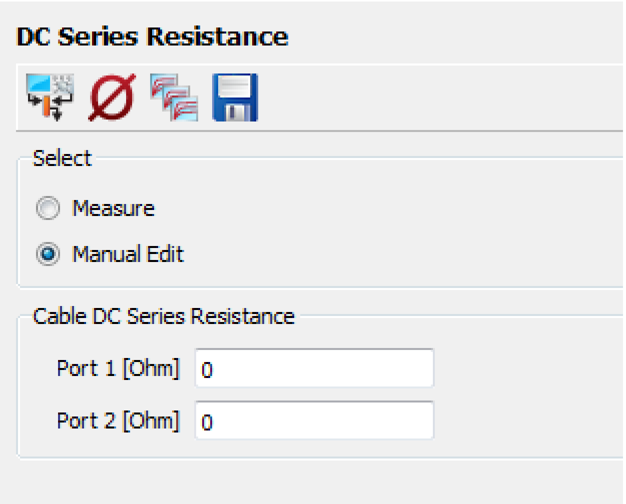
The Measurement Setting tab allows you to set Broadband Frequency sweep and Compliance setting for DC measurement. Some of these values can be overridden at the specific measurement page.



The Measurement Setting tab sets the default values of the frequency sweeps as well as compliance values for the SMUs. The values can be overridden on other pages.

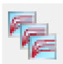
DC Series Resistance

In this workspace, the DC Series Resistance is measured. Alternatively, you can enter (or modify) the resistance. These values are used to remove the effect of the cable resistance from the DC measurements.



Save Instrument Options

The following table describes the DC Series Resistance toolbar options.

Option	Icon	Description
Measure		Performs measurement
Clear Data		Clears the resistance and device geometry data
Display Plot		Opens the plot window
Export Data		Exports data

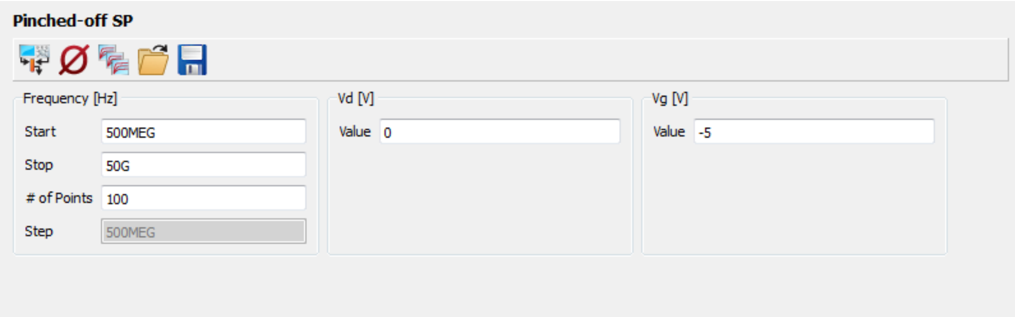
You should lower both probes on a shorted pattern and press the Measure icon. The program then performs the measurement and update the **Cable DC Series Resistance** fields with the measured values. When a measurement is performed, the IC-CAP Status Window provides a Stop measurement button.

The IC-CAP Status Window is modified during a measurement to include an overall stop button. Pressing this during a measurement aborts any active measurement.

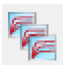
After a measurement, you can display the data with the Plot icon and inspect the I-V curves for correct behavior. The extracted values should be checked to be sure they are reasonable.

Pinched-off SP

In this workspace, the S-parameters used to determine the parasitic capacitor element values are measured (as decpited in the following figure). The values are extracted at one frequency but a broad frequency range allows inspection of how frequency-independent the result is. The drain value should be small or zero while the gate value should be sufficient to pinch off the device without causing breakdown current to flow. The following figure depicts the setup workspace for S-Parameter measurements to be used in Pinched-off extraction.



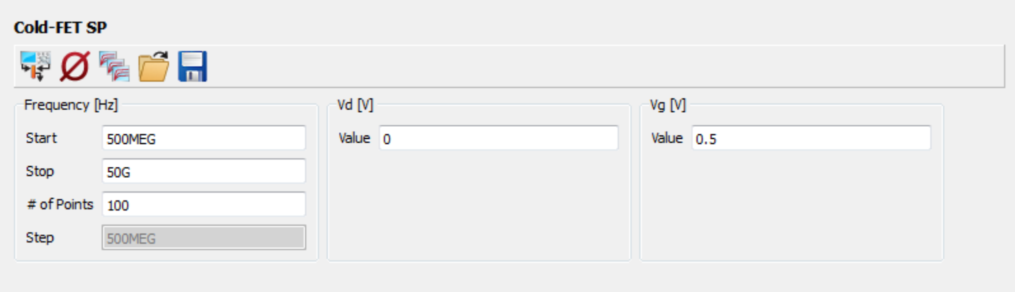
The following table describes the **Pinched-off SP** toolbar options.

Option	Icon	Description
Measure		Performs measurement
Clear Data		Clears the resistance and device geometry data
Display Plot		Opens the plot window
Import Data		Imports data
Export Data		Exports data

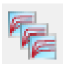
Cold-FET SP

In this workspace, depicted below, the S-parameters for determining the inductance and resistance parasitic element values are measured using the “cold-FET” method [G. Dambrine, A. Cappy, F. Heliodore, and E. Playez, “A new method for determining the FET small-signal equivalent circuit,” IEEE Trans. Microwave Theory Tech., vol. 36, pp. 1151–1159, July 1988].

The S-Parameters are measured while biasing device strong inversion but no current (Vgs >> Vt, Vds ~ 0). The drain value should be small or 0 while the gate value should be sufficient to forward bias the gate while not damaging the device. The values are extracted at one frequency but a broad frequency range allows inspection of how frequency-independent the result is. The following figure depicts the setup workspace for S-Parameter measurements to be used in Cold-FET extraction.

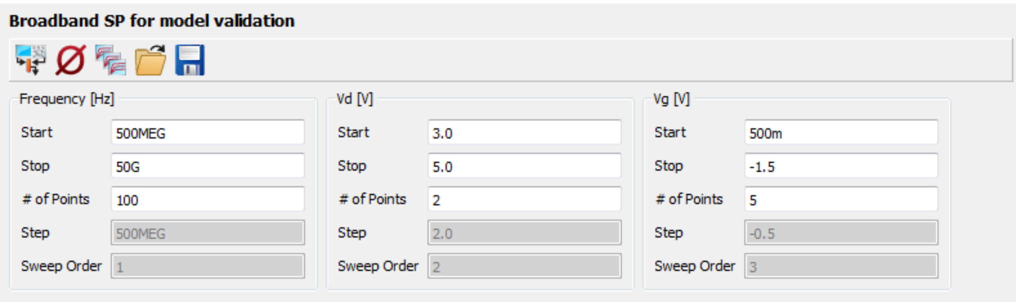


The following table describes the **Cold-FET SP** toolbar options.

Option	Icon	Description
Measure		Performs measurement
Clear Data		Clears the resistance and device geometry data
Display Plot		Opens the plot window
Import Data		Imports data
Export Data		Exports data

Broadband SP for model validation

In this workspace, depicted below the measurements of broad-band S-parameter at selected bias points for validation are taken. These measurements are not be used in the extraction or training aspects of the model fitting. It is best to choose a slightly different frequency grid and bias points than what is used in the model generation procedure. The following figure depicts the workspace for broadband S-Parameter measurements used to illustrate how well model fits.

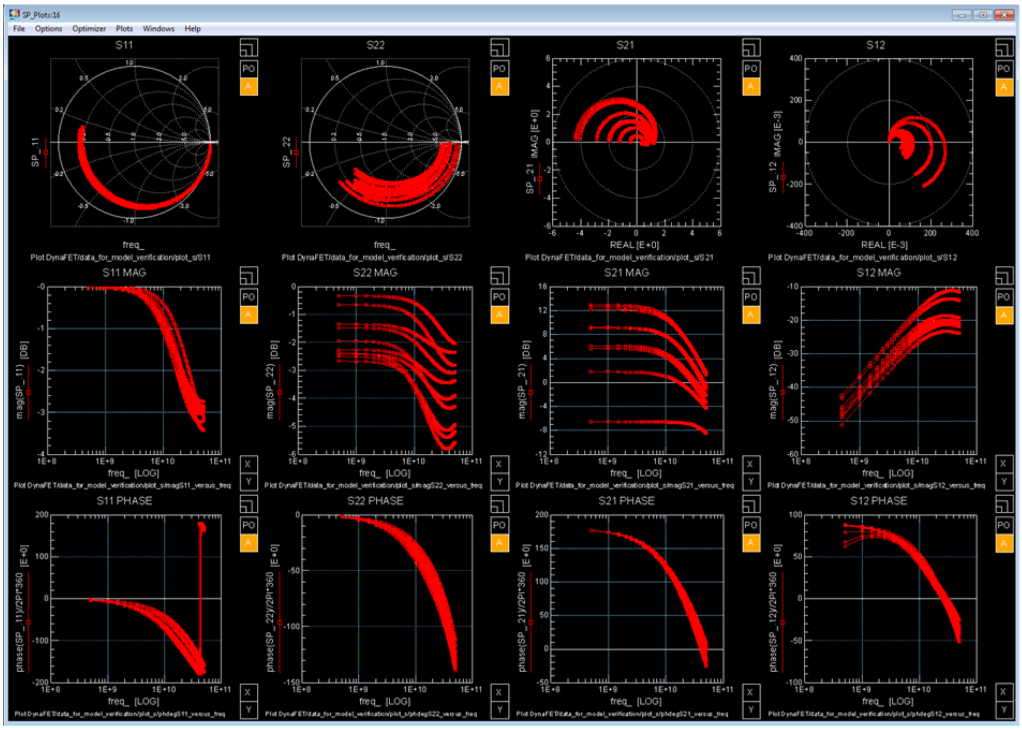


The following table describes the **Broadband SP for model validation** toolbar options.

Option	Icon	Description
Measure		Performs measurement

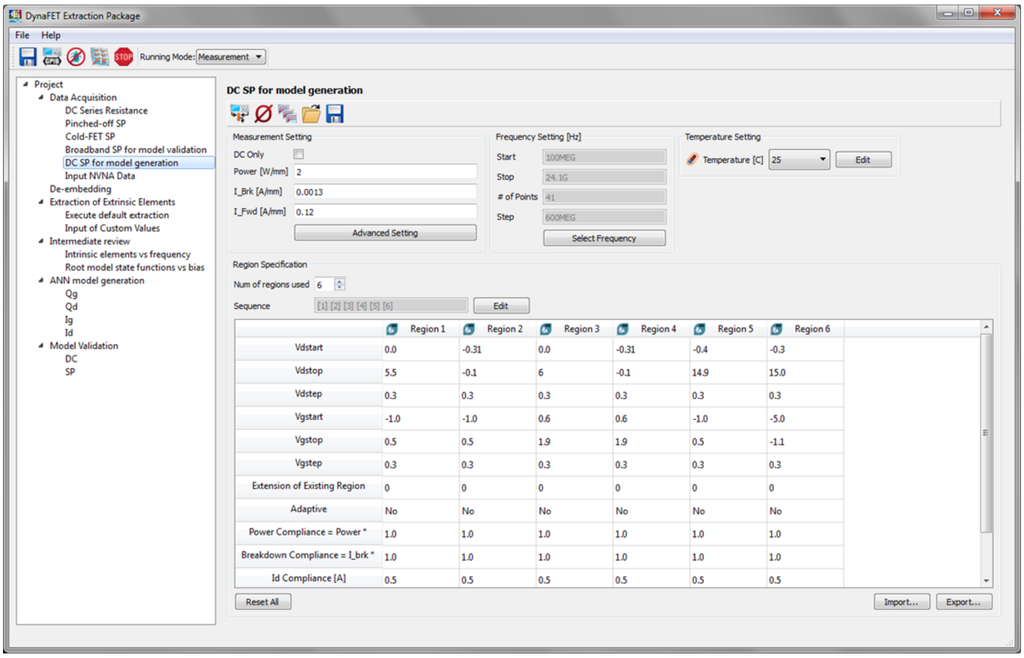
Clear Data		Clears the resistance and device geometry data
Display Plot		Opens the plot window
Import Data		Imports data
Export Data		Exports data

The Display Plot displays the data for inspection. The following figure depicts the S-Parameters measured over various bias conditions, to be used for model validation:



DC SP for model generation

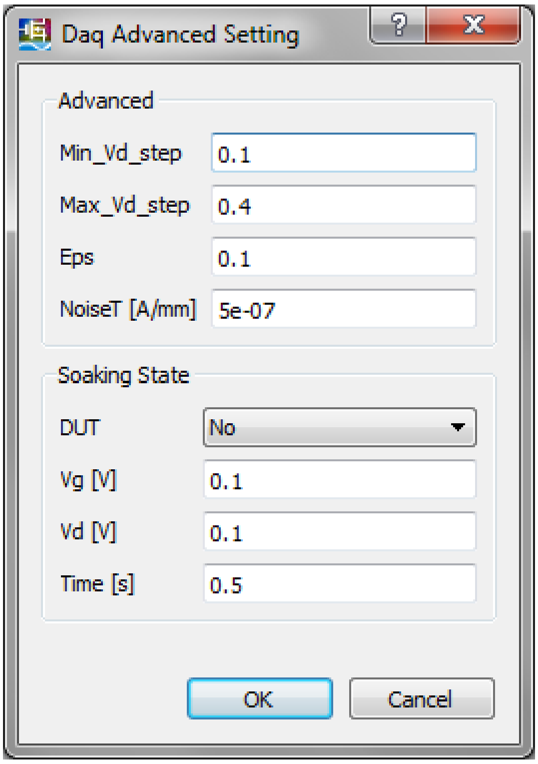
The DC SP for model generation workspace defines the settings for measuring the DC, S-Parameters for model generation. The measurements are made across the full safe operating region (SOR) of the device. The setup allows to select the bias sweeps in such a way that full coverage of the SOR is possible. For each measurement, the program first calculates the possibility that the device's compliance settings is reached. If that's the case, the program aborts the measurement for that segment and continues based on the setting of the Abort Type. The following figure depicts the workspace user interface for setting up the DC and S-Parameter linear measurements used during model generation.



The power compliance settings are described in the following table.

Setting	Description
Power	**Power Dissipation**: Power Dissipation is the power that is converted to heat and then conducted or radiated away from the device. It is a measure of the rate at which energy is dissipated from an electrical system. For power dissipation, enter the total maximum power dissipation value in the Power (W/mm) field. Typically, this should be 2 Watts per mm of gate width for devices up to 1 mm.
I_Brk	**Current Breakdown**: Enter the reverse gate current breakdown value in the I_Brk(A/mm) field. Typically, 2 mA per mm of gate width.
I_Fwd	**Conducting Gate Current :**Enter the forward conducting gate current in I_Fwd (A/mm). Typically, 200 mA per mm of gate width.
SMU_Compliance	**SMU Compliance: E**nter the drain SMU maximum current compliance value in Amps. This is the maximum output current capability of the source/monitor unit (SMU) supplying the drain terminal.

