Agilent Technologies

Advanced Design System 2002

DesignGuide Developer Studio

February 2002

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Agilent Technologies
395 Page Mill Road
Palo Alto, CA 94304 U.S.A.

Contents

3 Menu Editor

4 Palette Editor

5 Bitmap Editor

10 DesignGuide Style Guide

Chapter 1: Getting Started With DesignGuide Developer Studio

The DesignGuide Developer Studio is designed to help developers quickly and easily create and use custom DesignGuides for use in Advanced Design System. The following section provides basic introduction to the DesignGuide Developer Studio components and how to use them to create a custom DesignGuide.

Using DesignGuides

The DesignGuide Developer Studio is accessed from the DesignGuide menu in the ADS Main window.

The commands in this menu are as follows:

DesignGuide Studio Documentation > **Developer Studio Documentation** is only available on this menu if you have installed the DesignGuide Developer Studio. It brings up the DesignGuide Developer Studio documentation. Another way to access the Developer Studio documentation is by selecting *Help* > *Topics and Index* > *DesignGuides* > *DesignGuide Developer Studio* (from any ADS program window).

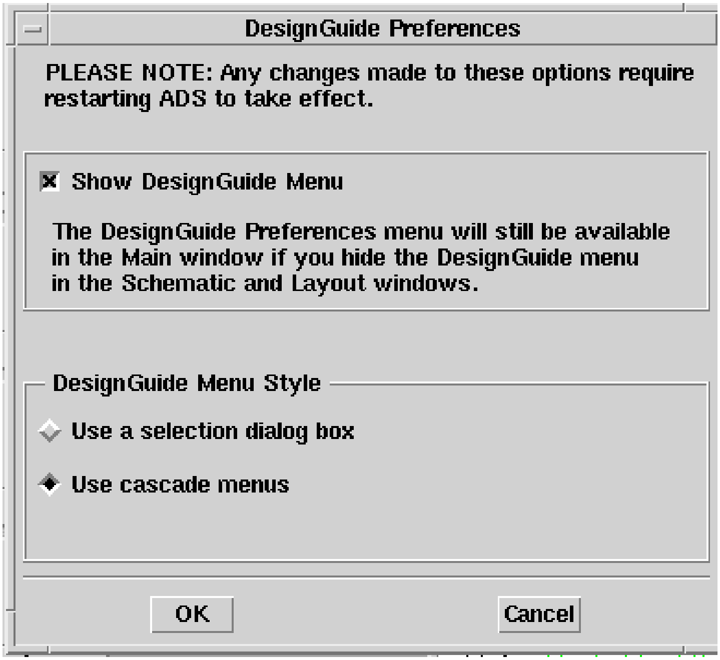
DesignGuide Developer Studio > **Start DesignGuide Studio** is only available on this menu if you have installed the DesignGuide Developer Studio. It launches the initial Developer Studio dialog box.

Add DesignGuide brings up 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 brings up a list of your installed DesignGuides. Select any that you would like to uninstall and choose the *Remove* button.

Preferences brings up a dialog box that is not applicable to the DesignGuide Developer Studio itself, but for all other installed DesignGuides (such as Bluetooth or RF System), it allows 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 will be removed if this box is unchecked.
- Select your preferred interface method (cascading menus vs. dialog boxes).



PLEASE NOTE: Any changes made to these options require restarting ADS to take effect.

☒ Show DesignGuide Menu

The DesignGuide Preferences menu will still be available in the Main window if you hide the DesignGuide menu in the Schematic and Layout windows.

DesignGuide Menu Style

○ Use a selection dialog box

● Use cascade menus

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. Thus the dialog box menu style might be best for these situations.

