Agilent Technologies

Advanced Design System 2011.01

Feburary 2011
DesignGuide Utilities

Acknowledgments

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7-zip - 7-Zip Copyright: Copyright (C) 1999-2009 Igor Pavlov. Licenses for files are: 7z.dll: GNU LGPL + unRAR restriction, All other files: GNU LGPL. 7-zip License: 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. unRAR copyright: The decompression engine for RAR archives was developed using source code of unRAR program.All copyrights to original unRAR code are owned by Alexander Roshal. unRAR License: The unRAR sources cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver. 7-zip Availability: http://www.7-zip.org/

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. AMD Availability: http://www.cise.ufl.edu/research/sparse/amd

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License

along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: http://www.cise.ufl.edu/research/sparse/umfpack UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at http://www.cise.ufl.edu/research/sparse . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at http://www.cise.ufl.edu/research/sparse . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. http://www.mathworks.com . COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at http://www.netlib.org). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

Qt Version 4.6.3 - Qt Notice: The Qt code was modified. Used by permission. Qt copyright: Qt Version 4.6.3, Copyright (c) 2010 by Nokia Corporation. All Rights Reserved. Qt License: Your use or distribution of Qt or any modified version of Qt implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the
terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User
documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. 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 Agilent Inc. at brian_buchanan@agilent.com for more information.

The HiSIM_HV source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code, is owned by Hiroshima University and/or STARC.

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at http://systemc.org/ . This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

Restricted Rights Legend U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

Load Pull Measurement Data Import Utility

The *Load Pull Measurement Data Import* utility is used to import *Maury* or *Focus* measured load pull data in ADS for interpolation and generation of load and source pull contours. It is assumed that the data is generated for a constant bias, fixed frequency, and constant input power.

The Load Pull Measurement Data Import utility can be selected and installed during installation process of ADS. After installation, the Load Pull utility can be accessed from the Schematic window.

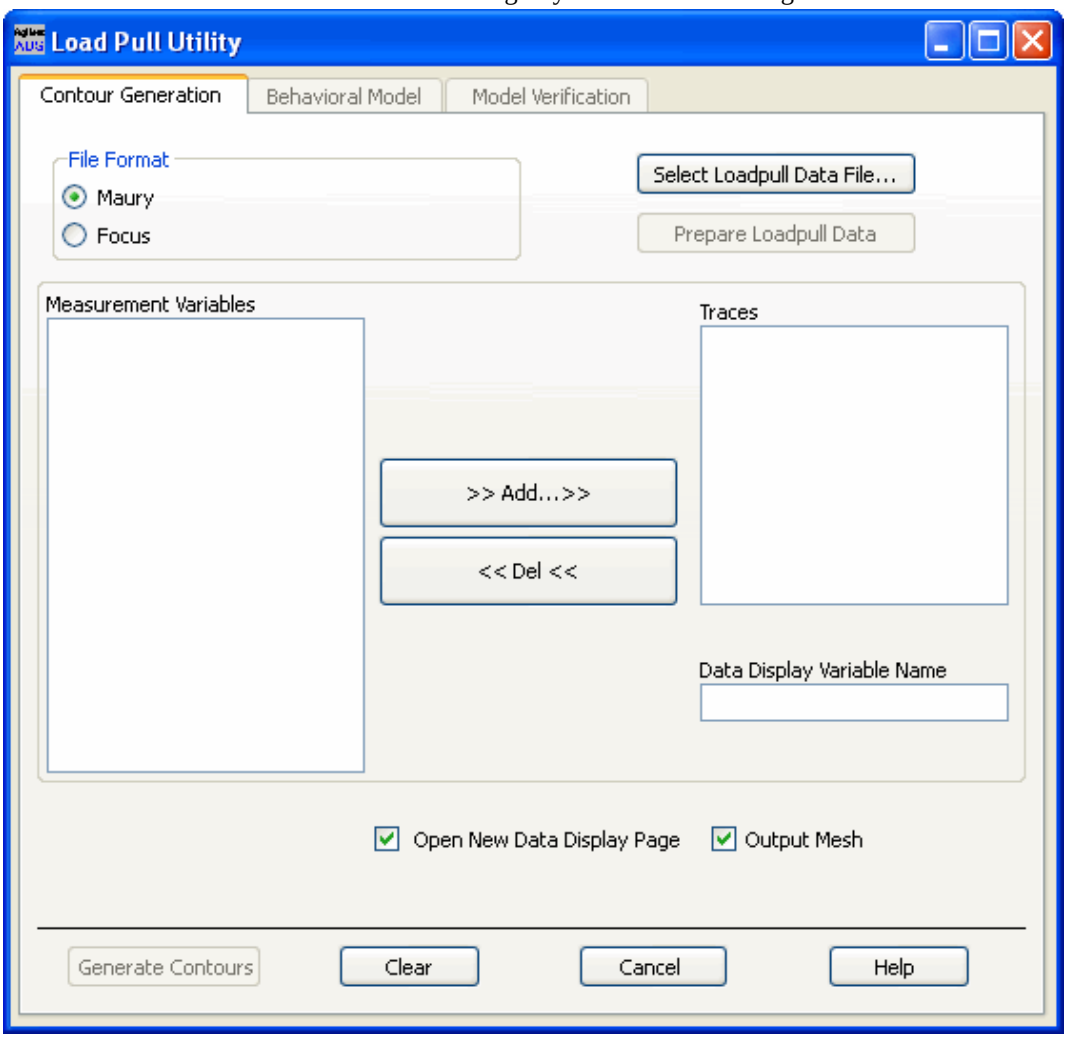
The process includes:

- Importing Loadpull Measurement Data
- Preparing Data for Interpolation
- Interpolating the Data and Displaying Results

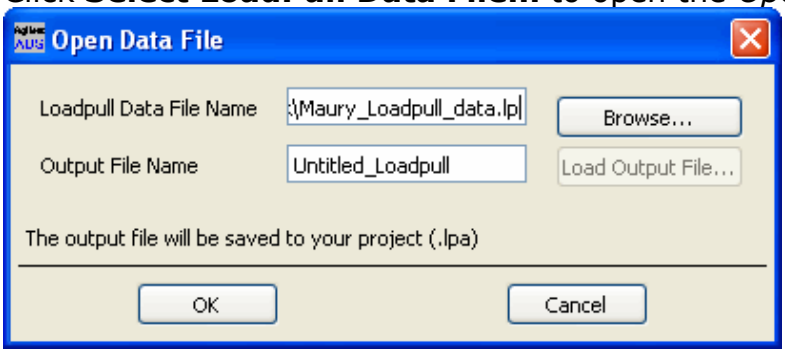
Importing Load Pull Measurement Data

Before you use the Load Pull Measurement Data Import utility, you must measure the loadpull data available on your system. If you do not have measured loadpull data, you can access and copy sample measured data file for *Maury* and *Focus* microwave system to your workspace directory by selecting **DesignGuides > Load Pull > Sample Load Pull Data File**.
To import load pull measurement data:

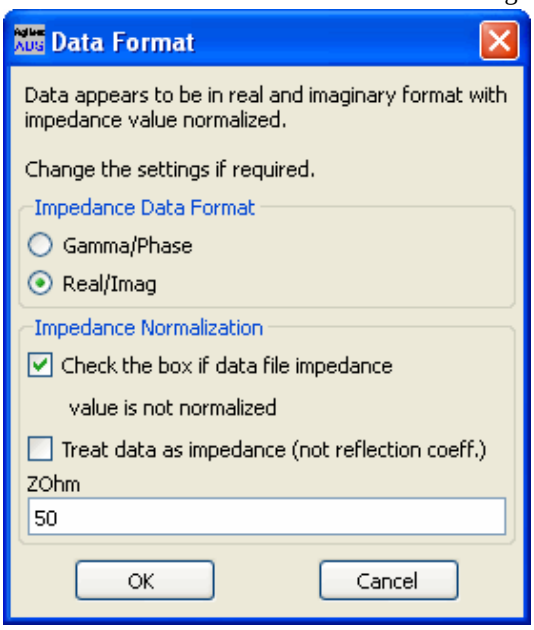
1. Choose **DesignGuides** > **Load Pull** > **Load Pull Measurement Data Import Utility** from the Schematic window.
2. Click **OK** to open the *Load Pull Utility* dialog box.



3. Choose *Maury* or *Focus* orotion from the **File Format** frame.
4. Click **Select LoadPull Data File...** to open the *Open Data File* dialog box.



5. Click **Browse** to select a specific data file from the workspace directory.
   You can click **Load Output File** option to load a previously-prepared load pull data file. Use this option if you have successfully prepared load pull data earlier and you want to load it now. All the prepared load pull file have an extension *.lp* . When selecting the data file, all the Measurement Variable for contour plots loads directly. You can add a Measurement Variable to Trace and Generate Contours.
6. Enter the file name in *Output File Name* and click **OK**.
   During selection of load pull-measured data file, the format of the load pull impedance data is determined. If the impedance value is normalized to its characteristic impedance, the *Data Format* dialog box displayed during this process reflects these settings.



In some cases, it is difficult to determine the data format correctly due to non-availability of measured data points across the entire Smith Chart region. In this case, you must make the changes manually.

If the impedance values are not normalized and appear in the load pull file as, for instance, 52+j×32, check the first checkbox under **Impedance Normalization**, and set the value of ZOhm to the applicable normalized impedance (for example, 50 ohms).