Additional settings can be made using the **Daq Advanced Setting** Dialog box. Click **Advanced Setting**. The **Daq Advanced Setting** dialog box is displayed as depicted in the following figure.



The property values of advanced settings are described in the following table:

Advanced Setting	Description
Min_Vd_step	**Min_Vd_step:** Defines the step size between densely-spaced measurements in Vd space in Min_step(Vd) field
Max_Vd_step	**Max_step**: Defines the measurement step size, where you expect the data to be relatively linear and you do not need densely-spaced measurements in Vd space.
Eps	**Epsilon:** Predicts the next point to be measured. Therefore, it controls the automatic data acquisition step size. Increasing the value of epsilon, decreases the sensitivity of the automatic data acquisition.
NoiseT	**Noise Threshold:** For Noise threshold, enter a current threshold value in the NoiseT(A/mm) for both drain current and gate current below, which you do not need closely-spaced measurements. The automatic step-size algorithm is not used below the Noise threshold value, where measured currents are sampled at the maximum step size. A typical value is 500n amps per mm. Note:Do not set Noise threshold so small that it measures numerous points below pinch-off, wasting measurement time and storage capacity. Also, do not make it too high.

Soaking state property settings in the **Daq Advanced Setting** dialog box is listed below:

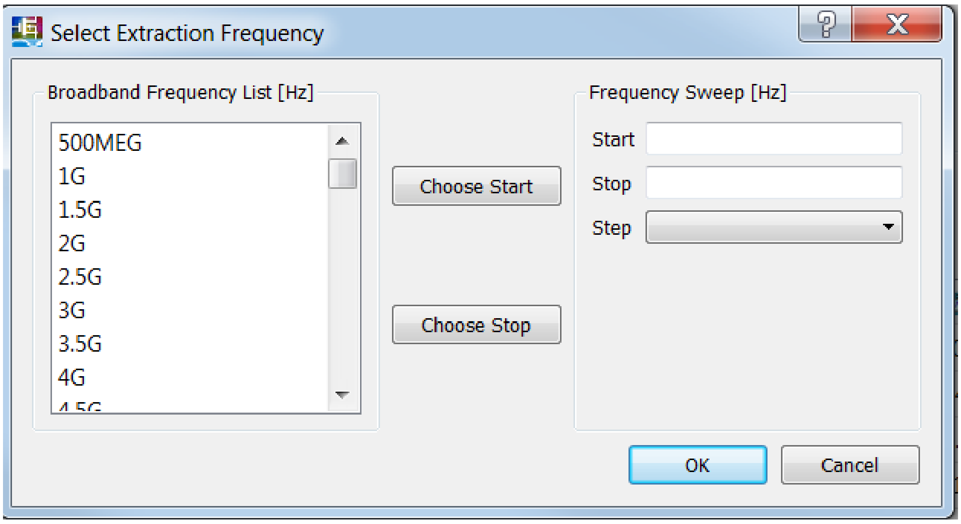
Soaking State Setting	Description
DUT	**DUT**: Determines whether or not to apply a soaking bias [No, Between each Vgs, Between regions]
Vg	**Vg** : Defines the gate bias to apply for the soak [V]
Vd	**Vd** : Defines the the gate bias to apply for the soak [V]
Time	**Time:** Denotes the time to apply the bias before measuring [Sec]

The **DC Only** check box skips the S-Parameter measurements and speed up the measurement time. This is useful to check for complete coverage of the device operating region.

The **Frequency Setting** sets the range and points, where S-Parameters is measured.

To set the frequency:

1. Select Frequency. The Select Measure Sequence dialog box is displayed.

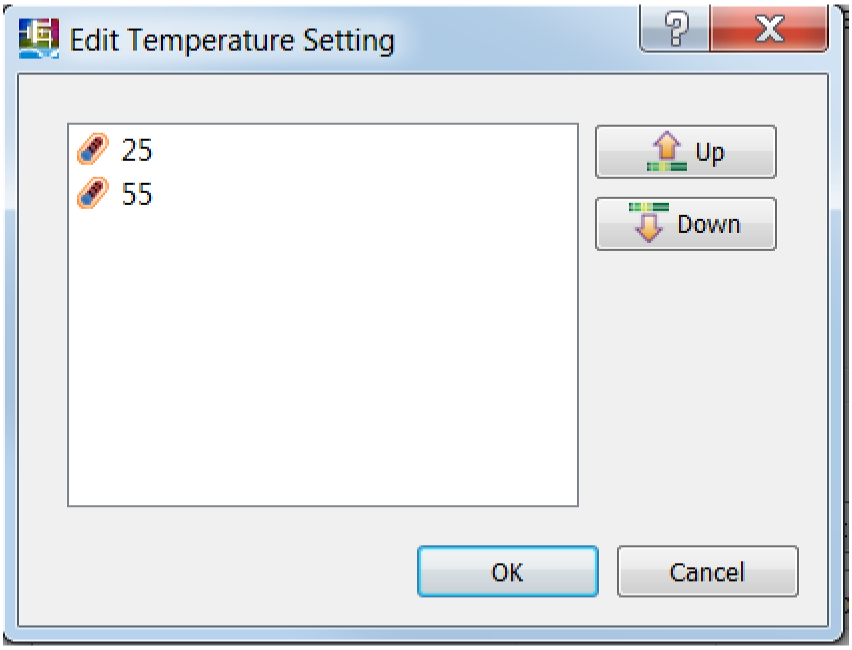


2. Set the extraction frequency.
3. Click OK.

The **Temperature Setting** indicates the temperature at which the measurements are made. The program does not control the temperature; this setting is so that the program embeds the temperature in the data files. The temperatures can be modified by pressing the **Edit Temperature** button and then selecting the values from the dialog box as depicted in the following figure.

To set the temperature of the data acquisition:

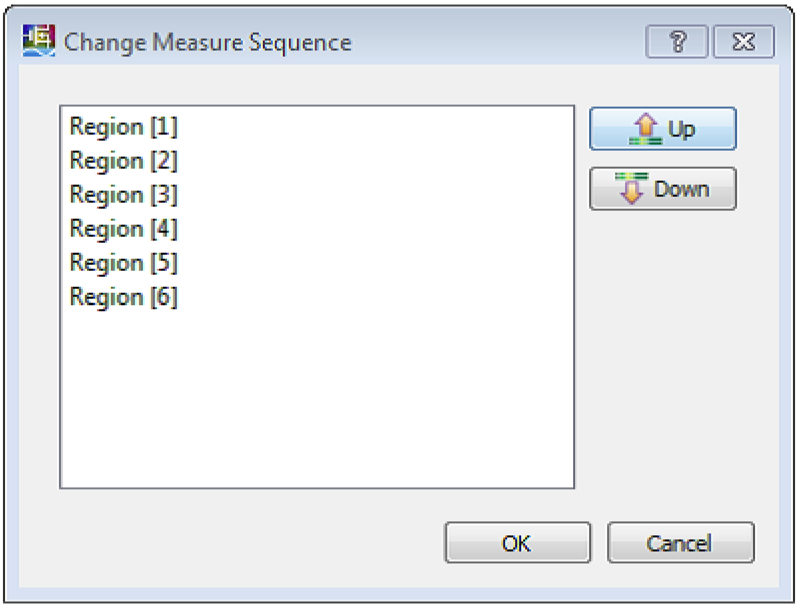
1. Click **Edit** under the Temperature Setting pane. The **Edit Temperature Setting** dialog box is displayed.



2. From the **Edit Temperature Setting**dialog box, select the required button:
   a. Click **Add**, to embed a temperature in the data file.
      or
   b. Double-click an existing value, to edit and override the existing value.
   c. Click **Up** or **Down,** to change the temperature sequence.
3. Specify the temperature and click **OK**. This updates the dialog box with the new or updated temperature setting.

To set the measurement sequence, do the following:

1. Click **Edit** under the Region Specification pane. The **Change Measure Sequence** dialog box is displayed.



2. Click **Up** or **Down,** to change the measurement sequence of a region.
3. Click **OK**.

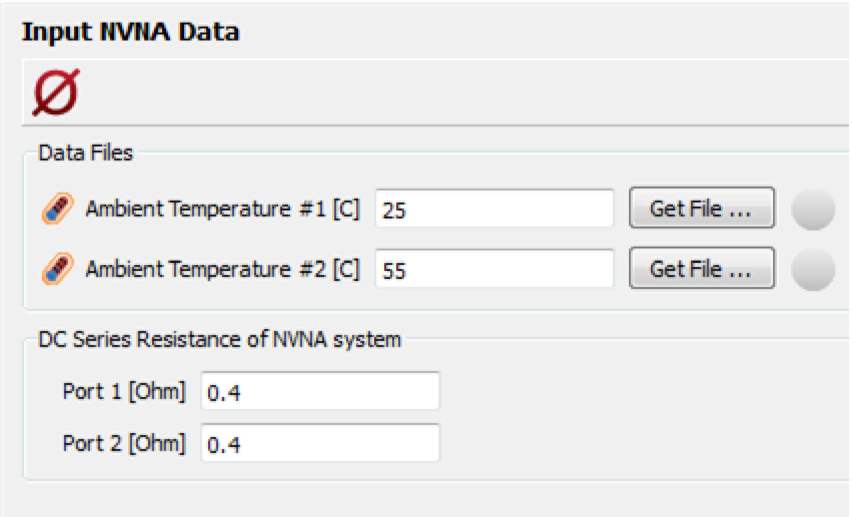
The following table describes the region specification fields and the respective values.

Vdstart, Vdstop, Vdsstep, Vgstart, Vgstop, Vgsstep	**Defines the region in *Vds-Vgs* space.**
Extension of Existing Region	Combines the data as one region (for plotting)
Adaptive	If set to **Yes**, the adaptive sampling is used. If set to **No**, then there is a fixed step size.
Breakdown Compliance = I_brk*	Scales the default value of I_brk by multiplying I_brk by this value.
AbortType	The possible values are: • **F**: Denotes the Exit region after **F**irst segment having no new data • **T**: Denotes the The **T**erminate extraction, if no data found in the region. • **M**: Denotes the **M**ove to next segment, if no data is found in the current one.

The settings can be exported/imported or reset to default values by using the appropriate button at the bottom of the workspace.

Input NVNA Data

The program copies the existing NVNA data files to the local project for use during the model generation. Two data files (at 2 temperatures) must be identified. The temperature value is manually entered for each of the files and you can keep a track of what temperature each NVNA file is associated with. Pressing **Get File...** displays the browser dialog to select a file. When the file is copied over, the circle color changes from grey to green, indicating that the file is copied successfully.



You must manually record the temperature associated with each measured data file and the DC resistances of the measurements.

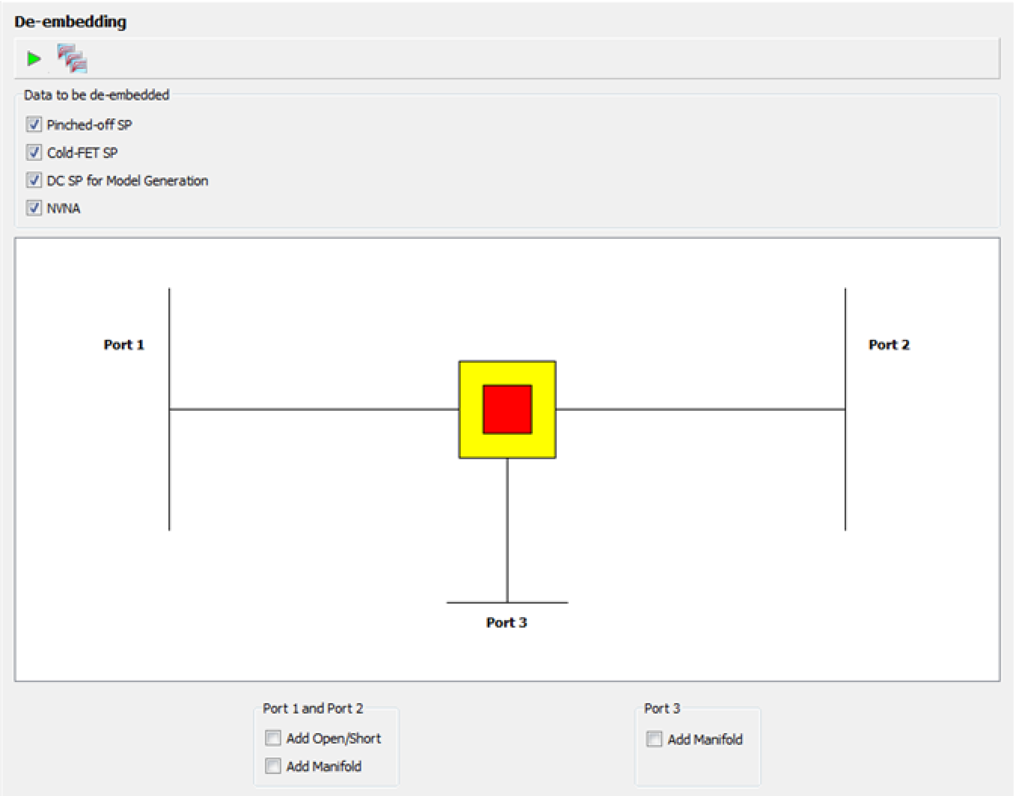
As the NVNA measurement is a different equipment setup, the DC cable resistances are different than those measured by the IC-CAP DynaFET program during the DC and S-Parameter values. Enter the values of the cable resistances for each port in the appropriate field.

De-embedding

The De-embedding step removes the extrinsic parasitic influence from the measured data. The DynaFET modeling package provides following types of de-embedding methods:

- Open/Short
- Manifold

The following figure depicts the De-embedding workspace:



De-embedding provides two operations – execute and plot.

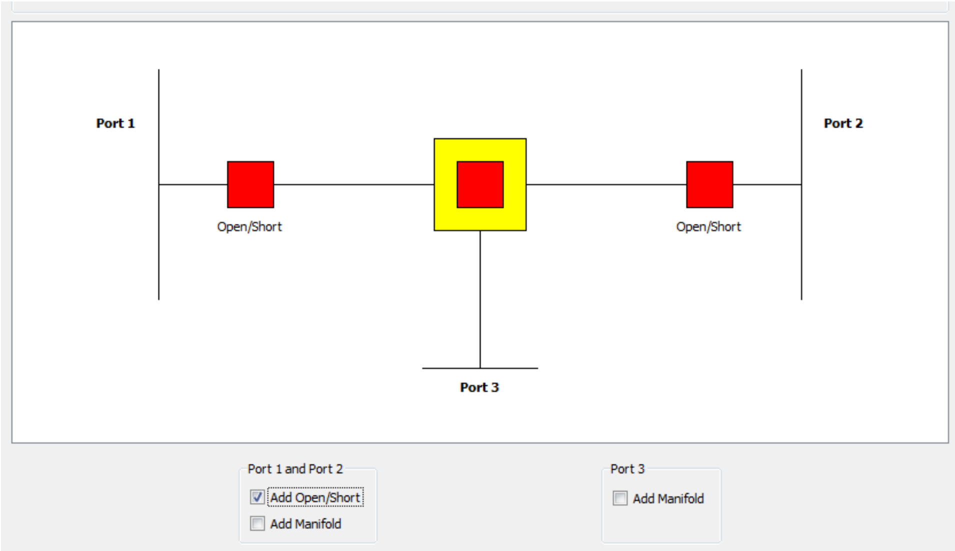
The program allows disabling of de-embedding certain datasets. This is provided to save time if de-embedding is repeated with changes and it is not necessary to de-embed all the data. However, you can keep track of whether data needs to be updated after any changes. The two methods of de-embedding are described below.