Accessing documentation

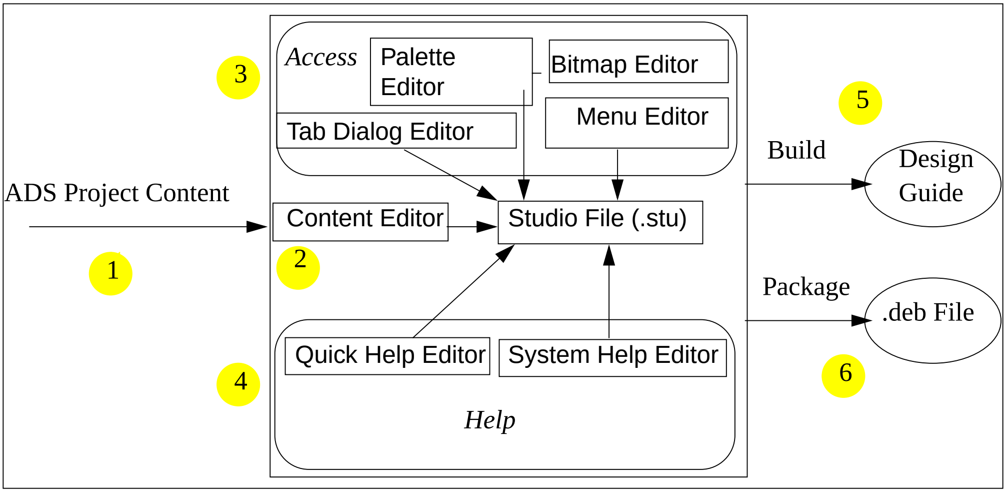
To access the documentation for the DesignGuide, select either of the following:

- **DesignGuide** > **DesignGuide Studio Documentation** > **Developer Studio Documentation** (from ADS Main window)
- **Help** > **Topics and Index** > **DesignGuides** > **DesignGuide Developer Studio** (from any ADS program window)

Developer Studio Overview

The DesignGuide Developer Studio is composed of the following tools:

- Main Window/Content Editor
- Menu Editor
- Palette Editor
- Bitmap Editor
- Tab Dialog Editor
- Quick Help Editor
- System Help Editor



Each individual editor saves its information inside a studio file (file extension *.stu*), so all DesignGuide information is kept in one place for easy saving and loading.

The six main steps in creating a custom DesignGuide follow:

1. Create custom content files using ADS (Schematic, Layout, Display, etc).
2. Add content files to studio project using the Content Editor.
3. Use the Menu, Tab Dialog, and Palette Editors to control access to content files.

4. Add appropriate help files using System and Quick Help Editors.

5. Use *Build* to create DesignGuide for use on current computer.

6. Use *Package* to create packaged files for installation/use on other computers.

The following section demonstrates how to use the DesignGuide Developer Studio quickly to create a DesignGuide.

Summary of Files and Directories

When you start to use the Developer Studio, it places a *studio_files* directory in your $HOME directory. The *studio_files* directory is the location for all project directories for each project (each project has its own folder). Contained in the project folder is the studio file (*xxx.stu*) and the debian file (*xxx.deb*). The *xxx.stu* file contains the menu definition, the palette definition, the tab dialog definition, and so on. When you package the project, it does the following:

- Copies all of the design files, datasets, data display pages, and so on, to the *studio_files/<project name>* directory
- Creates the *xxx.deb* file
- Deletes all of the files that went into the *xxx.deb* file.

The *xxx.deb* file is now the archive file, which is transportable and installable in ADS.

DesignGuide Name is the name of the Debian archive file.

Build Name is the name of the directory used as the project name in the *studio_file* directory and in the $HPEESOF_DIR/designguides/projects directory.

To change the DesignGuide Name or Build Name, refer to the section "Preferences" on page 2-12.

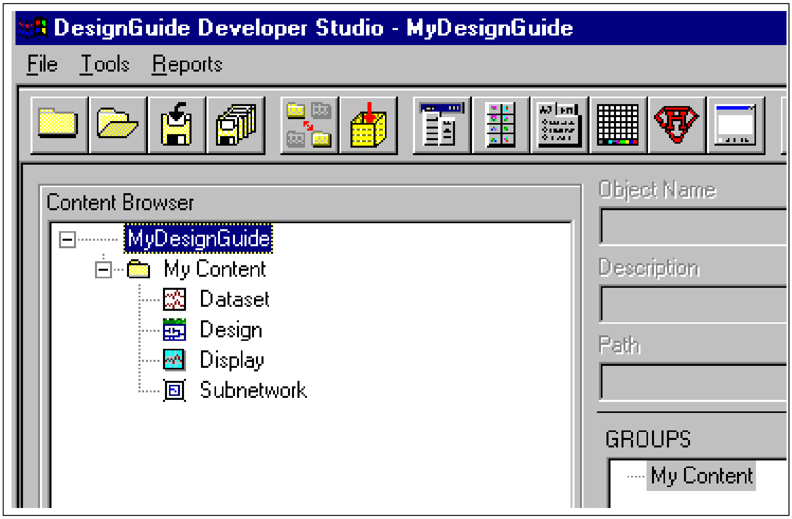
Custom Design Guides can be stored in the *$HOME/hpeesof/designguides/projects* directory (for individual users).