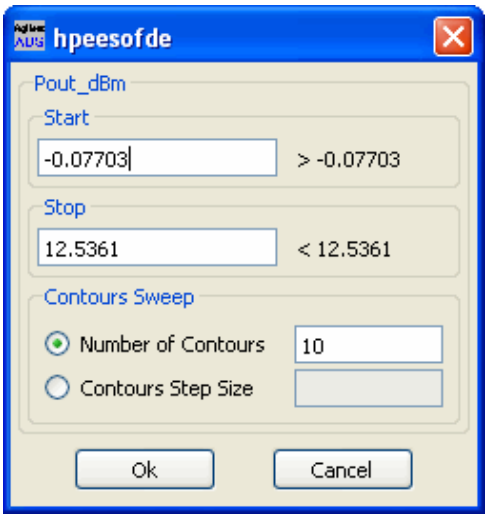
If the data needs to be treated as impedance rather than reflection coefficient, check the **Treat data as impedance (not reflection coeff.)** checkbox. (This box should not be checked if the data is in reflection coefficient (S11) format.)

Preparing Data for Interpolation

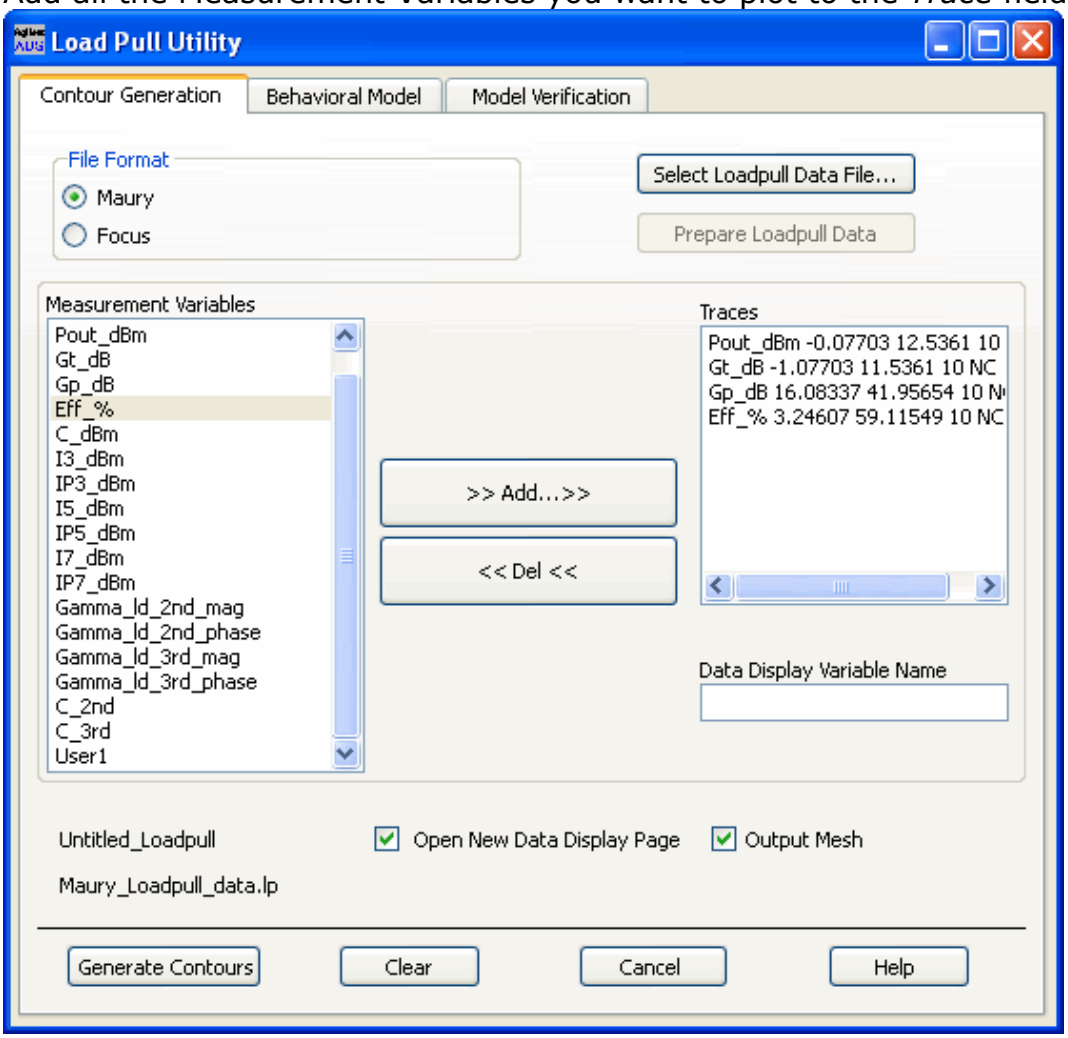
In this section, the tool reads the measurement points and prepares the data for interpolation. You also have an opportunity to select the measurement variables you want to plot.

To prepare data for interpolation:

1. Clcik **Prepare Loadpull Data** from the *Load Pull Utility* dialog box.
   The process takes all the measurement points and generates a triangular mesh to perform linear interpolation on scattered data. After the data is prepared for interpolation, the measurement variables for which the contours can be generated appear in the *Measurement Variable* field. The utility does not restrict the number of variables you can plot simultaneously. You can generate load pull contours for multiple variables at the same time, provided you have valid measurement data to generate load pull contours.
2. Select the variable from the *Measurement Variable* field.
3. Click **<<Add...>>** to enter the minimum and maximum bounds for your contour plots and the number of contours you want to generate within defined bounds.



4. Click **OK** to add the variable in the *Trace* field.
   Add all the Measurement Variables you want to plot to the *Trace* field.



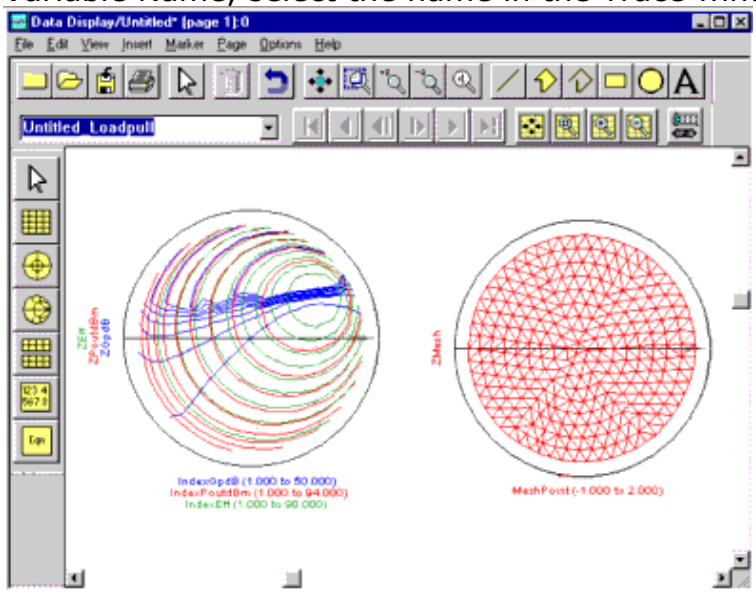
5. Select **Open New Data Display Page** so you can plot load pull contours on Smith Chart or on Polar Chart. (Note: to view contours on a rectangular grid, select Plot real(Variable_Name) vs. imag(Variable_Name)). For details on using the data display, refer to *Plots and Lists* (data) in the *Data Display* documentation.
6. Select **Output Mesh** to export the triangular mesh generated during Prepare Loadpull Data to a data display page. The output mesh can be plotted by plotting the variable Zmesh on the Smith Chart or Polar Chart. You can use the output mesh to view the measurement points and for debugging purposes.

Interpolating the Data and Displaying Results

The interpolation process generates load pull contours and displays the results.

To interpolate the data:

1. Click **Generate Contours** after preparing data for interpolation to start the interpolation process. This process interpolates the prepared data and generates load pull contours. At the end of the interpolation process, the Data Display window is opened.
2. Select the type of display from the list. During the translation process, the measurement variable name changes to remove all special character and a prefix "Z" is appended at the beginning. To view the new assigned name in the Data Display Variable Name, select the name in the Trace window.



Transistor Bias Utility

The *Transistor Bias Utility* documentation provides an introduction to the Transistor Bias Utility. The complexity of the Advanced Design System (ADS) is made easily accessible through the automated capability. A first-time or casual ADS user can begin using the capability of ADS quickly, while experienced ADS users can perform tasks faster than ever before. The Step-by-Step Example describes how a resistive bias network for a GaAs FET can be designed and verified in a few minutes.

The *Transistor Bias Utility* provides *SmartComponents* and automated-assistants for the design and simulation of common resistive and active transistor bias networks. The automated capabilities can determine the transistor DC parameters, design an appropriate network to achieve a given bias point, and simulate and display the achieved performance. All SmartComponents can be modified. You simply select a SmartComponent and, with little effort, redesign or verify their performance. Using SmartComponents provides details about using SmartComponents.

Step-by-Step Example

The step-by-step example takes you through the design and analysis for a resistive bias network for a GaAs FET. After completing this example, you should have a basic understanding of the Utility and be ready to begin using it. Follow these steps to begin:

- Setting Up the Design Environment
- Designing and Analyzing a Network

Note
You should already be familiar with the basic features of Advanced Design System. For help with ADS basic features, refer to the *Schematic Capture and Layout* (usrguide) documentation.