NOTE Although no de-embedding method is used by the model, however, it is mandatory to execute De-embedding ( ). First clear **Port 1 and Port 2**, and **Port 3** check boxes, and then click .

Open/Short De-embedding

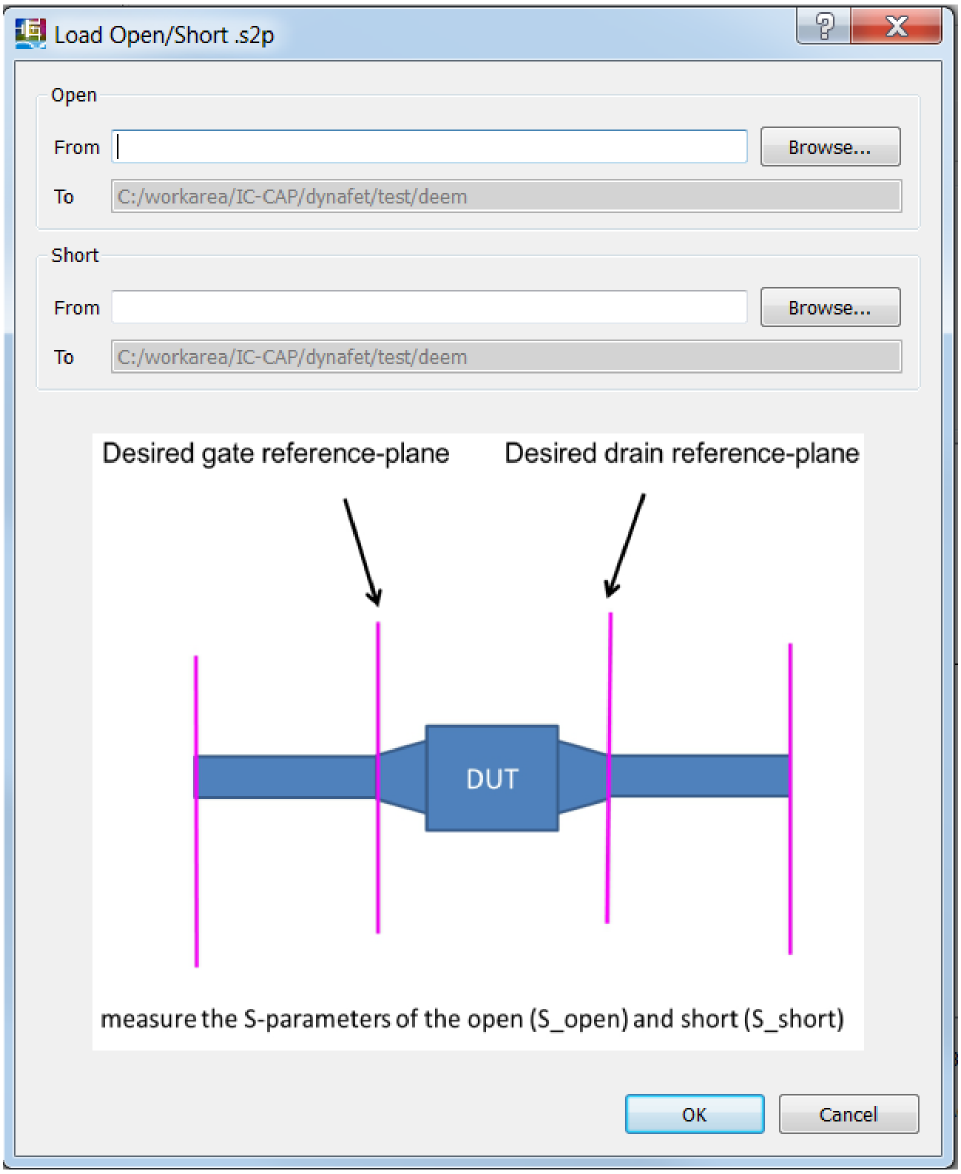
To execute de-embedding:

1. Select **Add Open/Short** option for **Port 1** and **Port 2** in the De-embedding workspace area. Red symbols are added to the schematic.



2. Double-click the **Open/Short** rectangular box for Port 1.

   The **Load Open/Short .s2p** dialog box is displayed.



3. Specify the locations for the files that contain the Open and Short S-Parameter data files (.s2p).
4. Click OK.
5. Repeat steps 1 to 4, for Port 2.
6. Click OK.

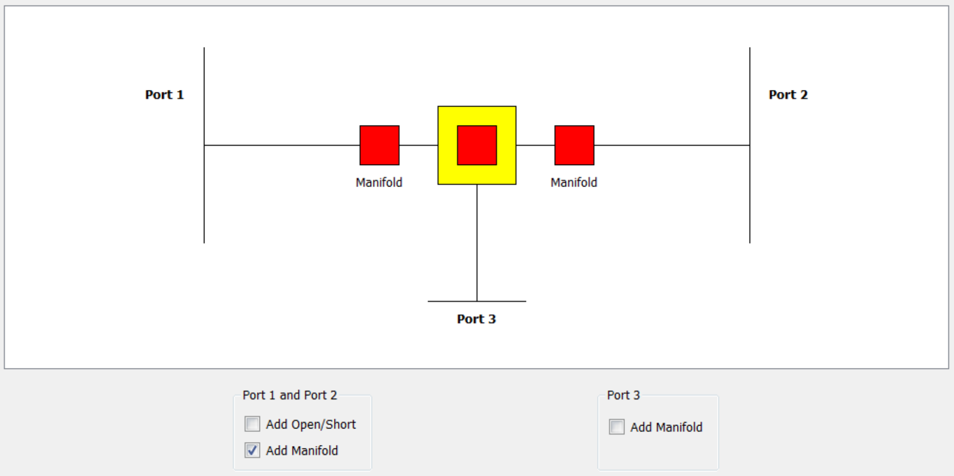
The following equation is used for calculating Open/Short de-embedding.

```
[T1] = [Ti]^-1 * [Tmeas] * [To]^-1
[T1] -> [Z1]
[Sv] -> [Zv]
[Zdut] = [Z1] - [zv]
[zdut] -> [Sdut]
where,
[Tmeas] is the T-parameters of device measured.
[Ti] is the T-parameters of pad/taper measured at Port1.
[To] is the T-parameters of pad/taper measured at Port2.
[Sv] is the 1-port S-parameter at Port3.
[Sdut] is the De-embedded S-parameters of the device.
```

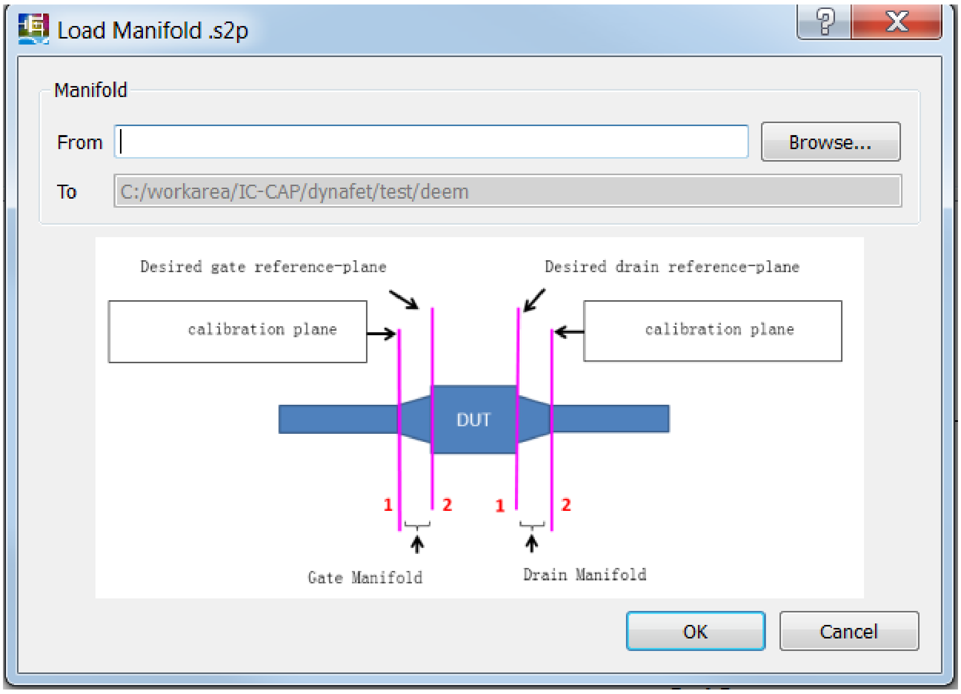
Manifold De-embedding

Manifold de-embedding can be done on Port 1 and 2, or Port 1, 2, and 3.

1. Select Add Manifold option for **Port 1** and **Port 2** in the De-embedding workspace area. Red symbols are added to the schematic.



2. Double-click the Manifold Port 1 box.
   The **Load Manifold .s2p** dialog box is displayed.



3. Specify the manifold S-parameter file.
4. Click **OK**.
5. Double-click the Manifold Port 2 box. The **Load Manifold .s2p** dialog box is displayed, as depicted in step 2.
6. Specify the manifold S-parameter file.

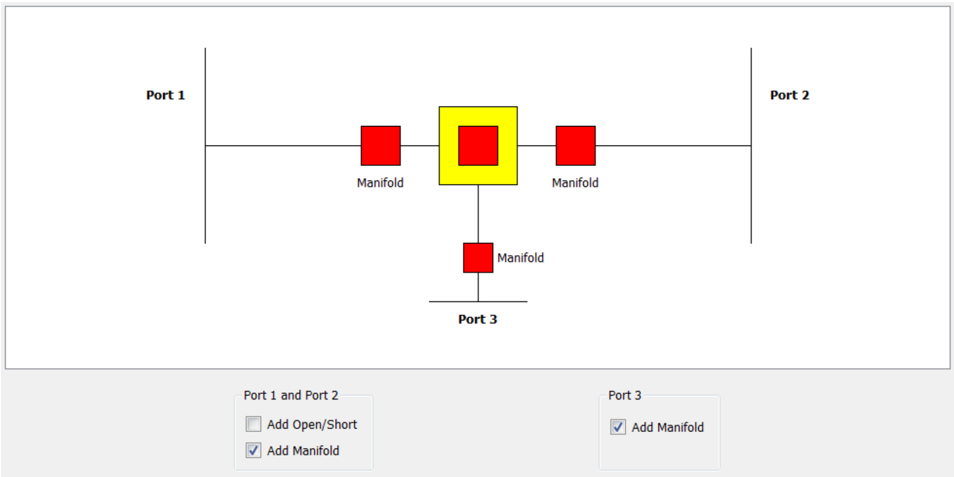
NOTE If the drain manifold is measured or simulated in the opposite fashion, check the **reverse** box (note the change in numbering on the Drain Manifold).

7. Click OK.

   In this case, the following equation is used to calculate the manifold de-embedding.

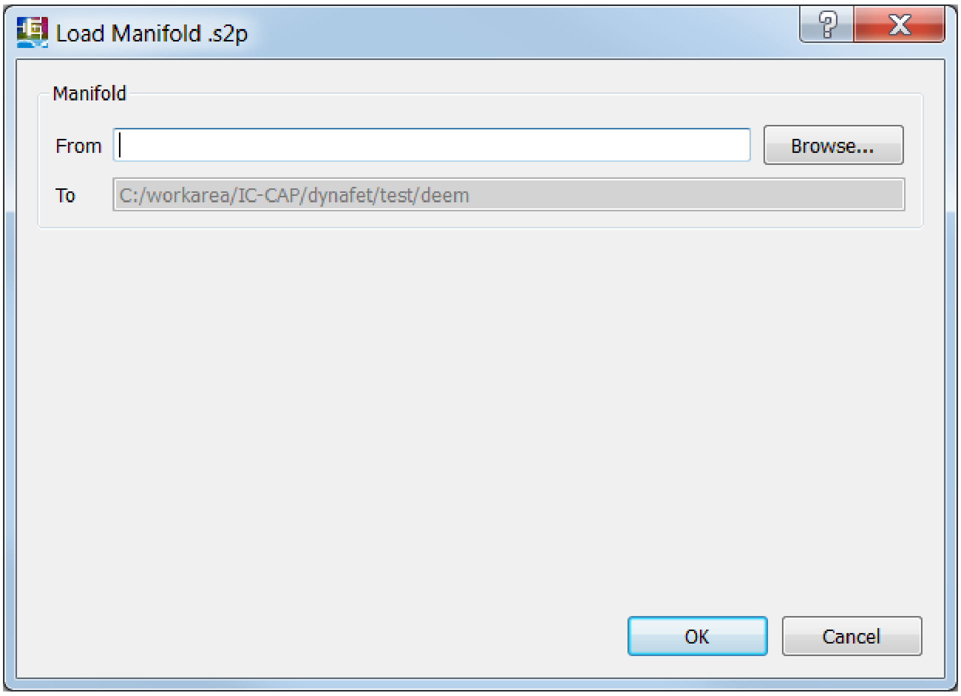
```
[Tdut] = [Ti]^-1 * [Tmeas] * [To]^-1
 where,
[Tmeas] is the T-parameters of device measured.
[Ti] is the T-parameters of pad/taper measured at Port1.
[To] is the T-parameters of pad/taper measured at Port2.
[Tdut] is the De-embedded T-parameters of the device.
```

8. If de-embedding of the Port 3 manifold is required, select Add Manifold option for Port 3 in the De-embedding workspace area. Red symbols are added to the schematic



9. Double-click the Manifold Port 3 box.

   The Load Manifold .s2p dialog box is displayed.



10. Specify the manifold S-parameter file.

11. Click OK

In this case, the following equations are used to calculate the manifold de-embedding.

```
[T1] = [Ti]^-1 * [Tmeas] * [To]^-1
[T1] -> [Z1]
[Sv] -> [Zv]
[Zdut] = [Z1] - [zv]
[zdut] -> [Sdut]
where,
[Tmeas] is the T-parameters of device measured.
[Ti] is the T-parameters of pad/taper measured at Port1.
[To] is the T-parameters of pad/taper measured at Port2.
[Sv] is the 1-port S-parameter at Port3.
[Sdut] is the De-embedded S-parameters of the device.
```

Extraction of Extrinsic Elements

A DynaFET model parasitic circuit consists following two parts:

- Extrinsic Part: The parasitic extraction process in the DynaFET modeling tool extracts the parameters of the extrinsic part.
- Intrinsic Part: The ANN Model Generation process in the DynaFET modeling tool determines the characteristics of the intrinsic part. For details, see ANN Model Generation.

This step determines the extrinsic element values. The intrinsic data is generated by removing the extrinsic circuit contribution from the measured data. The ANN training is performed on the intrinsic part of the data.

The default extraction, automatically determines the extrinsic circuit values in two steps, the capacitance extraction and the inductance and resistance extraction. Each step allows you to inspect the results and overwrite the values, if required.