Creating a DesignGuide Menu

The following brief tutorial will take you through the steps of creating a very simple DesignGuide menu that will appear on an ADS Schematic window. We recommend that you review the preceding section, "Summary of Files and Directories" on page 1-5 before proceeding.

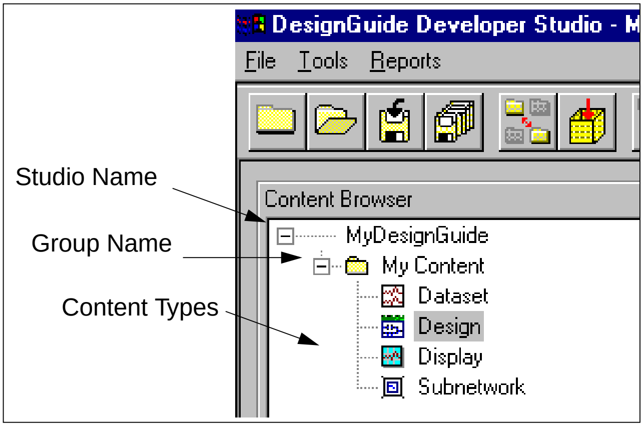
Creating a Studio Project

1. In ADS, create a new project called *MyProject*, and a schematic called *MySchematic*.
2. Start DesignGuide Developer Studio. From the ADS Main window, select **DesignGuide Developer Studio** > **Start DesignGuide Studio**.
3. In the Developer Studio, select **File** >**New.**
4. Type in a new name (*MyDesignGuide*) and select **Create Project.**
5. Under Content Browser, click **My Content**. The new studio project should be displayed as follows:



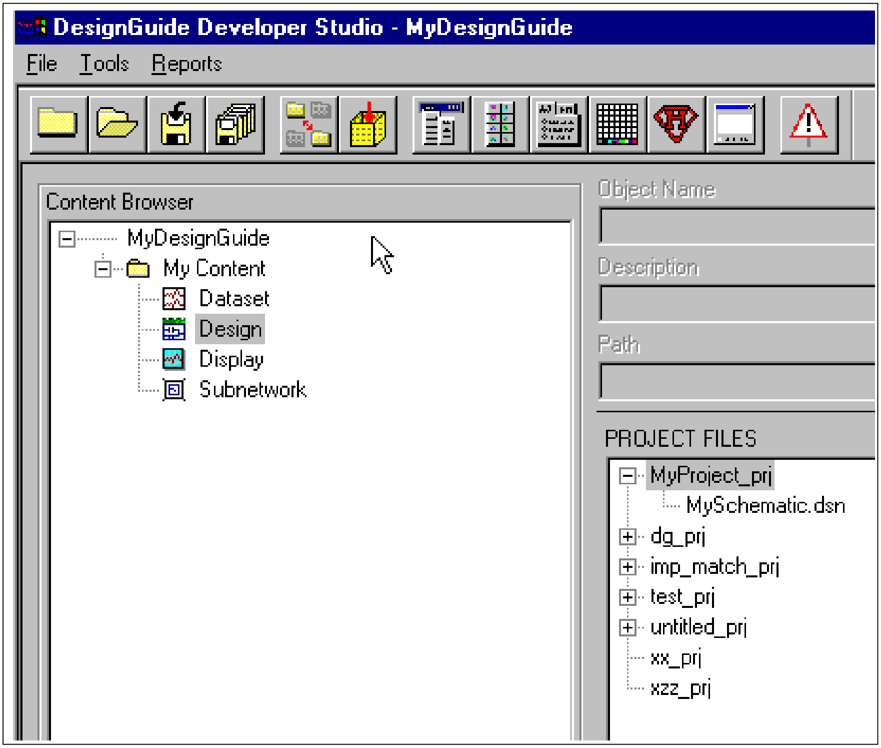
Adding Content

The next step is to add the user-created content to the Content Editor. Following are the components to a studio project:



To add content to the Content Editor:

1. Select **Design** from the list of Content Types. A list of projects will appear in the PROJECT FILES box on the right.
2. Open the project **MyProject_prj.** All *.dsn* files within the project will be displayed.
3. Select **MySchematic.dsn**. *MySchematic* will be added under the *Design* content type.