Setting Up the Design Environment

Before you can use the Transistor Bias Utility, you must set up the design environment by using these steps:

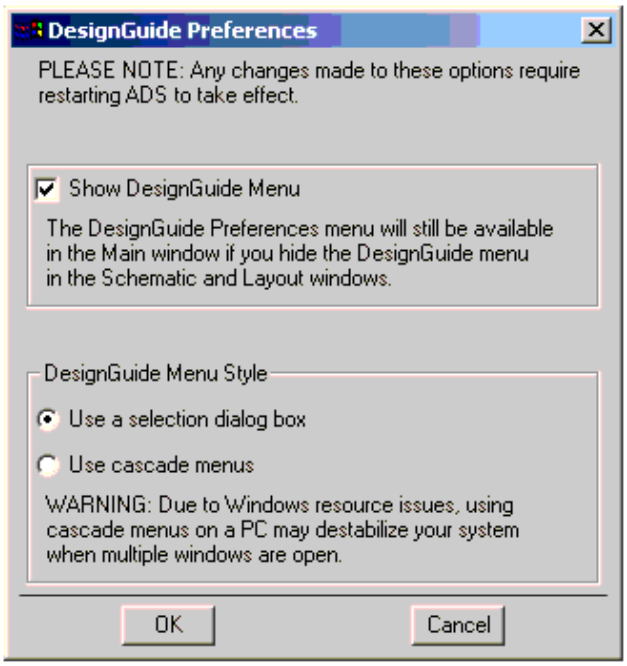
- Setting DesignGuide Preferences
- Opening a Workspace
- Opening a Schematic Window
- Opening the Transistor Bias Utility
- Displaying the SmartComponent Palette.

Note
Before beginning, you must have installed the DesignGuide with appropriate licensing codewords.

Setting DesignGuide Preferences

All DesignGuides can be accessed through either cascading menus or dialog boxes. You can configure your preferred method in the ADS Main window or from the Schematic window.
To configure access through menus or dialog boxes:

1. Click **DesignGuide** > **Preferences** from the ADS Main or Schematic window.
2. In the DesignGuide Menu Style group box, choose either **Use a selection dialog box** or **Use cascade menus**.



3. Click **OK**.
4. Close and restart the program for your preference changes to take effect.

Note
On PC systems, Windows resource issues might limit the use of cascading menus. When multiple windows are open, your system could become destabilized. Therefore, the dialog box menu style might be best for these situations.

The ADS Main window DesignGuide menu contains the following choices:
DesignGuide Developer Studio > **Start DesignGuide Studio** is only available on this menu if you have installed the DesignGuide Developer Studio to open the initial Developer Studio dialog box.
DesignGuide Developer Studio > **Developer Studio Documentation** is only available on this menu if you have installed the DesignGuide Developer Studio to open the DesignGuide Developer Studio documentation.

Note
Another way to access the DesignGuide Developer Studio documentation is by selecting *Help* > *Topics and Index* > *DesignGuides* > *DesignGuide Developer Studio* from any ADS program window.

Add DesignGuide opens a directory browser in which you can add a DesignGuide to your installation. This is primarily intended for use with DesignGuides that are custom-built through the Developer Studio.
List/Remove DesignGuide opens a list of your installed DesignGuides. Select any that you would like to uninstall and choose the *Remove* button.
Preferences opens a dialog box that enables you to:

- Disable the DesignGuide menu commands (all except Preferences) in the Main window by unchecking this box. In the Schematic and Layout windows, the complete DesignGuide menu and all of its commands are removed if this box is unchecked.
- Select your preferred interface method, either cascading menus or dialog boxes.

Opening a Workspace

The ADS design environment is set up within a workspace.
To create a new workspace:

1. Click **File > New > Workspace** from the *ADS Main* window, the New Workspace Wizard starts.
2. Follow the steps to enter the workspace location and assign a workspace name.

Opening a Schematic Window

A new schematic design is needed to contain the lowpass component for this example.

To open a Schematic window:

1. Click **Window** > **New Schematic** or click **New Schematic Window** on the toolbar from the *ADS Main* window. A new *Schematic* window appears.

Hint
Depending on how your ADS preferences are set, a Schematic window can appear automatically when you create or open a workspace.

2. Click **File** > **New Design** from the Schematic window, to create a design named *Example*.

Opening the Transistor Bias Utility

The Transistor Bias Utility is accessed from the DesignGuide menu.
To open the Transistor Bias Utility:

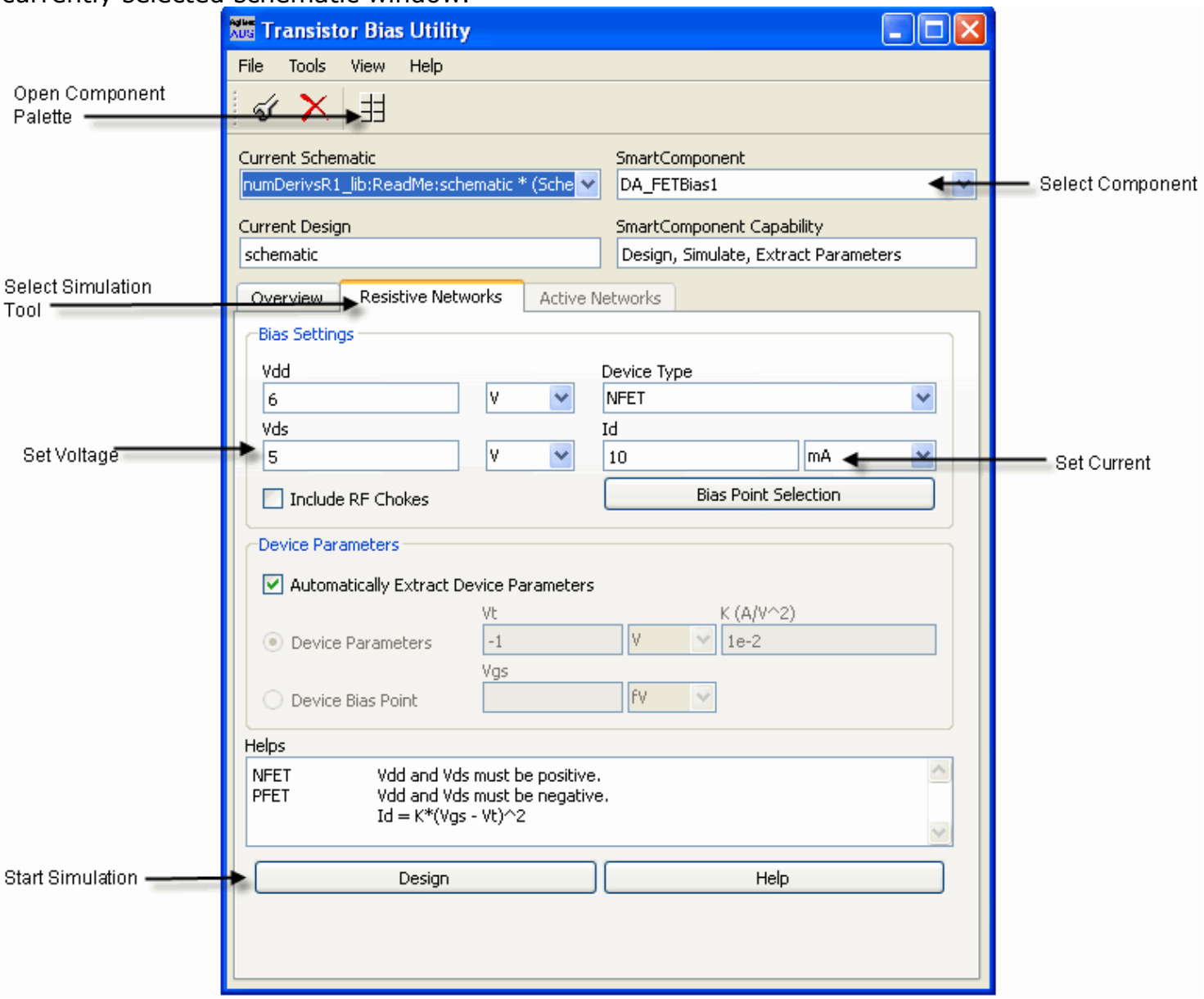
1. In the Schematic window, choose one of these paths from the DesignGuide menu:
   - **DesignGuide** > **Amplifier** > **Tools** > **Transistor Bias Utility**
   - **DesignGuide** > **Mixers** > **Tools** > **Transistor Bias Utility**
   - **DesignGuide** > **Oscillator** > **Tools** > **Transistor Bias**

   Hint
   Expand the list under Tools by clicking the _ sign.
2. Select **Transistor Bias Utility** and click **OK** to open the tool. The Control window appears.

Using the Control Window

All Utility features are available from the Control window. The Control window houses menus, a toolbar, and SmartComponent manipulation controls. The menus and toolbar buttons perform the basic functions of design, delete, and display the SmartComponent palette. Full features are available from each of the tab pages on the window. The window can be placed anywhere on the screen. Explore each tab page by clicking on the tab at the top of each page. Explore the window menus as well to familiarize yourself with the basic Utility capabilities.

The pull down lists at the top of the control window are designed to help you navigate multiple schematic windows and SmartComponents. You can use the Current Schematic drop-down list box to select any of the currently opened schematic windows. This field is updated any time Bias Control Window is selected from the DesignGuide menu. From the SmartComponent drop-down list box, you can select any of the SmartComponents on the currently selected schematic window.



To close the Control window:

- Click **File** > **Exit DesignGuide** from the Control window menubar. (You can also close the window by clicking the **x** at the top of the window.)

Continue the step-by-step example by Designing and Analyzing a Network.