Execute Default Extraction workspace comprises three tabs:

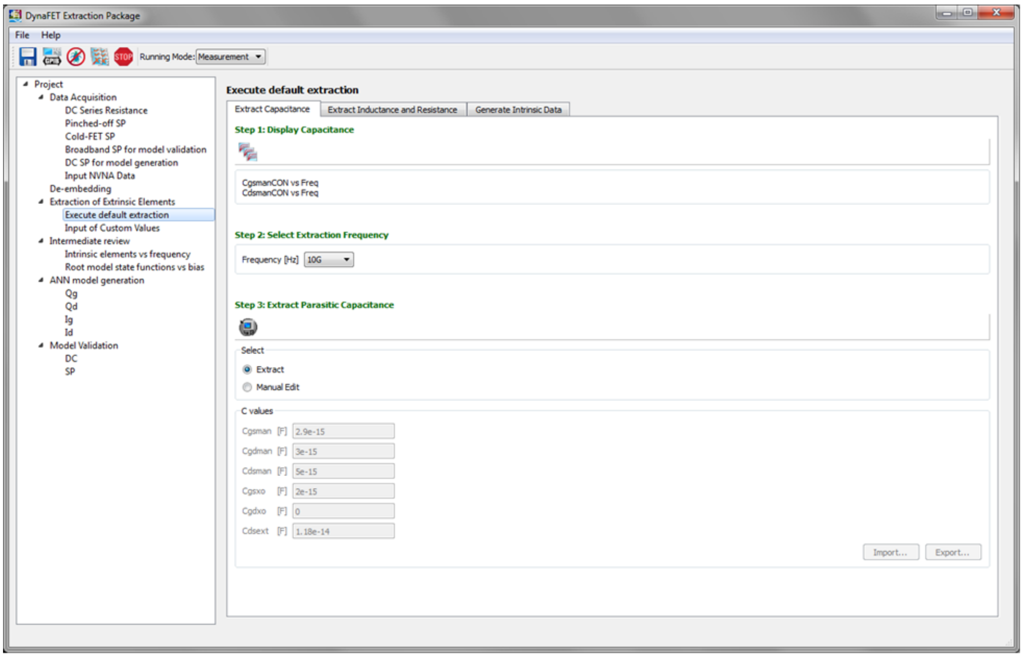
- Extract Capacitance
- Extract Inductance and Resistance
- Generate Intrinsic Data.

Each tab is detailed in the subsequent content.

Extract Capacitance

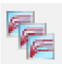
Select Execute Default Extraction from the DynaFET Extraction Package window.
The model with default values is displayed in the workspace area. The page is ordered by the typical steps. First the capacitance values are displayed versus frequency. This allows you to select a frequency value where the capacitance is constant, using this value to select the Extraction Frequency in step 2. The final step is to Extract the values. The extracted values are displayed in the table and can be manually edited, if desired.

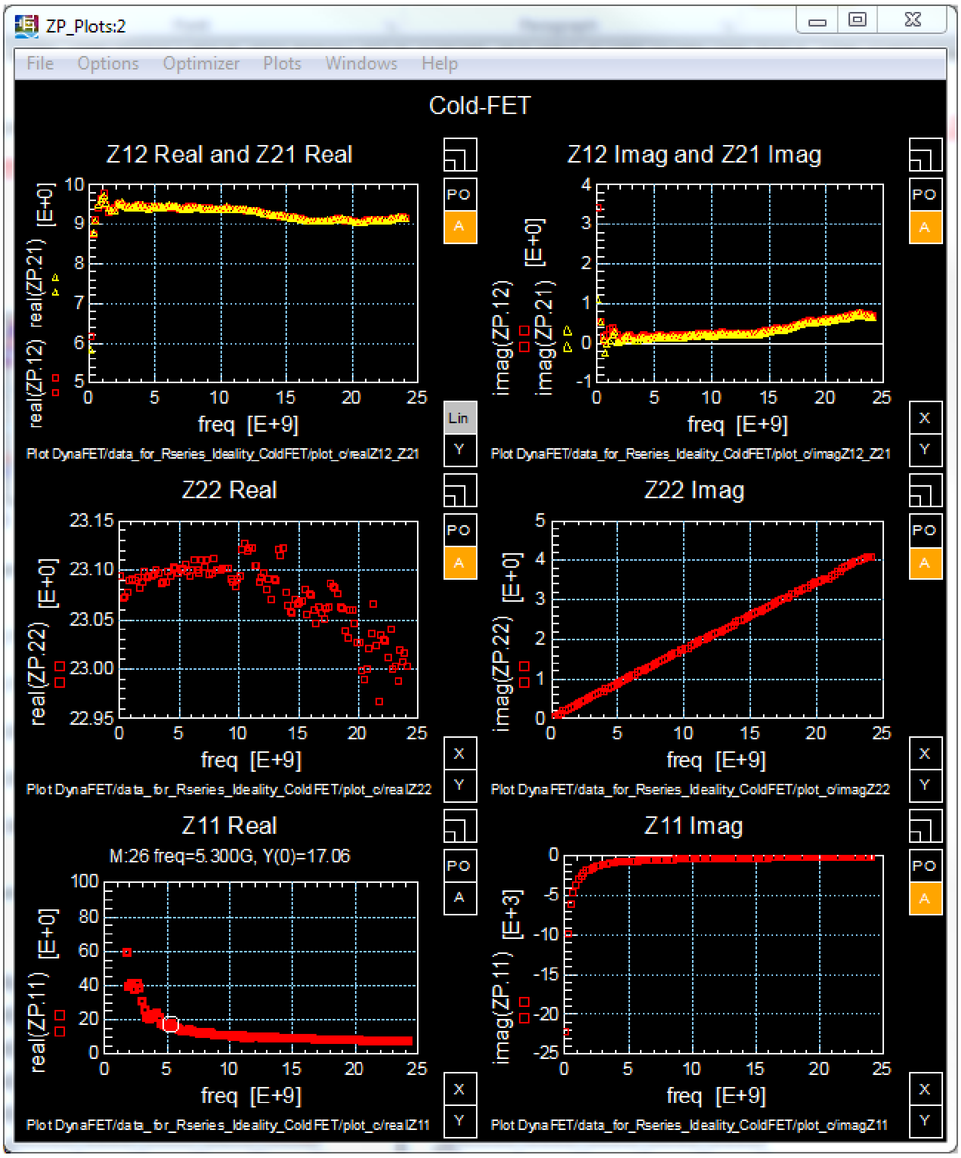
The C values can be exported/imported to a file by selecting the appropriate buttons.



Extract Inductance and Resistance

In this step, the Inductance and Resistance values are extracted from the data. The frequency range, where the data is extracted is limited to the range set in Step 1.

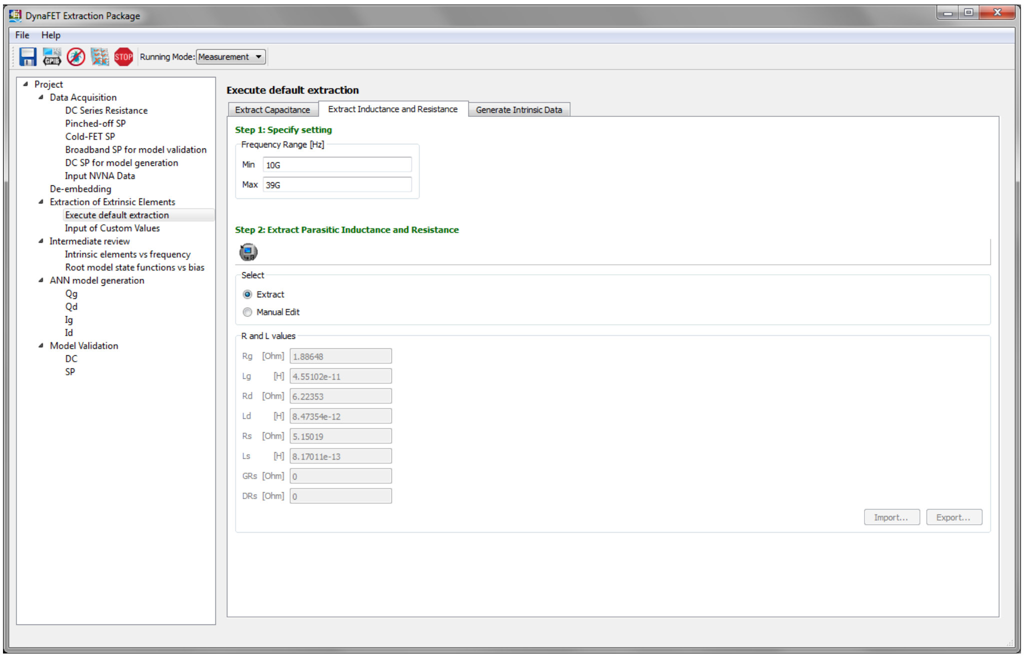
1. Click  to plot the Z-parameters versus frequency, as depicted below.



2. Select a frequency range over which the real parts of Z12 are relatively independent and where the imaginary terms of Z11/Z22 are linear with respect to frequency. If these assumptions are incorrect, you need to try other bias settings for the Cold FET measurements.

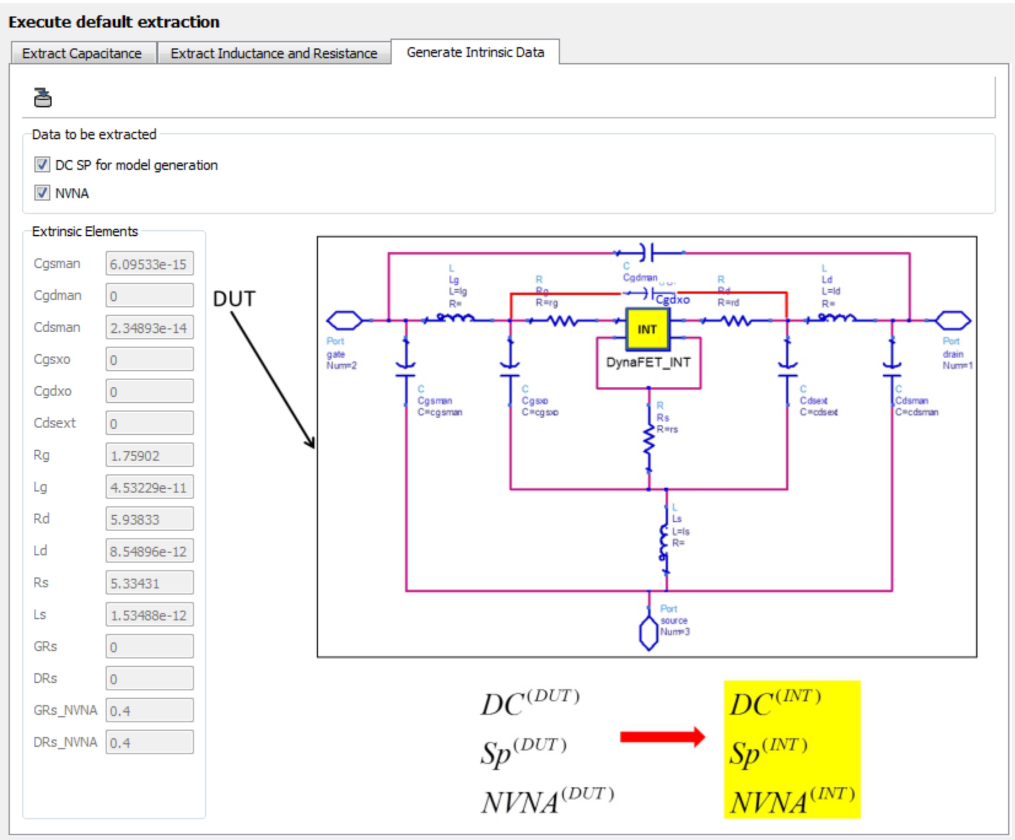
Step 2 executes the automatic extraction and displays the values in the fields, where they can be manually edited.

The R and L values can be exported/imported to a file by selecting the appropriate buttons.

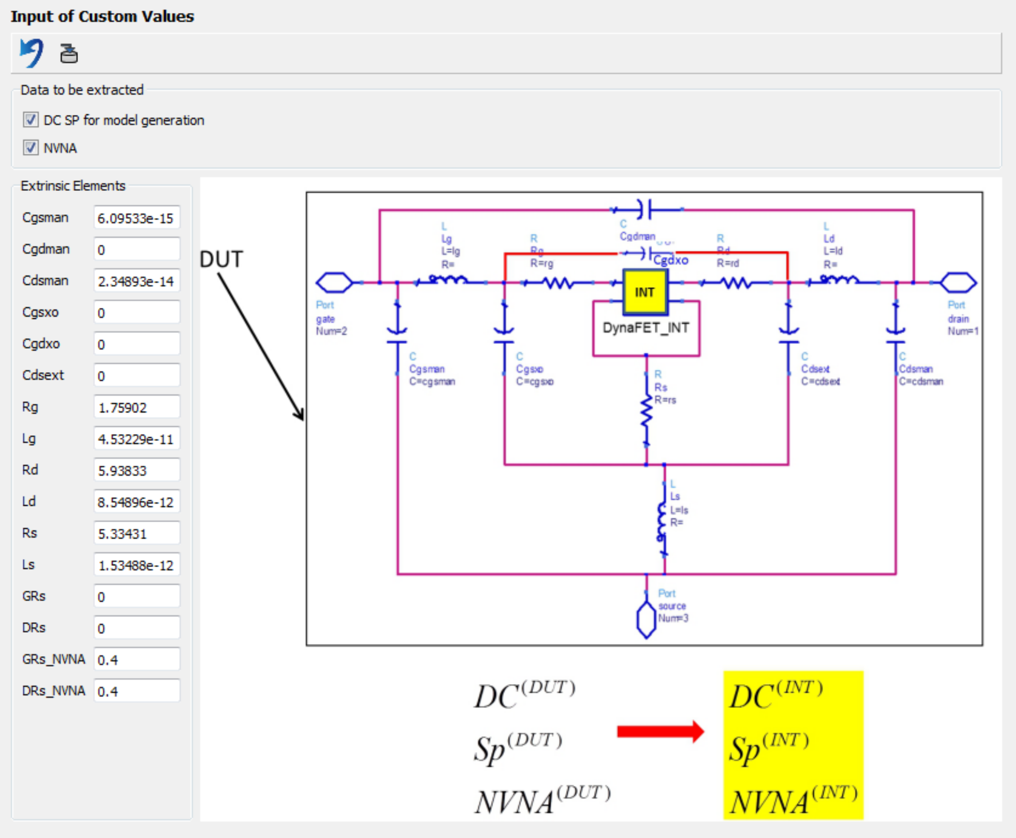


Generate Intrinsic Data

When the extrinsic elements are satisfactorily determined, the Generate Intrinsic Data page is used. The workspace lists a summary of the extracted values. It also provides the option of only generating a subset of the data to reduce the generation time. It is up to you to keep track if changes to the data require a re-generation.



Generate Intrinsic Data workspace showing circuit element values and topology. Alternatively, you can select the **Input of Custom Values** page.



Input of Custom Values page allows manual override of all the values. Values can be set to any arbitrary value (for instance, determined by an alternative fitting process). Also, by setting element values to zero the topology can be modified effectively.