Designing and Analyzing a Network

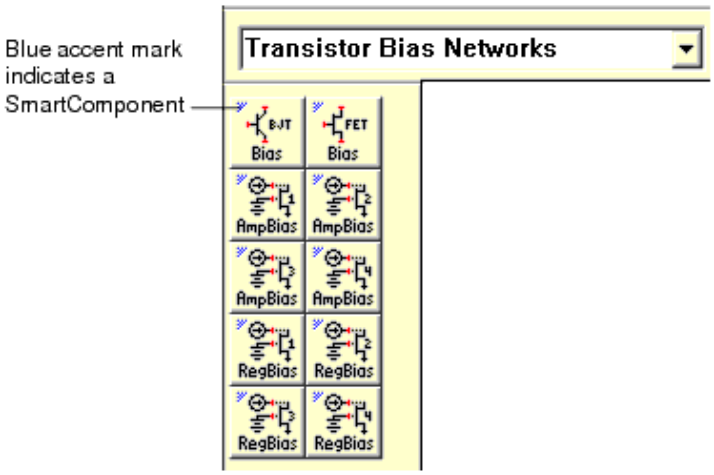
In this step-by-step example, you design and analyze a resistive bias network. A resistive bias network can be designed easily by using the default component parameter settings. Using the Utility follows a normal design flow procedure:

- Select and place a component needed for your design from the component palette (Displaying the SmartComponent Palette and Placing Example Component in the Design).
- Provide specifications (Changing SmartComponent Parameters).
- Design and analyze the component (Designing the SmartComponent).

Note
Before starting this section of the step-by-step example, confirm your setup (Setting Up the Design Environment).

Displaying the SmartComponent Palette

The DesignGuide contains a SmartComponent palette, *Transistor Bias Networks*, that provides quick and easy access to the SmartComponents. A blue accent in the upper-left corner of a palette button indicates the component is a SmartComponent.



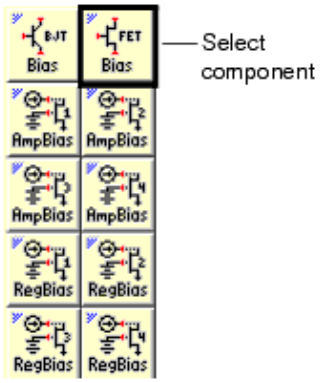
You can display the SmartComponent palettes in one of the following ways:

- Click *Component Palette* on the Control window toolbar.
- Click **View > Component Palette** from the Control window menu.
- Select *Transistor Bias* palette from the *Component Palette* drop-down list box in the *Schematic* window toolbar (directly above the palette).
  Continue the example by selecting the Transistor Bias palette. The palette displays in the *Schematic* window.

Placing Example Component in the Design

To place a SmartComponent in the design:

1. Click **FET Bias** on the component palette to select the component.

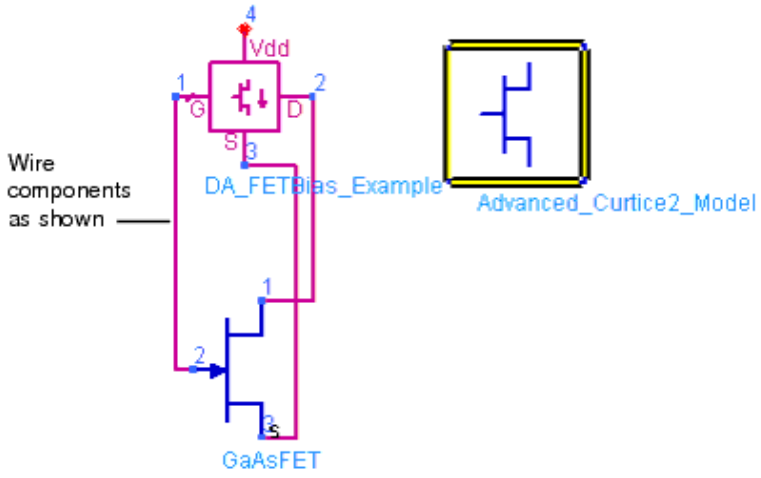


2. Click within the schematic window to place the component.
   - You can change the orientation of the SmartComponent *before* placement by selecting from the *Insert > Component > Component Orientation* commands or by selecting *Rotate by -90* repeatedly from the schematic toolbar.
   - The place component mode remains active until you choose End Command from the schematic toolbar.

> **Note**
> When a SmartComponent is placed initially, a temporary component is used to place and specify the parameters for the SmartComponent. This component *does not* contain a subnetwork design. After the utility has been used to design the SmartComponent, the temporary component is replaced with a permanent component. The SmartComponent is renamed to *DA_ComponentName_DesignName* and an autogenerated design is placed inside the SmartComponent's subnetwork design file. Subsequently, if the SmartComponent parameters are edited, the utility must be used again to update the subnetwork design file.

Continue the example by placing and wiring the remaining components:

1. Display the **Devices-GaAs** palette.
2. Place a **GaAsFET (GAASN)** device and an **Advanced Curtice 2 model (AdvCr2)** into the design.
3. Wire the gate, drain, and source of the device to the appropriate SmartComponent pins, as shown in Wire the Gate, Drain, and Source in the Example. The Vdd pin does not need to be connected at this time.



Wire the Gate, Drain, and Source in the Example.

Changing SmartComponent Parameters

Parameters can be changed directly from the DesignGuide Control window.
To edit the FETBias component parameters:

1. In the Control window, select the **FETBias** component from the SmartComponent drop-down list. This ensures all changes are referenced to this component.
2. Select the **Resistive Networks** tab.
3. Set *Vds* (drain to source voltage) to *3V* and *Id* (drain current) to *1 mA* on the control window **Bias Settings** edit boxes. Leave all other parameters at default.

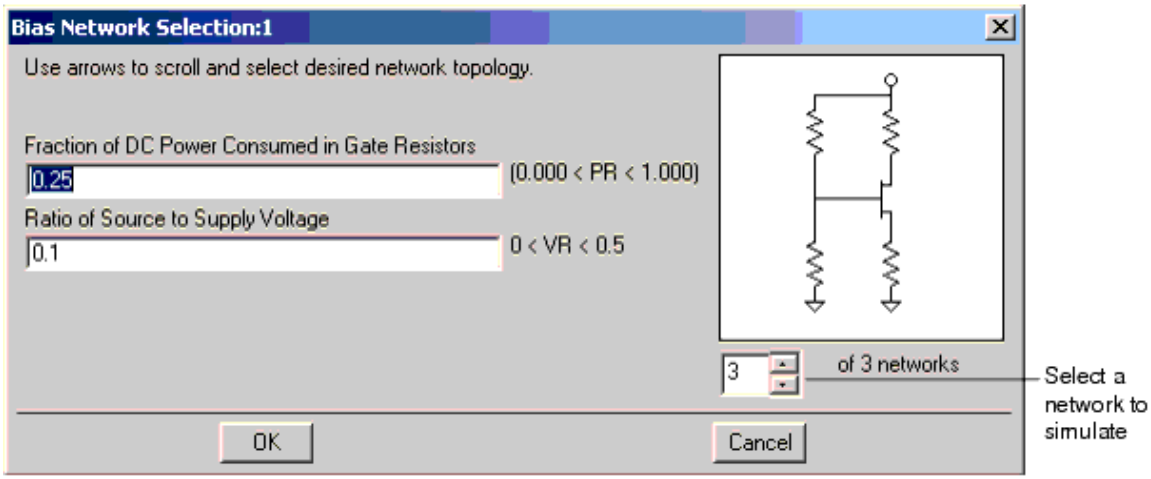
Note
See Placing and Editing SmartComponents for details on changing parameters in the design window or component dialog box.

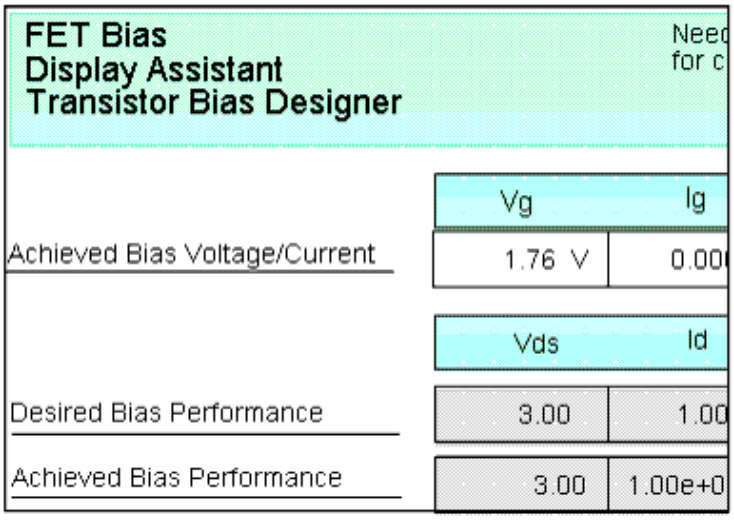
Designing the SmartComponent

You can design and analyze the SmartComponent from the Control Window.

To start the simulation:

1. On the **Resistive Networks** tab, click **Design** to start a simulation. The simulation determines the DC parameters of the device at the selected bias point. When the simulation has finished, a Bias Network Selection dialog box appears.
2. Select one of the networks and click OK to start a second simulation. (Networks that appear in gray cannot be designed for the current parameter settings.) A second simulation takes place and a data display window summarizing the DC performance of the device appears.





Closing the FETBias Analysis Results Window

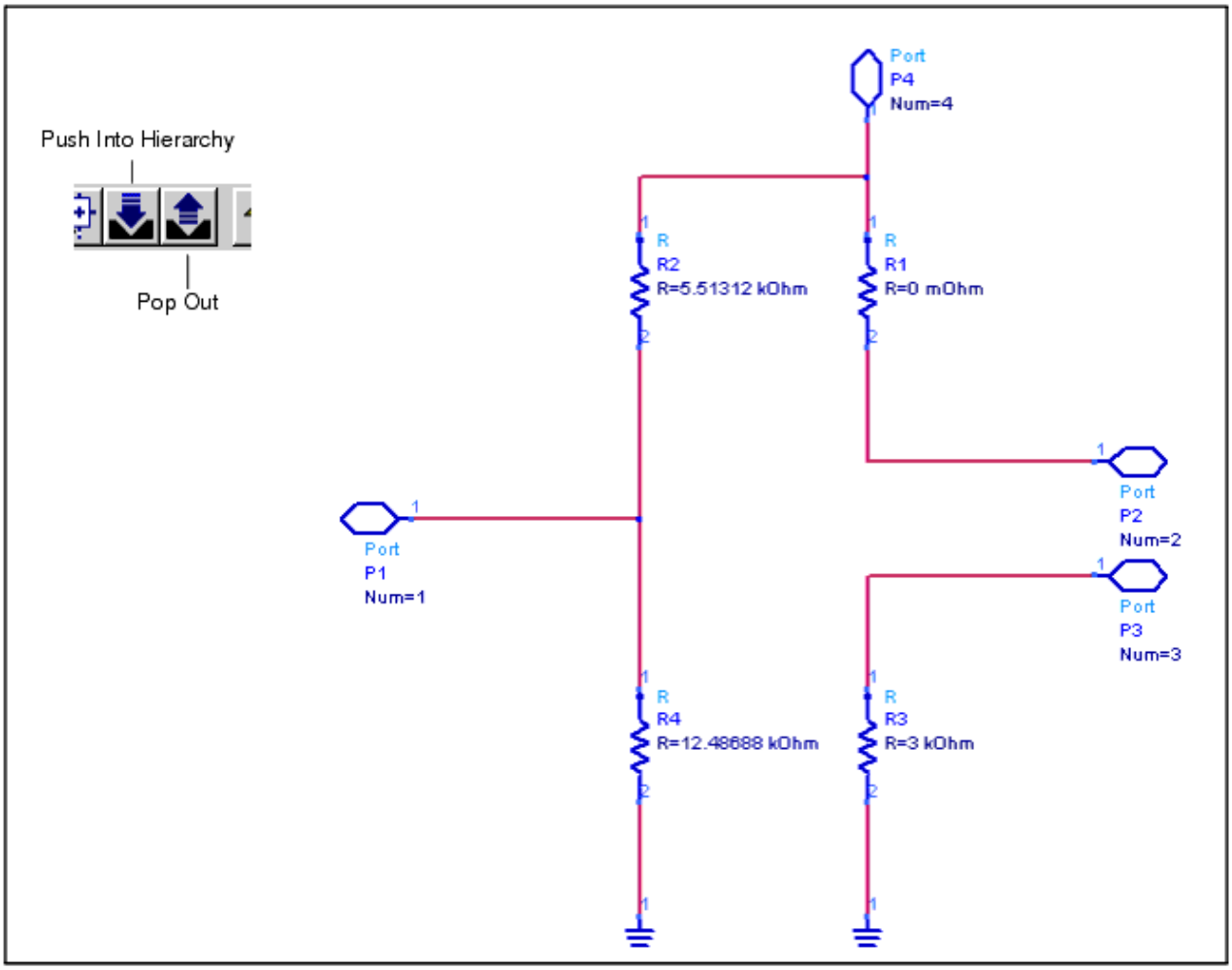
- Click **File** > **Close Window** to close the display window.

Examining the Component Design

You can look at the details of the autogenerated design inside the SmartComponent's subnetwork.

To examine the component's subnetwork:

1. Select the component **FETBias**.
2. Click **Push Into Hierarchy** on the schematic toolbar.
3. After examining the design, click **Pop Out** on the schematic toolbar to close the view.



Deleting the SmartComponent

- Click **Tools** > **Delete SmartComponent** from the DesignGuide control window to delete the FETBias SmartComponent.

> **Note**
> The *Delete* button on the DesignGuide control window is different from the *Delete* button on the ADS Schematic window toolbar.

This completes the step-by-step example.

Using SmartComponents

This Utility provides several SmartComponents representing resistive and active bias networks. SmartComponents are smart sub-network designs that provide the container for specification parameters and a schematic representation of the design when placed into a design. The utility provides automated design and analysis for these SmartComponents.
SmartComponents can be placed, copied, edited and deleted like other components in the

Advanced Design System. The basics of placement, copying, editing and deleting are described briefly in this section.

For help with ADS basic features, refer to the *Schematic Capture and Layout* (usrguide) documentation.

Placing and Editing SmartComponents

The components are placed in the schematic by selecting the SmartComponent from the palette and clicking at the point where you want to place the component in the schematic.

Placing SmartComponents

To place a SmartComponent in the design:

1. In the Schematic window, select the component from the SmartComponent palette.
2. Click within the design window at the location where you want to place the SmartComponent.
   - You can change the orientation of the SmartComponent *before* placement by selecting from the *Insert* > *Component* > *Component Orientation* commands or by selecting *Rotate by -90* repeatedly from the schematic toolbar.
   - The place component mode remains active until you choose End Command from the schematic toolbar.

Changing Position and Orientation

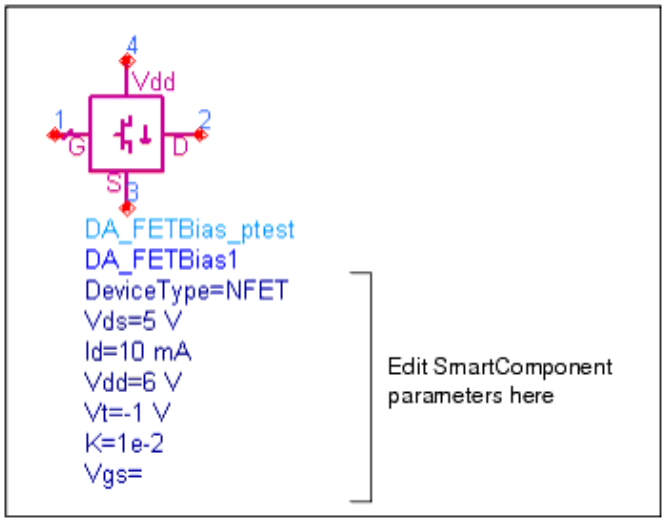
A SmartComponent is moved by dragging it to any location in the Schematic window. To change the component's orientation:

1. Select **Edit** > **Advanced Rotate** > **Rotate Around Reference** from the Schematic window or select **Rotate Items** from the toolbar.
2. Click the SmartComponent you want to use.
3. Rotate the component. The rotate mode remains active until you select End Command from the toolbar.

Editing SmartComponents

Specifications of the SmartComponent are entered directly on the Resistive Networks or Active Networks tab on the Control window. You can also modify the specifications in one of these ways:

- Click the SmartComponent parameters in the schematic window and change them (see The FET Bias Network Component.)
- Double-click the SmartComponent to open a dialog box containing all parameters



The FET Bias Network Component

The SmartComponent design (schematic) can be viewed by pushing into the SmartComponent's subnetwork. See Examining the Component Design.

A SmartComponent subnetwork is empty until the design is generated (see the note in the section Placing and Editing SmartComponents).

Copying SmartComponents

SmartComponents can be copied within a design, to another design, or to another Schematic window.

Copying Within a Design

To copy a SmartComponent to the same design:

1. Click the SmartComponent to be copied.
2. Select **Edit** > **Copy** and then **Edit** > **Paste** from the schematic window.
3. Click where you want the copy placed.

Copying Between Designs or Schematic Windows

To copy a SmartComponent to another design:

1. Click the SmartComponent to be copied.
2. Select **Edit** > **Copy** from the Schematic window.
3. Display the design or schematic window you want to copy the SmartComponent to.
4. Select **Edit** > **Paste** to copy the SmartComponent to the design.
5. Click where you want the component placed.

Copying a SmartComponent as a Unique Design

Initially, all copied SmartComponents refer to the same SmartComponent design. When the *Design Assistant* is used to perform a design operation, the Design Assistant transforms each copied SmartComponent into a unique SmartComponent design. A design operation is accomplished from the Utility Control Window.

Deleting SmartComponents

SmartComponents can be deleted from a design like other components, but completely removing a SmartComponent's files requires the actions described here.

Deleting from Current Design

A SmartComponent can be deleted from a design in one of these ways:

- Select the component and click **Delete**,
- Click **Delete** from the toolbar,
- Click **Edit** > **Delete** from the schematic window.

> **Note**
> This procedure does not remove the SmartComponent files from the workspace directory. To delete files from the workspace directory, see Deleting from Current Workspace.

Deleting from Current Workspace

To delete a SmartComponent and all associated files from your workspace:

1. From the DesignGuide Control window, click **Tools** > **Delete SmartComponent** or on the toolbar, click **Delete SmartComponent**.
2. Click the SmartComponent you want to delete. This deletes the SmartComponent from the current design and removes all of its files from your workspace. The SmartComponent delete mode remains active until you select End Command from the schematic toolbar.

Deleting Manually Using File System

You can use your computer's file system to delete a SmartComponent by deleting the appropriate files in the network subdirectory of a workspace. Delete files that start with *DA_* or *SA, contain the SmartComponent title, and end with _.ael, .atf,* or *.wrk*.