You need to assess the quality of the fit to the model and potentially modify the parasitic topology or extracted values. This assessment is done at each extraction step and after the intrinsic model fitting. The next step (intermediate review) provides an early feedback describing how well the model is fitting the measured data. After the intermediate review using the Root model, you are required to modify the parasitic values.

Intermediate Review

In the intermediate review step, a Root model is generated. The results can be used as a quick method to ascertain the quality of the data without the relatively long ANN training cycle.

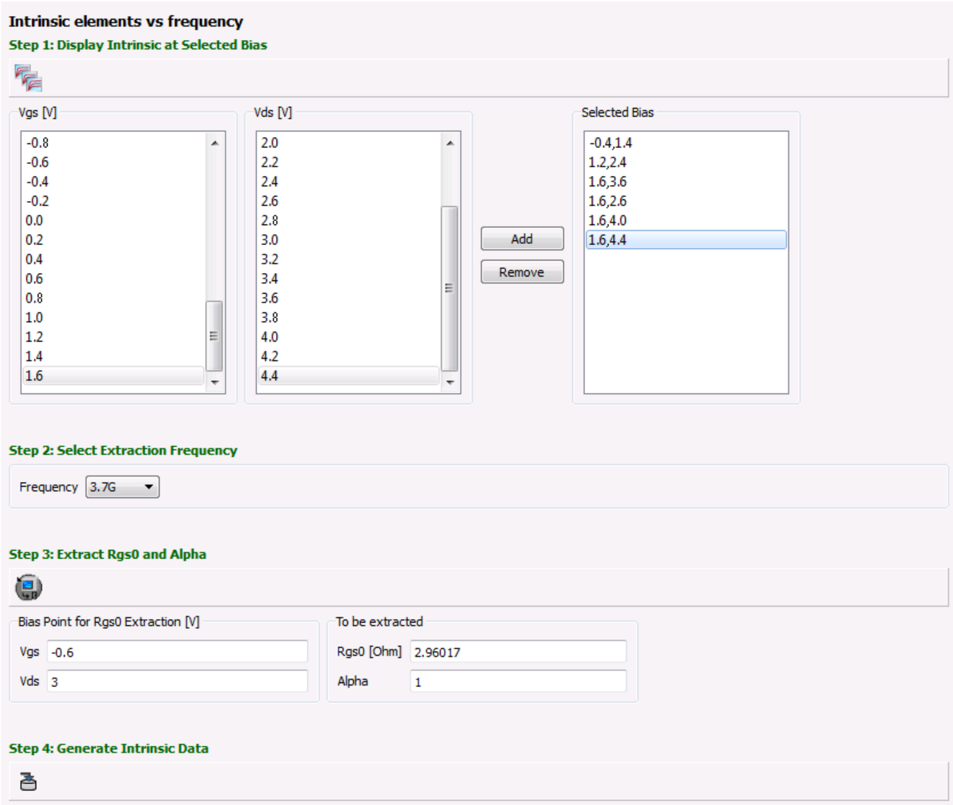
NOTE It is recommended to execute this step before NVNA measurement of GaN devices outside IC-CAP, if you have a concern that device might degrade during the DC and S-Parameter measurement sequences.

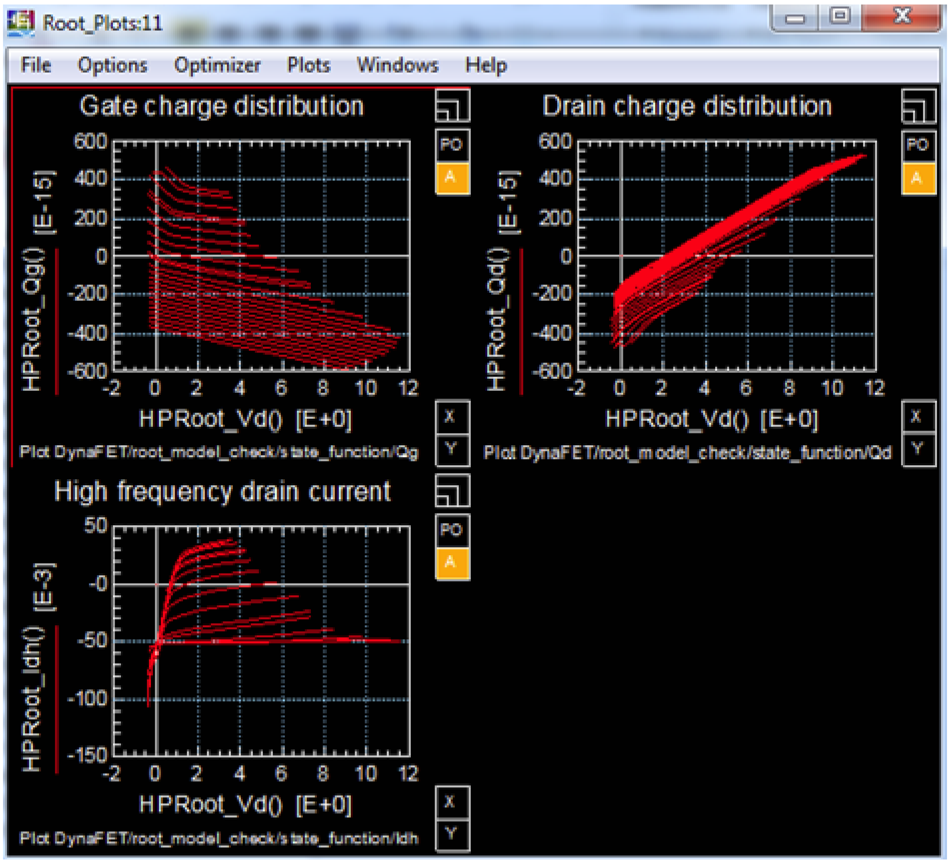
To perform intermediate review:

1. Select **Intrinsic elements vs frequency** from the DynaFET Extraction Package window.
2. Select the value of **Vg** and **Vd**.
3. Select the extraction frequency.
4. Set the bias point for Rgs0 extraction (typically the application operating point)

5. Click to extract RGs0 and Alpha values. Review the extracted data and modify (if required) before proceeding to the next step.

6. Generate the intrinsic data.





7. Click the **Check** icon () to perform data sanity check.

   Review the data and check for discontinuities, unexpected behavior or changes indicating that the device is degraded or is not functioning as expected. If issues are found, they should be resolved prior to NVNA measurements or ANN model generation.

   The review determines if the problem is:

   - either with the measured data (noisy) or extracted values, or
   - either due to incorrect selection of bias or frequency ranges where a particular parameter was fit or because the model topology does not reflect the topology of the actual device.

   In such scenarios, you need to determine why the model does not fit and take appropriate steps to determine the parasitic values. For instance, poor fitting of only S11 indicates an error with the input (gate) resistance or the gate-source capacitance.

ANN Model Generation

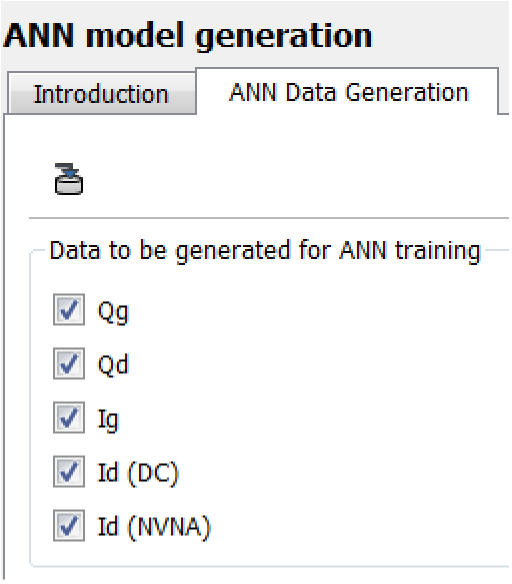
Artificial Neural Network (ANN) is an information processing system with their design inspired by the studies of the ability of the human brain to learn from observations and to generalize by abstraction. ANN has the ability to model highly nonlinear relationships with multiple inputs and outputs.

The objective of the training procedure is to identify the functions I_D Q_D , I_G Q_G from the set of DC, S-Parameter, and large-signal data measured at different temperatures. In the case of the drain current, dynamical variables (for instance, trap states voltages and junction temperatures) are computed from simple functions of the measured NVNA waveforms. The drain current constitutive relation for the model, as a function of V_{gs}, V_{ds} junction temperature, T_j, and trap states (f_1 , f_2) and the new dynamical variables, is represented by an artificial neural network (ANN), trained using the waveforms as well as the DC and S-parameters, de-embedded from the extrinsic parasitic network, as targets.

Start ANN Training

To start the ANN training setting, do the following:

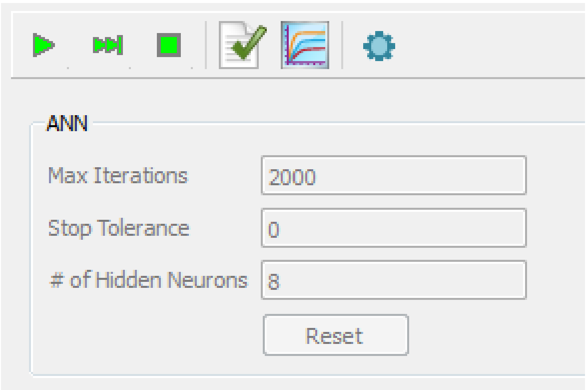
1. Select the **ANN model generation** from the Navigation pane, as depicted below. The overview pane by default lists all the intrinsic data to be generated when the **Generate Training Data** button,  , is pressed. When the function is executed, the program samples the data and prepares it for the training procedure. Check boxes are included to allow only subsets of the data to be generated if necessary (to save time).



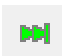
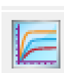
Once the training data is generated, the individual functional outputs can be trained.

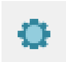
2. Select **Qg, Qd, Ig, Id** from the DynaFET Extraction Package window. All **Qg, Qd, Ig, Id**panes have similar features.

   Each has a control bar and a display area, showing current settings.



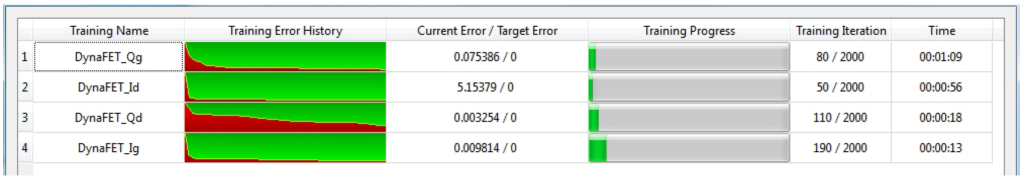
The control functions are summarized in the following table.

Function	Description
	Starts the ANN training
	Resumes the paused training session
	Allows to pause the training session
	Perform the test results using Sweep settings
	Displays the Plot results

	Allows to modify settings

Check ANN Training Status

You can start the training by pressing the Start button, if settings do not require a modification. When the training starts, the progress window is display (depicted below), displaying what is being trained, a graphical view of the training error (s), numerical value and display, a graphical representation of the progress, and the time spent.



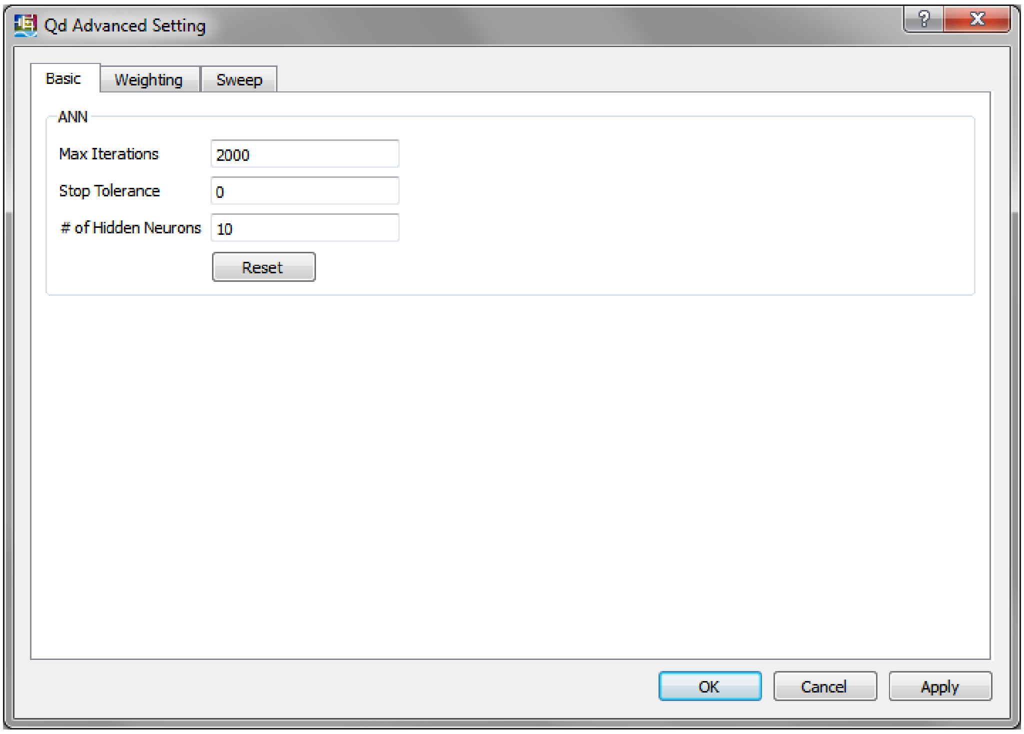
	Training Name	Training Error History	Current Error / Target Error	Training Progress	Training Iteration	Time
1	DynaFET_Qg		0.075386 / 0		80 / 2000	00:01:09
2	DynaFET_Id		5.15379 / 0		50 / 2000	00:00:56
3	DynaFET_Qd		0.003254 / 0		110 / 2000	00:00:18
4	DynaFET_Ig		0.009814 / 0		190 / 2000	00:00:13

Multiple training sessions can run simultaneously by simply navigating to other pages and starting the training sequence.

Individual sessions can be paused on their page, or all of the training can be stopped by pressing  from the main DynaFET control bar. By default, the target error is set to 0.0 and the number of iterations is set to 2000, so the training always run for the complete number of iterations. The status window also displays textual information.

After generating the data, the **Test ANN Training** displays the modeled data and the measured in a plot. The data can then be inspected for errors.

The **Sweep ANN training** evaluates the model at data points other than those used to create the data (for instance, smoothness). If there are problems, the model can be retrained after adjusting training parameters. It is important to check the model with swept inputs because if the model is over trained, the result could be ringing in the model with large mismatch at the bias points that do not exist in the measured data. These sweep parameters are available by pressing the Settings button. The first tab page contains the basic settings that are displayed on the main workspace window.

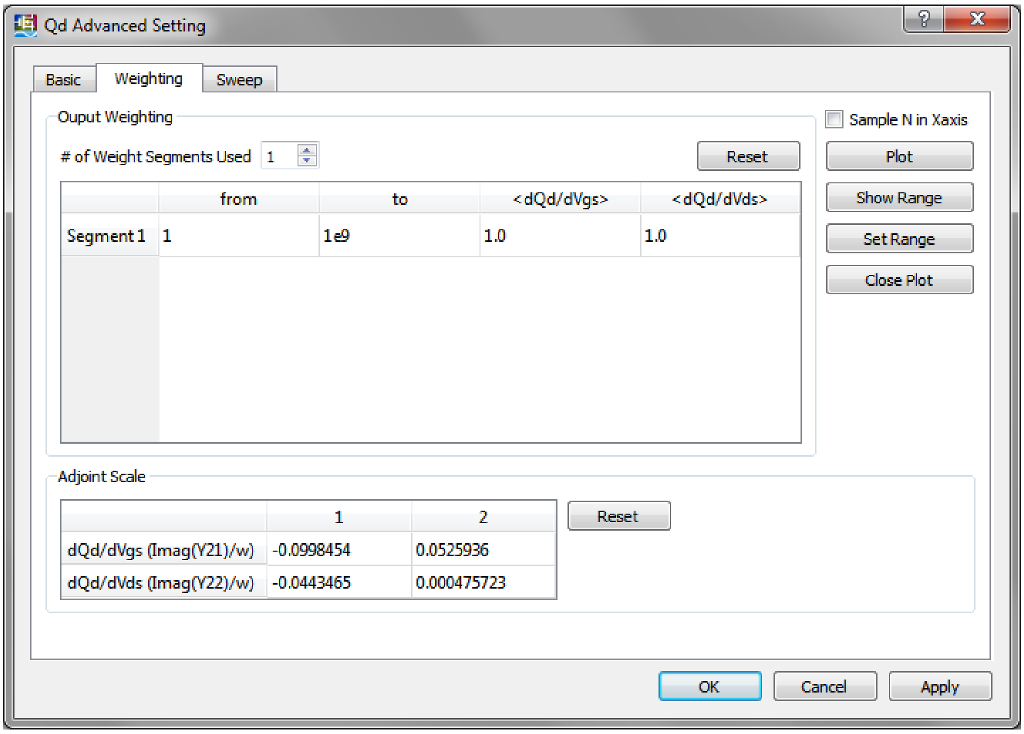


Basic properties in the Settings dialog box.

The properties are described in the table below.

Property	**Description**
Max Iterations	Indicates the maximum number of iterations the training can execute to reach the Stop Tolerance.
Stop Tolerance	Indicates that the training is complete, when the error term is defined as: $$E = \sum_{k=1}^{\#\ of\ waveforms} (W_1 \cdot E_{k,1} + W_2 \cdot E_{k,2})$$
# of Hidden Neurons	Indicates the number of neurons for each of the two levels in the neural net. Increasing this value could improve fit to complex output values at the expense of more training time.

The second tab page allows you to apply different weighting levels to the data set. This can be useful to fit the model better in regions that are important to the model's application.

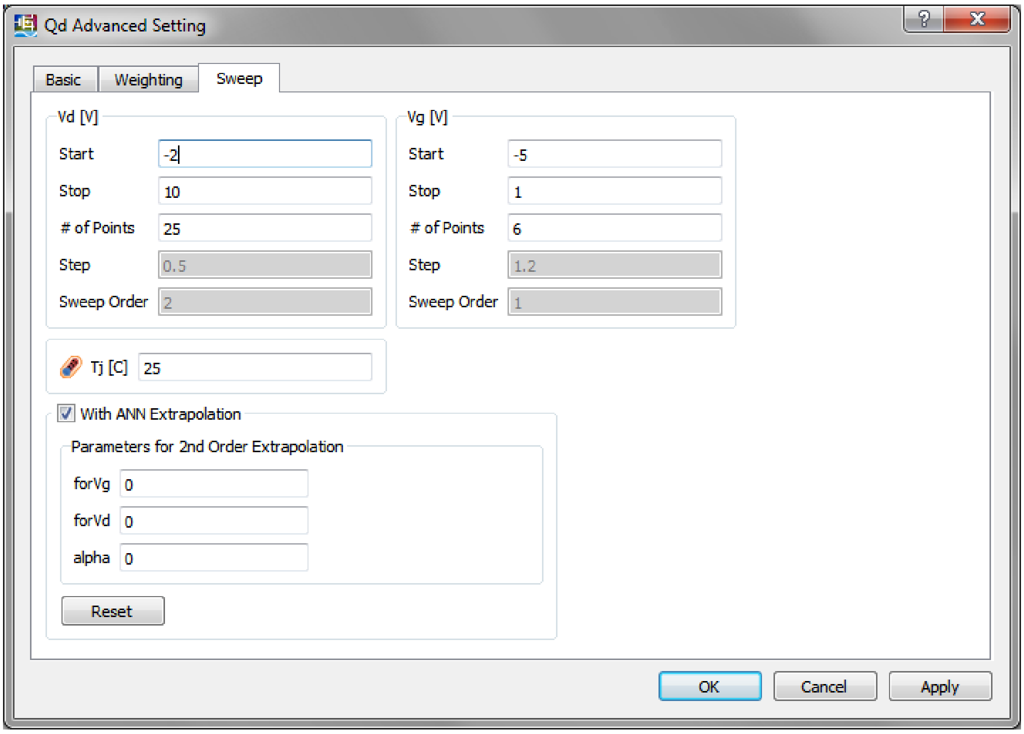


Weighting function advanced settings allow you to give emphasis to some regions of data.

To determine the segment, press the plot button that displays the appropriate data in a window. The data is numbered sequentially. The **# of Weight Segments Used** can be incremented to add additional segments, which can then have different weights applied. You can highlight a segment and press **Show Range** to display it as a blue box on the plot. Using the mouse to select and drag a rectangle displays a white box on the plot to define a new range. Pressing **Set Range** sets the selected segment to that range. The weighting function for each term can then be manually set in the appropriate field. The **Sample N in Xaxis** restricts the amount of data plotted for large datasets.

The range in the Adjoint values can be limited in range by setting the values in the array at the bottom of the dialog box. The pertinent values for the particular function are displayed and you can double-click on each field and enter a new value. These values limit the data used in training to those data points whose adjoint value is within the Adjoint Scale range.

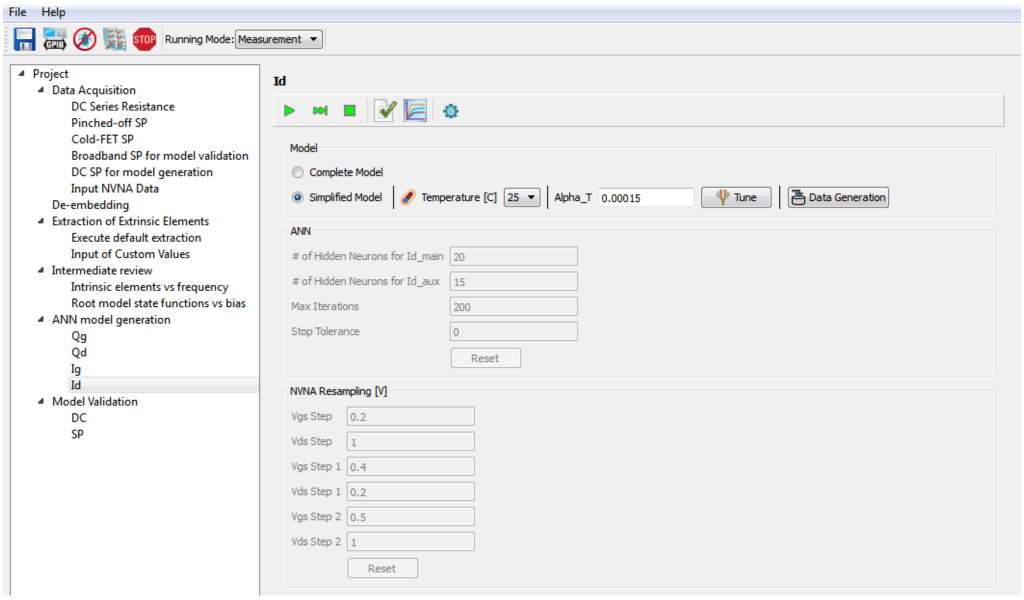
The third tab page is used to set the properties for the Sweep ANN Training function. The Vd and Vg sweep parameters set the values of biases to use, the temperature Tj sets the temperature for the simulation and the Option **With ANN Extrapolation** allows the model to be extrapolated beyond the training range. The **Parameters for 2nd Order Extrapolation** sets the weights (forVg, forVd) and exponential factor (alpha) in the extrapolation equation.



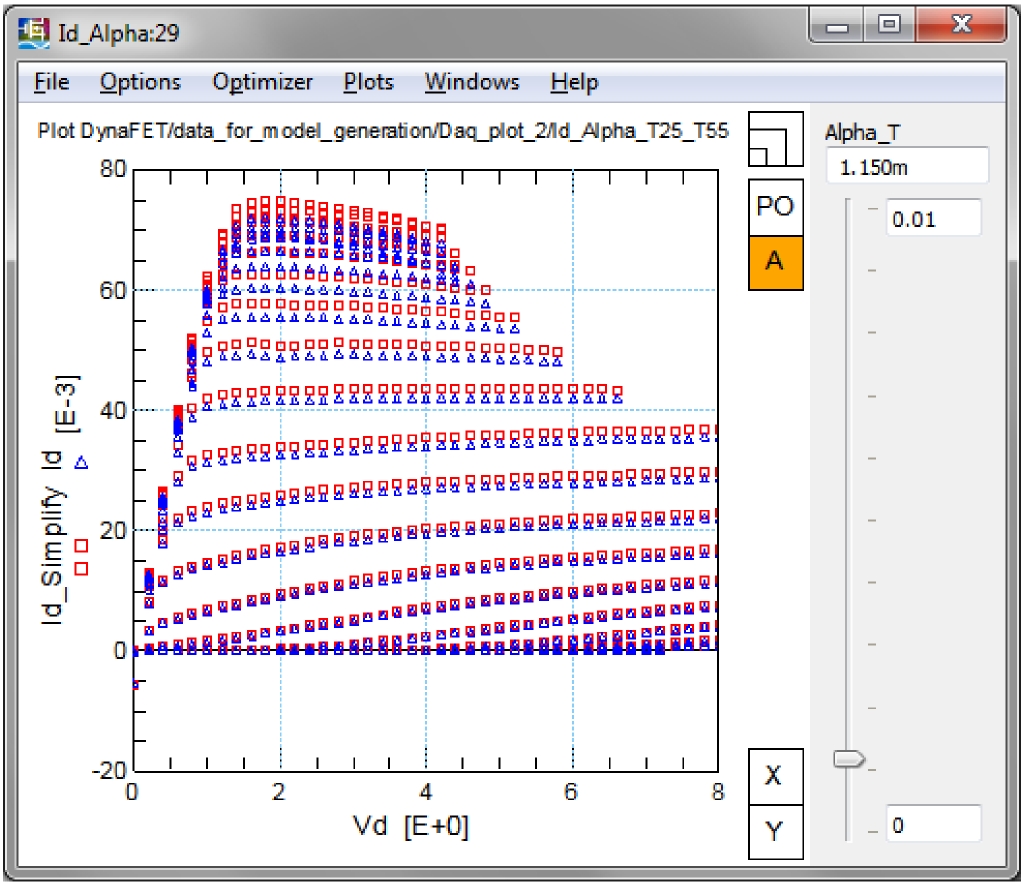
Sweep settings used by the Sweep ANN Training function. These settings are not used during the model training but only used to illustrate how well the model fits at bias points other than those used to train it.

ANN Training I D

The drain current model supports additional features. You can chose between a Complete Model (the default) or a Simplified Model.



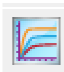
The simplified model only fits over 1 temperature, reducing the training time and non-linear data measurement time (the DC and S-Parameters must still be measured at more than one temperature, however). The simplified model's ANN assumes a fixed dependence on temperature. You must supply the coefficient **Alpha_T**. Pressing the Tune button displays the effect of the alpha coefficient. You can manually set **Alpha_T** based on the plot fit. Once the value is selected, press Data Generation to create the data to be trained on, then press **Start ANN Training**.



Tuning Alpha_T by adjusting the slider to fit the data at 2 temperatures in the simplified model.

The Settings dialog for the drain current includes additional properties. The number of Hidden Neurons can be set for both the main drain current (Id_main) or for the drain current due to influence of the auxiliary variables (Id_aux). The Basic setting allows choosing the re-sampling values for the NVNA non-linear waveform data. The Output Weighting can be adjusted for both the DC component of the non-linear waveforms and the DC values from the DC SP for model generation data. Because of the large number of non-linear waveforms, their weighting is much less than for Id_DC, by default. As the Sweep function can only plot 2-dimensional sweeps, the Setting include values to fix the trap voltages.

The extrapolation variables include a flag, **Force0**, which is used to tune the drain current at low V_{DS}. Setting a large value (i.e., 10) cause the parameter to have little effect. When the value is reduced below 1 (e.g., 0.5) the trained data is strongly coerced toward 0. Very small values (close to 0) should be avoided, as this can cause non-physical behavior.

After training, T**est ANN training** () displays how well the model fits against the measured data on which it was trained while **Sweep ANN training** () interpolates between data points and extrapolates beyond. These settings just define the bias region, to create the information and the sweep is completely unrelated to training.

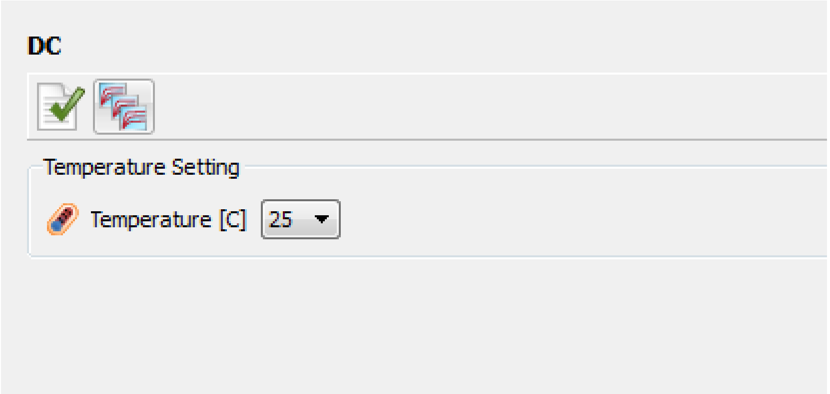
Model Validation

After generating the ANN model, you can run DC or SP model validation.

DC Model Validation

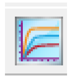
For DC model validation,:

1. Select **DC** from the Model Validation page in the navigation pane and select the temperature to be used.



2. Click the Validation icon ().

   The comparison between the simulated data and the previously measured data is displayed for validation.

3. If the data is generated, the plot windows can be re-opened using the Plot icon ().

SP Model Validation

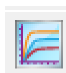
For SP model validation:

1. Select **SP** from Model Validation page in the navigation pane. The SP settings are displayed in the workspace area.

   NOTE Grayed out fields indicate that they cannot be edited.

2. Click the Validation icon ().

   The comparison between the simulated data (S-parameters versus bias and frequency) and the previously measured data is displayed for validation.