Using SmartComponents as Standalone Components

After SmartComponents are designed and tested, they can be used as standalone components. The Bias Utility is not needed to use them in new designs unless you wish to modify or analyze them. When using the SmartComponent in a design, however, the power supply pins (Vdd, Vcc, Vp, Vm) must be connected to a DC voltage source whose voltage level corresponds the parameter setting.

Using an Existing SmartComponent Within the Same Workspace

To use an existing SmartComponent within the same workspace:

1. Open the Component Library window by selecting **Insert** > **Component** > **Component Library** from the Schematic window or **Display Component Library List** on the toolbar.
2. Select the Library name under **All Libraries** list at the left of the Component Library window. Available components are listed in the Components list at the right of the Component Library window.
3. Select the SmartComponent in the Components list.
4. Place the SmartComponent into your schematic by clicking in the Schematic window at the location you wish it placed. The insert mode remains active until you click **End Command**.

Using an Existing SmartComponent in Any Workspace

A library of predesigned reusable SmartComponents can be created by placing the reusable SmartComponents in a workspace. This workspace can be included in any workspace and its SmartComponents can be accessed using the Component Library.
To use an existing SmartComponent in any workspace:

1. Open the Workspace where the SmartComponent needs to be inserted.
2. Open the Library in the Workspace by selecting **File > Open > Library**.
3. Open the Component Library window by selecting **Insert** > **Component** > **Component Library** from the Schematic window or **Display Component Library List** from the toolbar.
4. Select the Library name under **All Libraries** list at the left of the Component Library window. Available components are listed in the Components list at the right of the Component Library window.
5. Select the SmartComponent in the Components list.
6. Place the SmartComponent into your schematic by clicking in the Schematic window at the location you where you want to place the component. The insert mode remains active until you click **End Command**.

Automated Design and Analysis

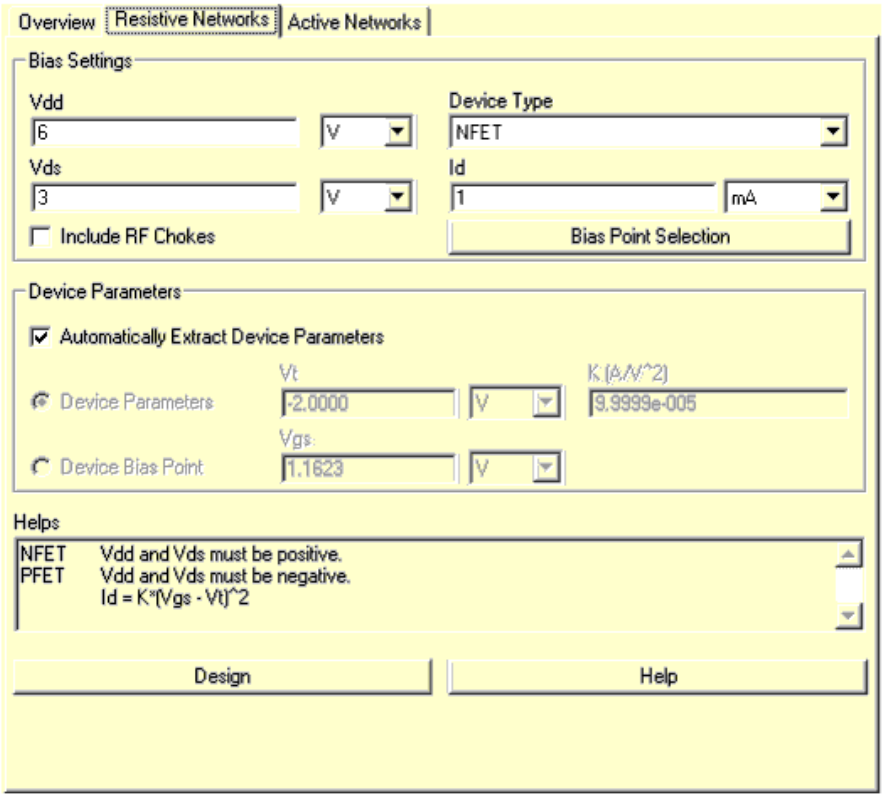
The Automated Assistants provide quick design and performance analysis for SmartComponents. Two Automated Assistants are available in this Utility for design:

- Resistive Networks is used to design and simulate the performance of resistive bias networks for BJT and FET devices.
- Active Networks is used to design and simulate the performance of active bias networks for BJT and FET devices.

Explore each tab page by selecting the associated tab on the control window.

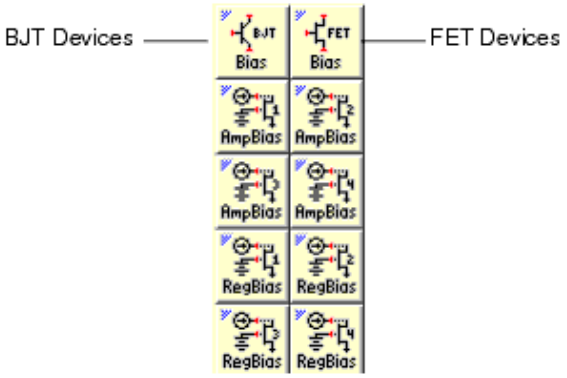
Resistive Networks

The Resistive Networks tab is used to generate and update the design contained within a resistive bias network SmartComponent from the given specifications. This tool is accessed using the Bias Utility control window. From the control window, full design control is enabled from the *Resistive Networks* tab. Component design operations can also be accomplished using the control window menu and toolbar. Any parameter change made from the *Resistive Networks* tab is reflected on the SmartComponent in the schematic.



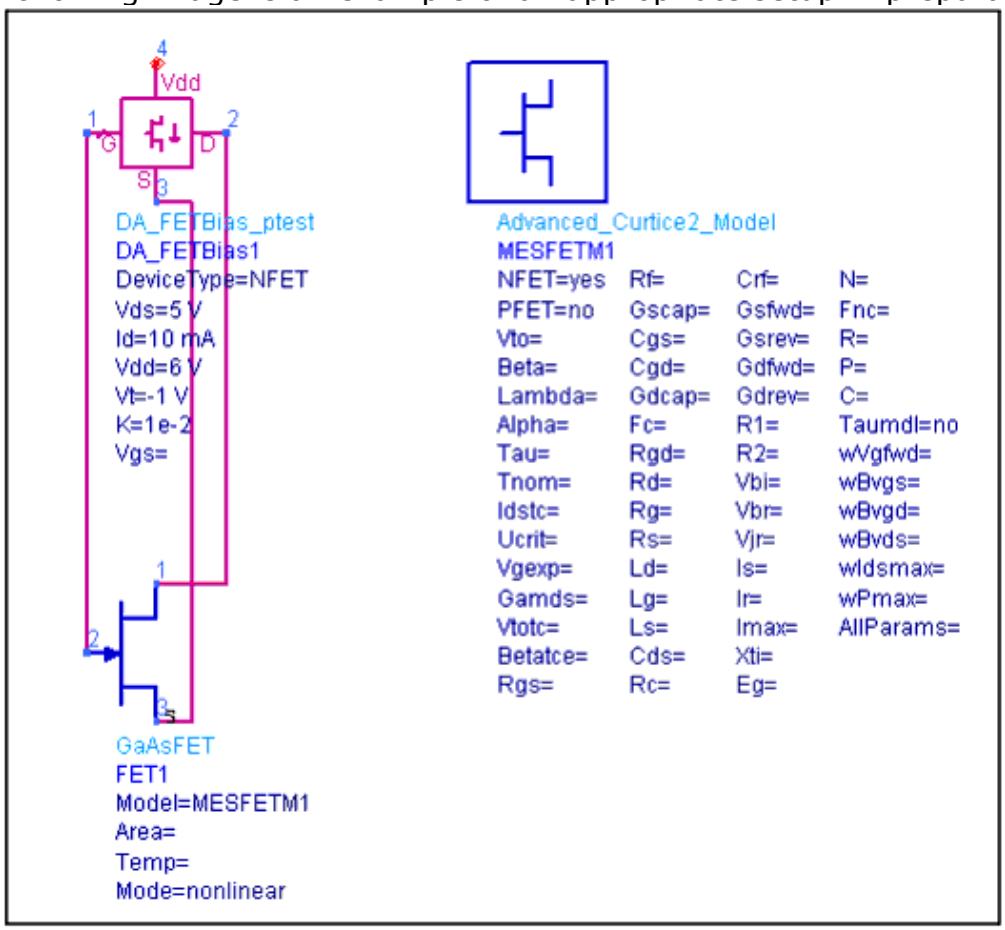
Resistive Bias Network SmartComponents

Resistive bias networks can be designed for NPN BJT, PNP BJT, NFET, or PFET devices. Two different SmartComponents are available on the Utility palette as shown.



Before designing a network, the SmartComponent pins must be wired to the corresponding pins of the device for which the bias network is to be designed. If the device also has a model associated with it, then this model must be placed on the schematic as well. If the device has pins that should be grounded, this grounding must also be done before a design is attempted.

The SmartComponent supply pin (Vdd or Vcc) does not need to be connected at this time. However, when the SmartComponent is used in a design, this supply pin must be connected to a DC voltage source set at the appropriate supply voltage level. The following image is an example of an appropriate setup in preparation for design.