3. If the data is generated, the plot windows can be re-opened using the Plot icon ().

ADS Model Distribution

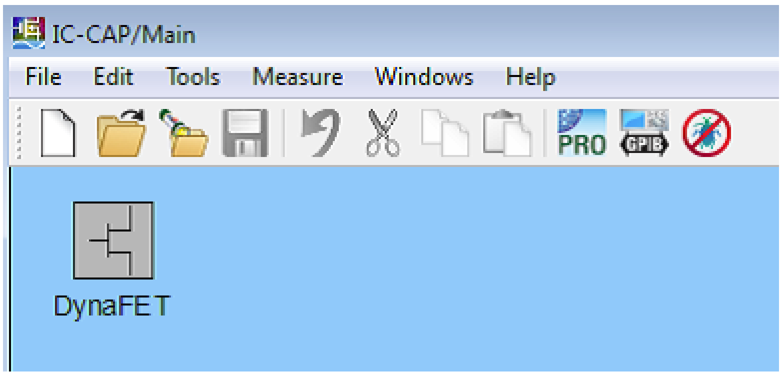
The DynaFET model files can be found in the project's validation directory. These files can then be referenced by the ADS modelcard.

Extracting a DynaFET Model

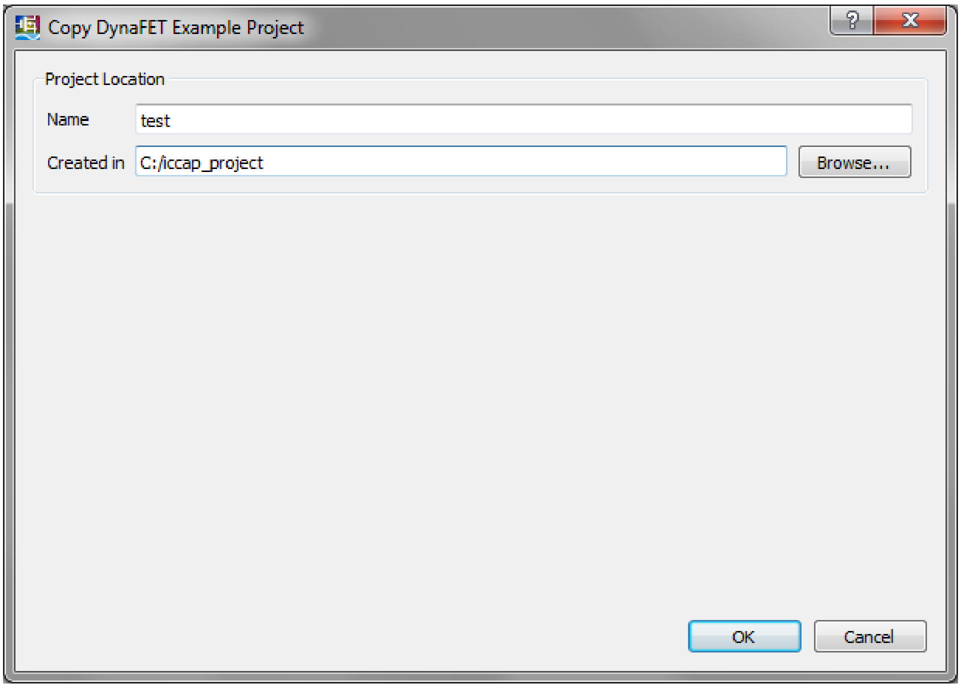
This section describes the step-by-step instructions and test procedure to measure, extract, and generate a DynaFET model using the DynaFET Modeling Package. DynaFET model extraction procedure is classified into 16 steps:

Step 1 – Launching the DynaFET Extraction Modeling Package

1. To start the program, from the **IC-CAP Main Window**, open the **Examples** folder and move to the model_files /mesfet/dynafet directory and select the DynaFET.mdl file.



2. Double-click the DynaFET icon in the **IC-CAP Main Window**, this displays a dialog box prompting for project location details.
3. Enter the project name and the directory, where the project is to be created
4. Click **OK**.

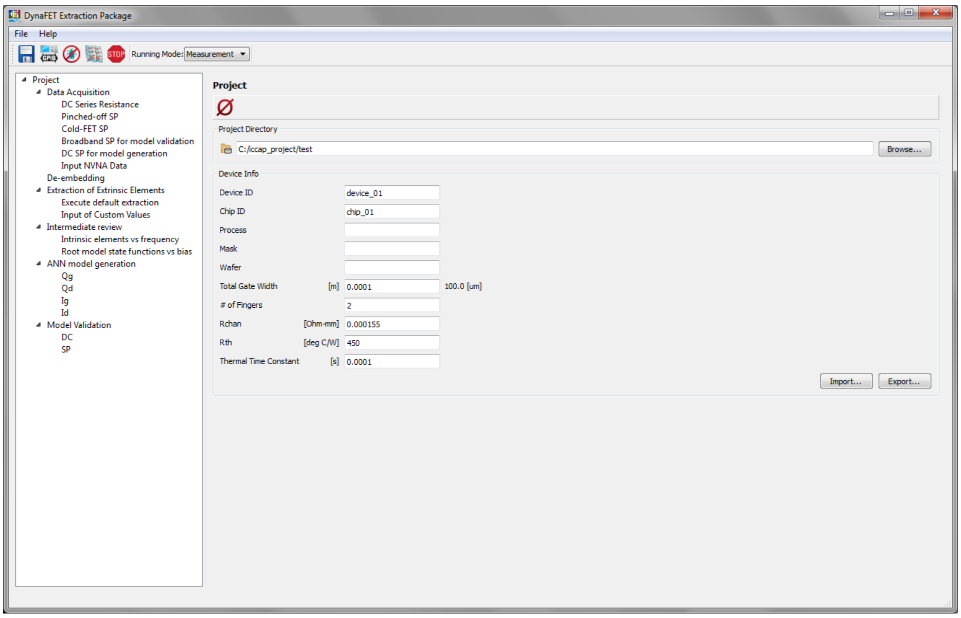


Step 2 – Entering the Device information

NOTE To load previously stored set of device information, click **Import...** and select the respective file.

1. Enter a string for the Device ID and the Chip ID. If required, values can be entered for the Process, Mask, and Wafer.
2. Enter the total gate width in meters and the # of fingers for the device.
3. Enter the channel resistance in Ohm-mm and the thermal resistance (in degree C / Watt) and thermal time constant (in seconds).

   NOTE The device information can be stored in a file, so it can be conveniently loaded in other projects. To load the informaiton, click **Export...** and provide the file name.



Step 3 – Configuring the Hardware

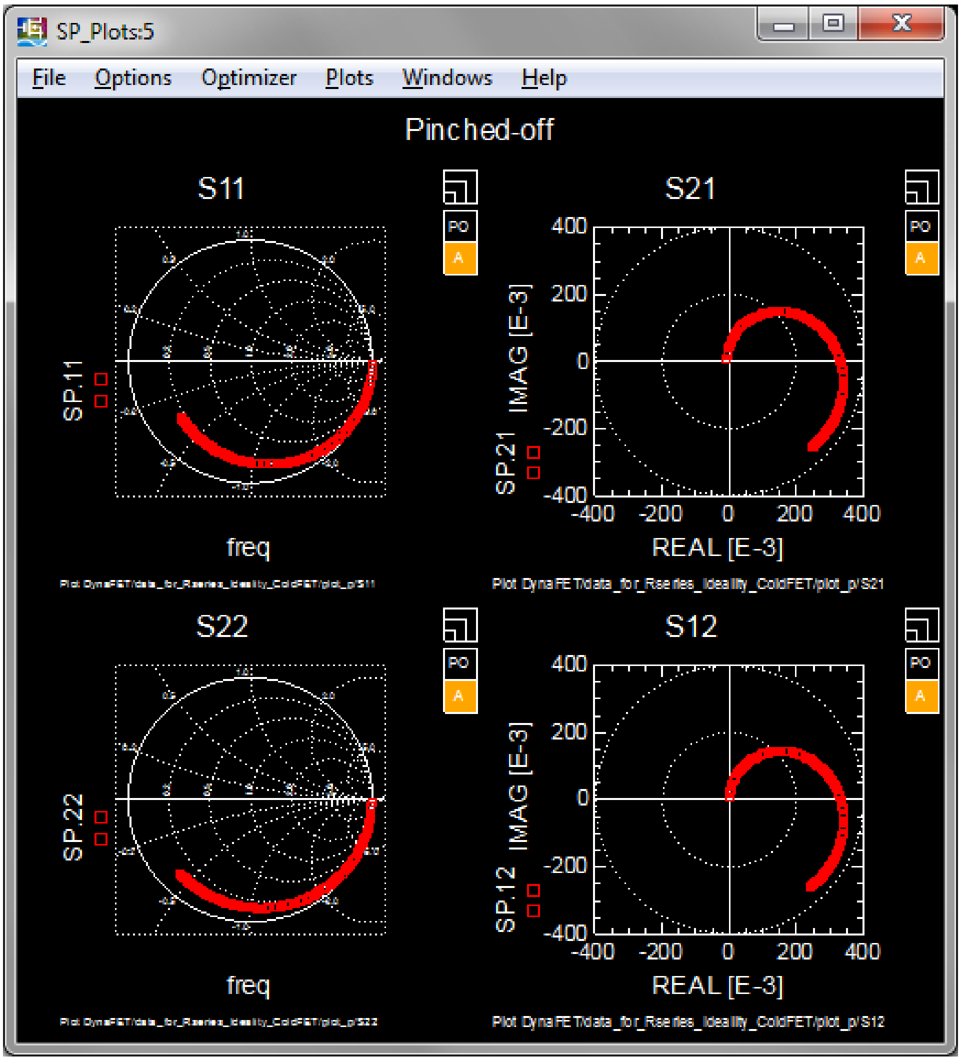
1. Setup the hardware with a PNA-X NVNA and a power supply. The PNA-X should be calibrated over the frequency range of interest with the power and attenuated settings, and the average values.
2. Save the instrument state after making all the settings.
3. Click on Data Acquisition in the Navigation pane and the Instrument Setting tab. Open the Hardware Window () and add an interface and rebuild the instrument list, if necessary. Select the power supply and press **Configure...** and assign the SMU1 and SMU2 names.
4. Close the Hardware window and download the instrument information by pressing  .
5. In the middle column, select the DC Source/Monitor and the NWA from the pull down lists.
6. Select each instrument and set any property values in the table. It is recommended to use an integration time of at least Medium. In the NWA settings, set the instrument state and other values to match the state settings or set the property “Use PNA Calibration Settings” to “Yes”.
7. Click the **Measurement Setting** tab and modify the required fields.

Step 4 – Measuring the DC Series Resistance

1. Select the **DC Series Resistance** page from the **Data Acquisition** parent page in the **Navigation** pane.
2. Lower both probes onto shorting structures.
3. Select **Measure** and press the **Measure** icon  .
4. Press the **Plot** icon and inspect the plots. The I-V curves should be straight lines with minimal noise.

Step 5 – Measuring the Pinched-off FET characteristics

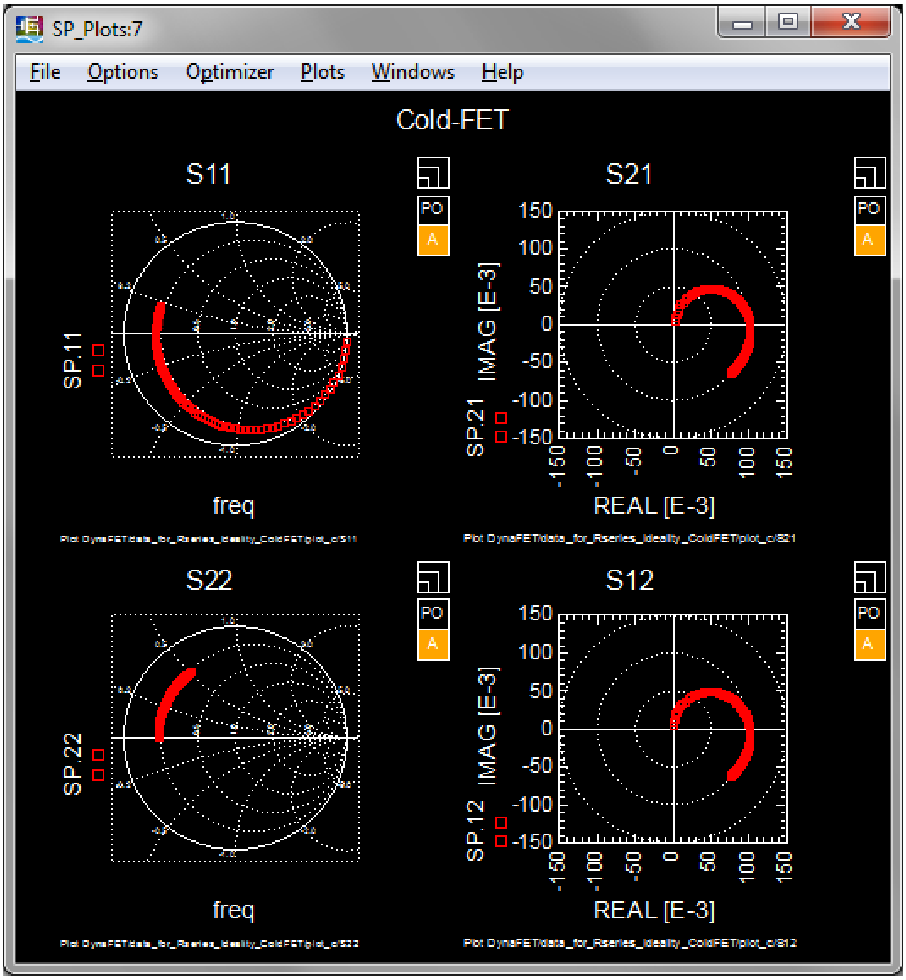
1. Select the **Pinched-off SP** page from the **Data Acquisition** parent page in the **Navigation** pane.
2. If required, modify the Frequency sweep or bias settings.
3. Select **Measure** and press the **Measure** icon  .
4. Press the **Plot** icon and inspect the plots. The S-Parameters should be smooth and appear similar to those shown below.



5. Press the **Save** icon  to save the data.

Step 6 – Measuring the Cold-FET characteristics

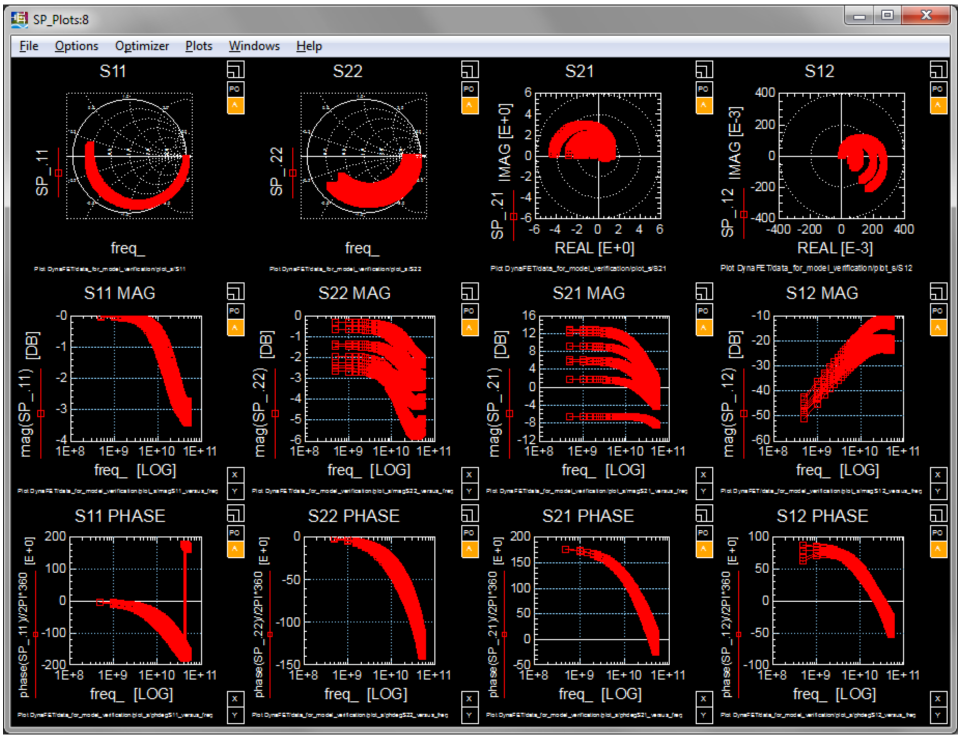
1. Select the **Cold-FET SP** page from the **Data Acquisition** parent page in the **Navigation** pane.
2. If required, modify the Frequency sweep or bias settings.
3. Select **Measure** and press the **Measure** icon  .
4. Press the **Plot** icon and inspect the plots. The S-Parameters should be smooth and appear similar to those shown below.



5. Press the **Save** icon  to save the data.

Step 7 – Measuring the Broadband S-Parameters for model validation

1. Select the **Broadband SP for model generation** page from the **Data Acquisition** parent page in the **Navigation** pane.
2. If required, modify the Frequency sweep or bias settings. Choose bias points that is different from those used to extract the model.
3. Select **Measure** and press the **Measure** icon  .
4. Press the **Plot** icon and inspect the plots. The S-Parameters should be smooth and appear similar to those shown below.



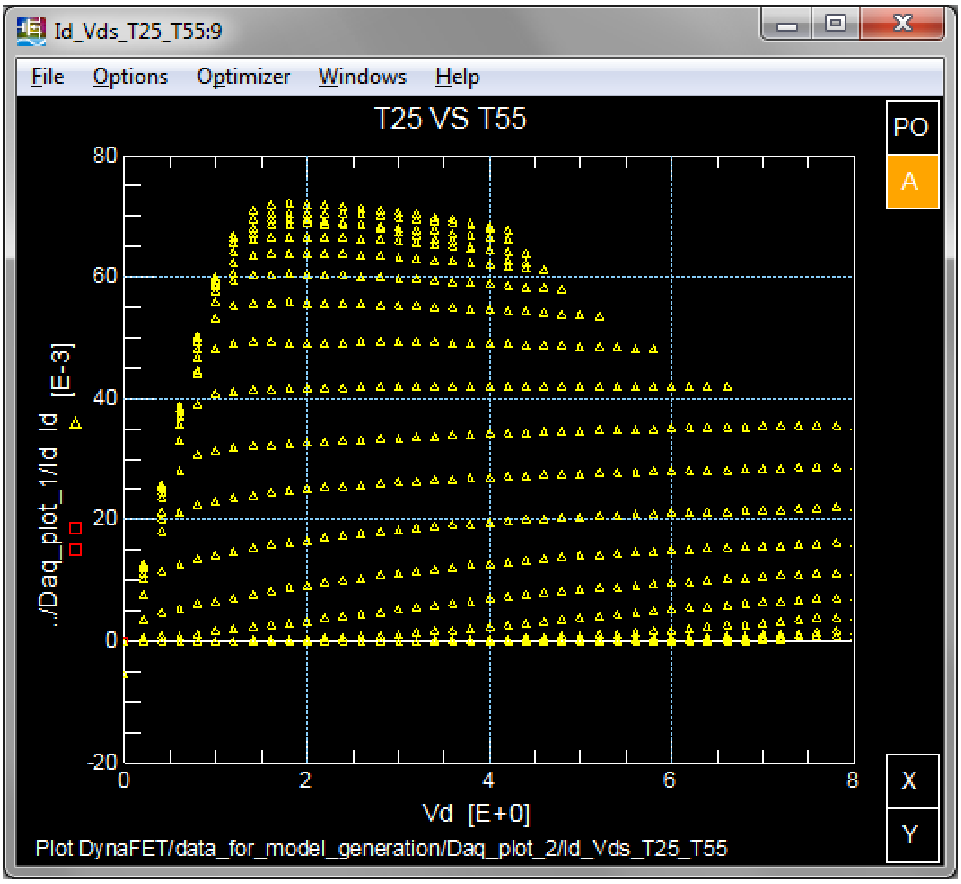
5. Press the **Save** icon  to save the data.

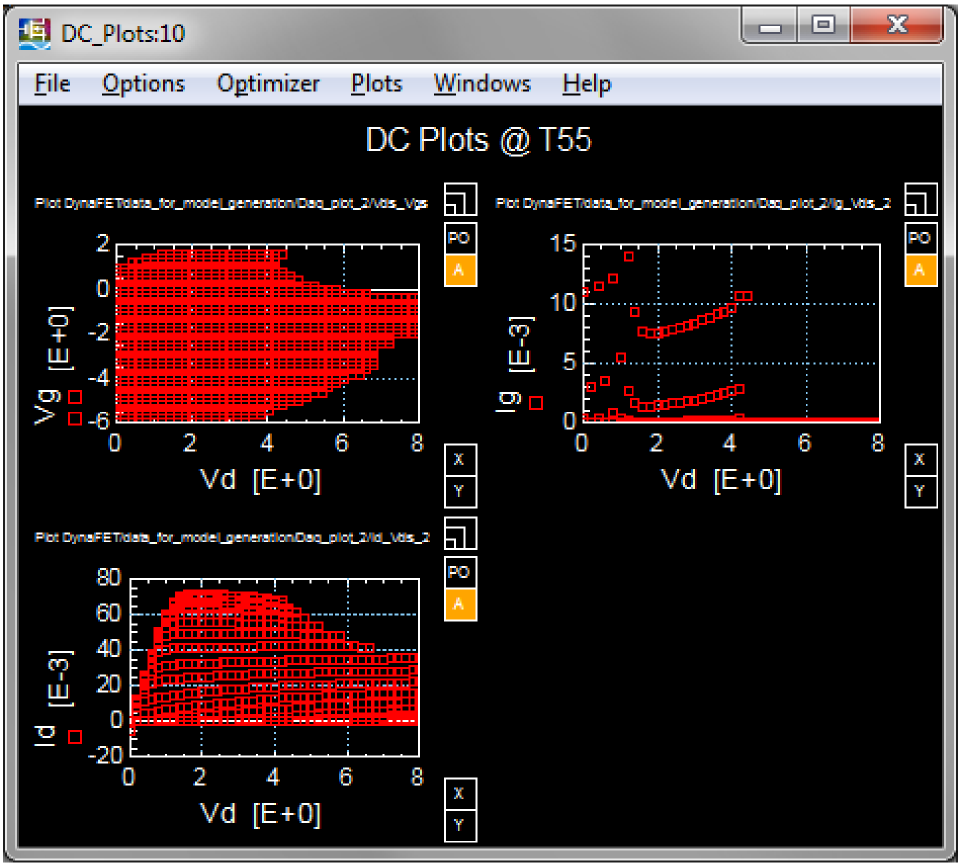
Step 8 – Measuring the DC and S-Parameters for model generation

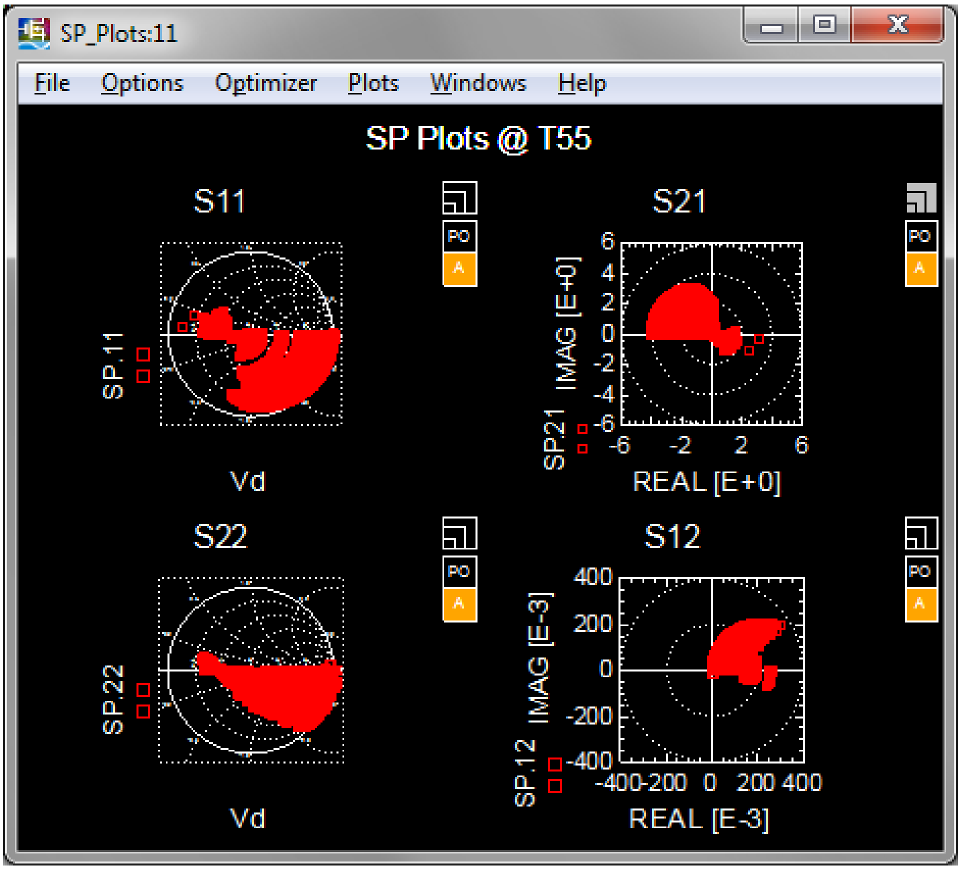
1. Select the **DC SP for model generation** page from the **Data Acquisition** parent page in the **Navigation** pane.
2. Set the power, breakdown, and compliance values associated with the device.
3. If required, press the **Advanced Setting** button and set additional parameters related to the adaptive step and bias soak feature.
4. Set the appropriate temperature value for the actual device temperature.

   NOTE The temperature settings are managed manually.

5. Set the number of regions and the bias sweep and options for each region.
6. If the safe operating region is not known, select the **DC Only** option and slowly approach the region boundaries. It may be necessary to use a sacrificial device.
7. Select **Measure** and press the **Measure** icon  .
8. Press the **Plot** icon and inspect the plots. The DC-IV curves should cover the device's operating region. The S-Parameters should be smooth and appear similar to those shown below.


Id_Vds_T25_T55:9
File Options Optimizer Windows Help
T25 VS T55
PO
A
../Daq_plot_1/Id Id [E-3]
Vd [E+0]
X
Y
Plot DynaFET/data_for_model_generation/Daq_plot_2/Id_Vds_T25_T55


DC_Plots:10
File Options Optimizer Plots Windows Help
DC Plots @ T55
Vg [E+0]
Vd [E+0]
Ig [E-3]
Vd [E+0]
Id [E-3]
Vd [E+0]

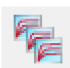


9. Press the **Save** icon  to save the data.

Step 9 – Bringing the NVNA data into the DynaFET Extraction Package

1. Select the **Input NVNA Data** page from the **Data Acquisition** parent page in the **Navigation** pane.
2. For each data set, enter the corresponding temperature the data was acquired at.
3. Press **Get File...** and select the NVNA waveform data file.
4. Enter the DC series resistance of the NVNA setup.

Step 10 – De-embedding the extrinsic test structure

1. Select the **De-embedding** page from the **Navigation** pane.
2. Toggle the check boxes for either the Add Open/Short or Add Manifold settings.
3. Double-click on each red box on the schematic and select the corresponding S-Parameter (s2p) file.
4. Check that ll of the check boxes in the **Data to be de-embedded** section is checked.
5. Click  to de-embed the data files.
6. Click  to plot the de-embedded data. Inspect the data for unusual behavior.

Step 11 – Extracting the extrinsic element network from the data

1. Select **Execute default extraction** from the **Extraction of Extrinsic Elements** page in the **Navigation** pane.
2. Select the **Extract Capacitance** tab page.
3. Click the plot button to display the capacitance values versus frequency.
4. Using the plots to find a suitable region where the capacitance is independent of frequency, select an extraction frequency value from the pull down list.
5. Click to extract the values.
6. Select the **Extract Inductance and Resistance** tab.
7. Select a frequency range to extract the resistances and capacitors.
8. Click to extract the values.
9. Select the **Generate Intrinsic Data** tab page.
10. Ensure that both **DC SP for model generation** and **NVNA** checkboxes are selected.
11. Click the **Generate Intrinsic Data** button .

Step 12 – Extracting Rgs0/Alpha

1. Select the **Intrinsic elements vs frequency** page under the **Intermediate Review parent** page in the **Navigation** pane.
2. Select Vgs and Vds values and add them to the **Selected Bias** column.
3. Click the plot button to display the intrinsic values.
4. Using the plots to determine a suitable frequency where the intrinsic values are frequency independent, set the **Extraction Frequency**.
5. Set an appropriate bias for Rgs0 and Alpha (typically an operating point where the device is used).
6. Click to extract the values.
7. Click to **Generate Intrinsic Data**.

Step 13 – Reviewing the data using a Root model

1. Select the **Root model state functions vs bias** page under the **Intermediate Review parent** page in the **Navigation** pane.
2. Specify a bias to generate the Root model.

3. Click to extract the model. The state functions are displayed.
4. Click to generate the Root model data to compare to the measured data.
5. Inspect the plots for unusual results that might indicate device issues (e.g., degradation).

Step 14 – Training the Artificial Neural Network

1. Select the **ANN model generation** page from the **Navigation** pane.
2. Check that all of the functions are checked for data generation.
3. Click to **Generate Training Data**.
4. For each Qg, Qd, Ig, Id, go to the appropriate page and click to start the training.

   NOTE Multiple ANN trainings can run in parallel.

5. Click to plot the trained ANN versus the measured data it was trained on.
6. Press to sweep the ANN training over new bias ranges based on the Settings->Sweep tab page.

Step 15 – Validating the model

1. Select **DC** or **SP** pages from the **Model Validation parent** page in the **Navigation** pane.
2. If rquired, modify the frequency and bias ranges.
3. Click to generate the data and display the plots.

Step 16 – Accessing the model in Advanced Design System (ADS)

1. Copy the DynaFET_model.enc file from the project's validation directory to the ADS workspace data directory.
2. From the ADS schematic, add the DynaFET instance and model card from the Devices-JFET palette.
3. On the model card, set the parameter DynaModelFile to point to the file.