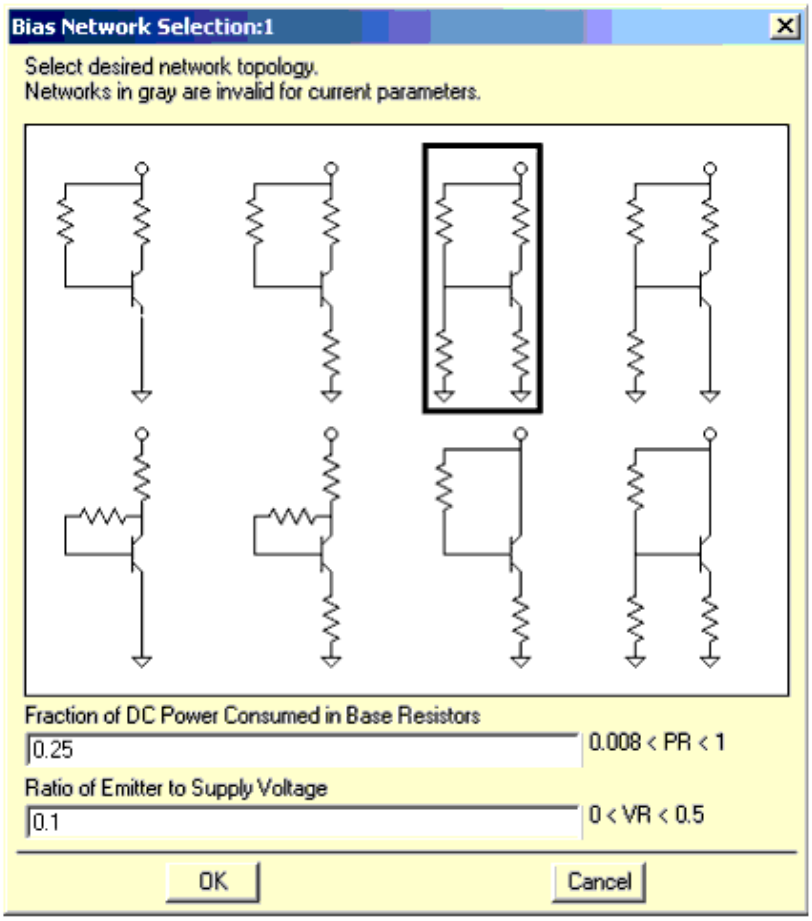
Appropriate Design Setup for Resistive Bias Network

BJT Networks

Resistive bias networks for BJT devices have the following options:
Vcc - DC supply voltage value. This is the DC voltage that will be connected to the collected side of the bias network.
Vce - Target bias point collector-to-emitter voltage.
Ic - Target bias point collector current.
Device Type - BJT device type. For NPN design, all voltages must be positive. For PNP design, all voltages must be negative.
Include RF Chokes - If this option is set, the design incorporates RF choke (DC Feed) elements to isolate the bias network from the RF signal.

Automatically Extract Device Parameters - The DC operation of the device near the operating point is modeled as $Ic = beta \times Ib$, where beta is a device parameter and Ib is the base current at the desired operating point. Furthermore, the device is characterized by a base-emitter voltage drop Vbe at the desired operating point. If the option *Automatically Extract Device Parameters* is set, the utility attempts to extract beta and Vbe parameters using a simulation. If the target bias point is inappropriate for the device, then this extraction can fail. Alternately, the parameters beta and Vbe can be manually specified.

After parameters have been specified and you have pressed *Design* , the utility starts the design process. If requested, the device parameters are extracted first. If this extraction is successful, then a dialog opens so you can select the bias network topology. Any networks appearing in gray cannot be designed for the current bias parameters. If all networks are gray, then the bias settings must be altered.



For 3 or 4 resistor topologies, additional specifications must be provided:
Fraction of DC Power Consumed in Base Resistors - This parameter sets the current through the base bias network when two resistors are used. The specification is made in terms of the fraction of the total resistive power dissipated in the device that is due to the base resistors.
Ratio of Emitter to Supply Voltage - For specifying the emitter resistance, the emitter voltage (specified relative to the supply voltage Vcc) must be specified.

After a suitable network topology has been selected (indicated by a box around the topology), pressing OK completes the design and simulation. A display window opens showing the achieved performance of the network.

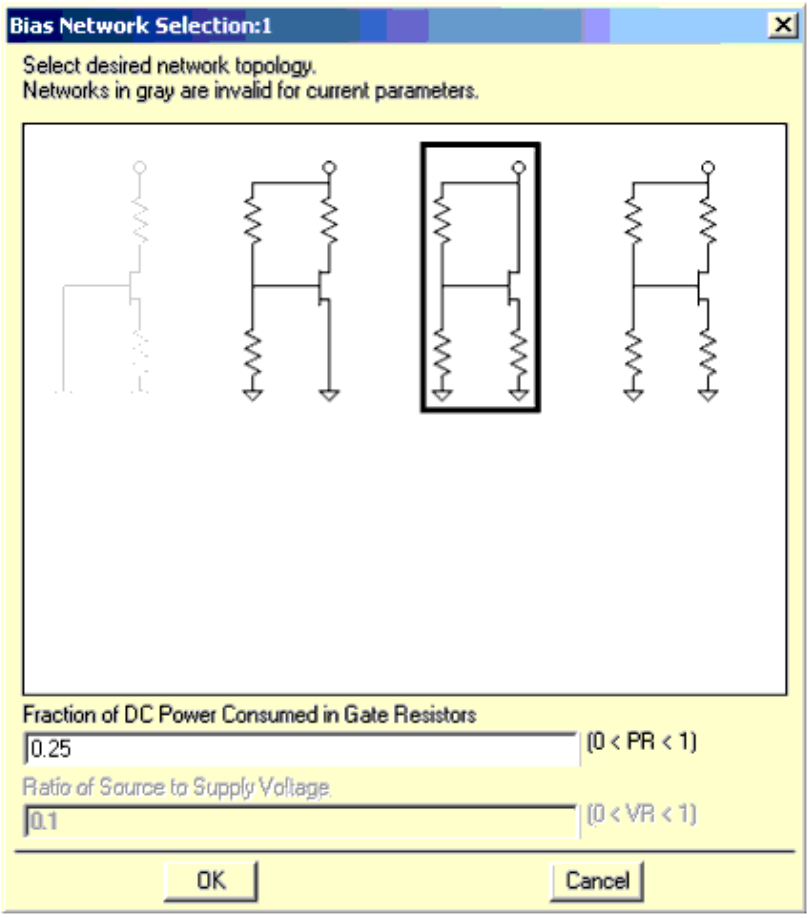
FET Networks

Resistive bias networks for FET devices have the following options:

Vdd - DC supply voltage value. This is the DC voltage that will be connected to the drain side of the bias network.
Vds - Target bias point drain-to-source voltage.
Id - Target bias point drain current.
Device Type - FET device type. For NFET design, Vdd and Vds must be positive. For PFET design, Vdd and Vds must be negative.
Include RF Chokes - If this option is set, the design incorporates RF choke (DC Feed) elements to isolate the bias network from the RF signal.

Automatically Extract Device Parameters - The DC operation of the device near the operating point is modeled as $Id = K \times (Vgs - Vt)^2$, where K and Vt are device parameters and Vgs is the gate-to-source voltage at the desired operating point. If the option *Automatically Extract Device Parameters* is set, the utility attempts to extract these parameters using a simulation. If the target bias point is inappropriate for the device, then this extraction can fail. Alternately, either the parameters K and Vt or the bias point Vgs can be manually specified.

After parameters have been specified and you have pressed *Design* , the utility starts the design process. If requested, the device parameters are extracted first. If this extraction is successful, then a dialog opens so you can select the bias network topology. Any networks appearing in gray cannot be designed for the current bias parameters. If all networks are gray, then the bias settings must be altered.



For 3 or 4 resistor topologies, additional specifications must be provided:
Fraction of DC Power Consumed in Gate Resistors - This parameter sets the current through the gate bias network when two resistors are used. The specification is made in terms of the fraction of the total resistive power dissipated in the device that is due to the gate resistors.

Ratio of Source to Supply Voltage - For specifying the source resistance, the source voltage (specified relative to the supply voltage Vdd) must be specified.

After a suitable network topology has been selected (indicated by a box around the topology), pressing OK completes the design and simulation. A display window opens showing the achieved performance of the network.

Bias Point Selection for Resistive Bias Networks

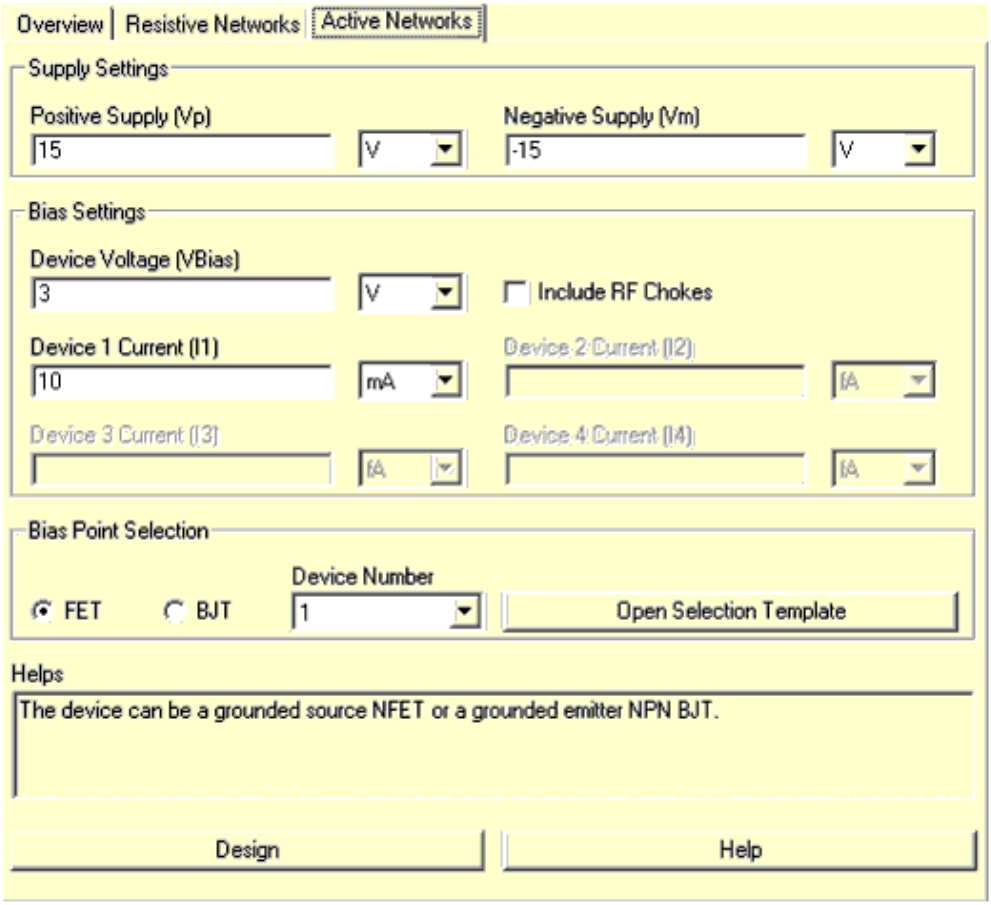
Typically, selection of the bias point is performed based upon specifications provided by device manufacturers. To assist in this selection process, simulation and display templates are provided. You can use these templates to choose the bias point based upon optimal Class A operation for power amplifiers, or to achieve target gain or noise figure specifications for small-signal amplifiers. The templates contain text on the schematic and display windows indicating the sequence of steps to follow to assess the device performance.

After a device bias point has been determined from these templates, the schematic

template must be closed and the design containing the original SmartComponent must be visible before the design can proceed.

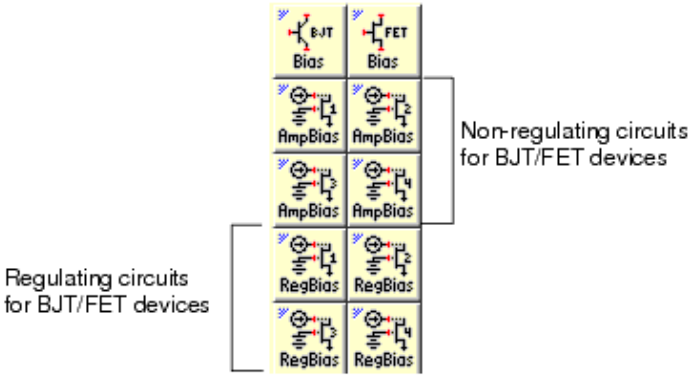
Active Networks

The Active Networks tab is used to generate and update the design contained within an active bias network SmartComponent from the given specifications. This tool is accessed using the Bias Utility control window. From the control window, full design control is enabled from the *Active Networks* tab. Component design operations can also be accomplished using the control window menu and toolbar. Any parameter change made from the *Active Networks* tab is reflected on the SmartComponent in the schematic.



Active Bias Network SmartComponents

Active bias networks can be designed for NPN BJT or NFET devices. Eight different SmartComponents are available on the Utility palette as shown.

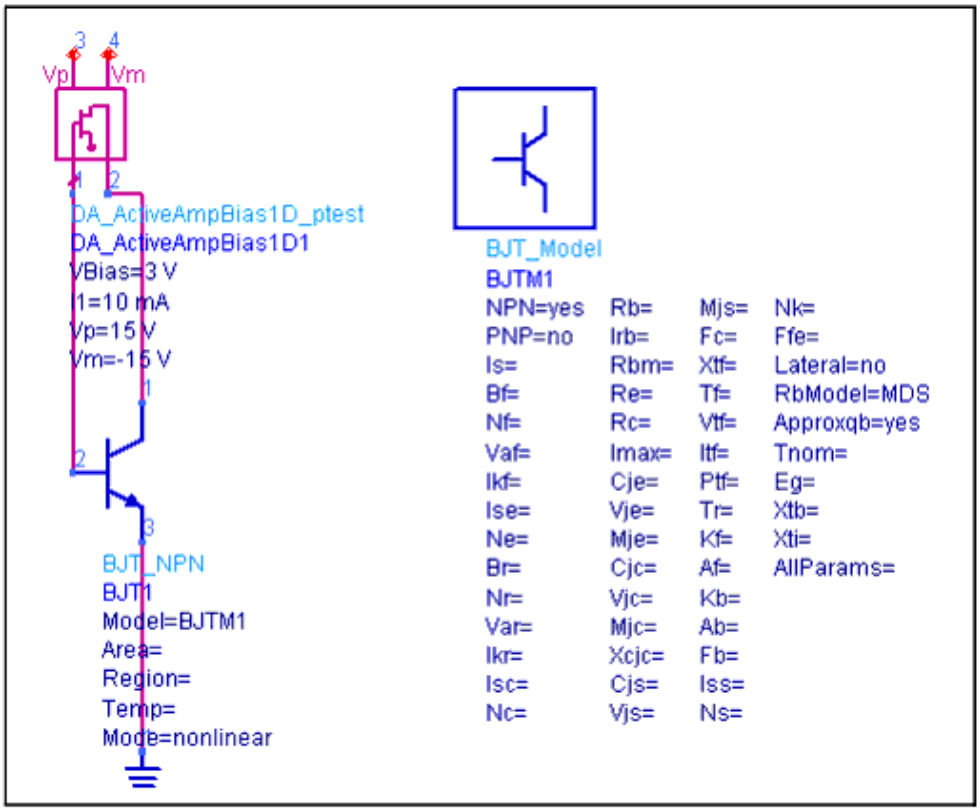


Active bias networks use operational amplifiers to create a network that can offer the specified bias point independent of the device characteristics. The bias voltage and current are set independently. A non-regulating (OpAmp based) design provides a simple network with low parts count. However, the performance can vary as a function of the tolerances of the parts used to fabricate the network. The regulating designs use a zener diode to provide a more tolerant design at the expense of a more complicated network.

Before designing a network, the SmartComponent pins must be wired to the corresponding pins of the device for which the bias network is to be designed. If the device also has a model associated with it, then this model must be placed on the schematic as well. If the device has pins that should be grounded, then this grounding must also be done before a design is attempted. Since these bias networks can be used only for grounded source FET or grounded emitter BJT devices, these pins on the device must be grounded.

The SmartComponent supply pins (Vp and Vm) do not need to be connected at this time. However, when the SmartComponent is used in a design, these supply pins must be connected to DC voltage sources set at the appropriate supply voltage levels.

For each type of network (non-regulating and regulating), the four SmartComponents available can accommodate different numbers of devices (1 through 4 devices). All devices biased by a given SmartComponent must share the same bias voltage (Vce or Vds), but can have independent bias currents.
Appropriate Design Setup for Active Bias Networks is an example of an appropriate setup in preparation for design.



Appropriate Design Setup for Active Bias Networks

Network Design

Active bias networks for NPN BJT and NFET devices have the following options:
Positive Supply (Vp) - DC positive supply voltage value. This DC voltage runs the operational amplifiers that are used to create the bias.
Negative Supply (Vm) - DC negative supply voltage value. This DC voltage runs the operational amplifiers that are used to create the bias.
Device Voltage (VBias) - Target bias point collector-to-emitter or drain-to-source voltage for all devices.
Device Current (I) - Target bias point collector or drain current. Each device can have a unique bias current.
Include RF Chokes - If this option is set, the design incorporates RF choke (DC Feed) elements to isolate the bias network from the RF signal.

After parameters have been specified and you have pressed *Design*, the utility starts the design process. A simulation is performed after the design is complete, and a display window opens showing the achieved performance.

Regulating Networks

The Regulating Bias Networks (RegBias) use a zener diode to regulate the actual bias voltage level achieved. This makes for a design that is more tolerant to variations in

component values. Furthermore, these networks use RC networks such that the drain voltage is applied before the gate voltage is applied. For many FET devices, this dramatically reduces device failure due to damaged gate oxide.

For these networks, the zener diode voltage (*Vzener*) and maximum power rating of the resistors used in the network (*Pmax*) must be specified. These parameters cannot be specified on the control window, and therefore must be specified directly on the SmartComponent on the schematic window or using the parameter dialog box that appears by double-clicking on the SmartComponent.

Bias Point Selection for Active Bias Networks

Typically, selection of the bias point is performed based upon specifications provided by device manufacturers. To assist in this selection process, simulation and display templates are provided. You can use these templates to choose the bias point based upon optimal Class A operation for power amplifiers, or to achieve target gain or noise figure specifications for small-signal amplifiers. The templates contain text on the schematic and display windows indicating the sequence of steps to follow to assess the device performance.

To use this capability for Active Bias networks, the number of the device (1-4) as well as the device type (BJT or FET) connected to the SmartComponent must be specified. Pressing *Open Selection Template* opens the appropriate schematic and display templates.

After a device bias point has been determined from these templates, the schematic template must be closed and the design containing the original SmartComponent must be visible before the design can proceed.