DesignGuide Developer Studio - MyDesignGuide
File
Tools
Reports
Content Browser
MyDesignGuide
My Content
Dataset
Design
Display
Subnetwork
Object Name
Description
Path
PROJECT FILES
MyProject_prj
MySchematic.dsn
dg_prj
imp_match_prj
test_prj
untitled_prj
xx_prj
xzz_prj

Creating a Menu

The next step in creating a DesignGuide is creating a menu to use the new content. To create a menu:

1. Open the Menu Editor by selecting **Tools > Menu Editor** or the Menu Editor Toolbar item.

2. Replace *MyDesignGuide* in the *Menu Name* text area with an appropriate name for the application (shown as **Amplifier** in this example). This will be the name of the menu in ADS once it is created.

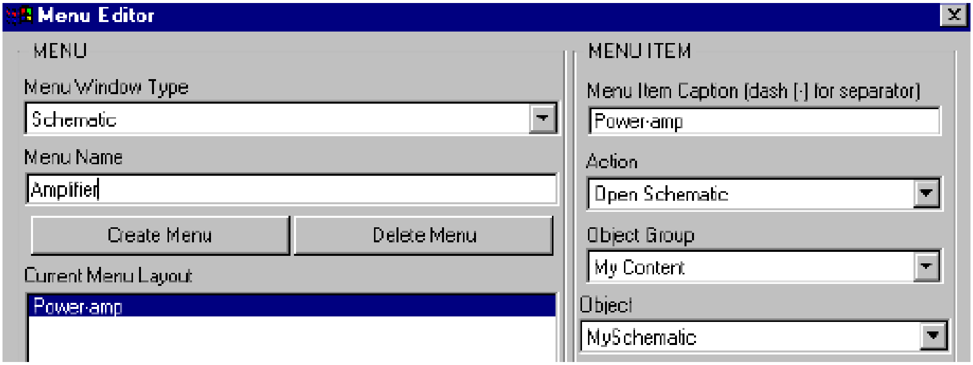
3. In the Menu Item Caption field, enter an appropriate name (**Power-amp** in this example).

4. Select **Power-amp** from the *Current Menu Layout* box.

5. Select **Open Schemati**c from the *Action* list on the right.

6. Select **My Content** from the *Object Group* list (this corresponds to the group names in the Content Editor).

7. Select **MySchematic** from the *Object* list. The Menu Editor should now look as follows:



Menu Editor

MENU

Menu Window Type: Schematic

Menu Name: Amplifier

Current Menu Layout: Power-amp

MENU ITEM

Menu Item Caption (dash [-] for separator): Power-amp

Action: Open Schematic

Object Group: My Content

Object: MySchematic

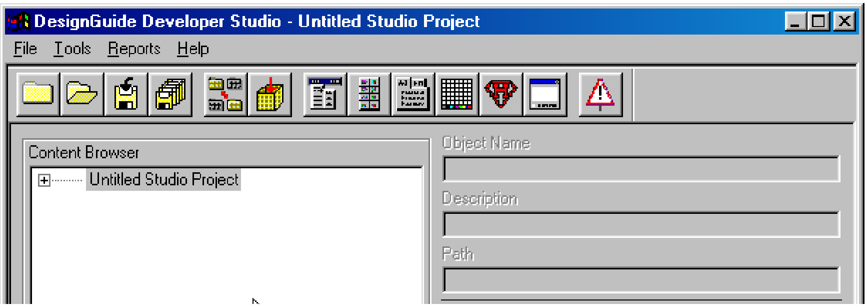
8. Choose **OK.**

Building the DesignGuide

The final step in the design process is to build the DesignGuide for use in any ADS Schematic window.

1. Build the studio project by selecting **File > Build**, or choose the **Build** toolbar button.
2. Select **Yes**.
3. To look at the new menu, close Developer Studio, exit and restart ADS, then open a Schematic window
4. The new menu (*My Menu*) is within the DesignGuide menu, with one menu item (*My Menu Item*).
5. Select **Power-amp,** and the schematic you created will appear.

Main Window/Content Editor

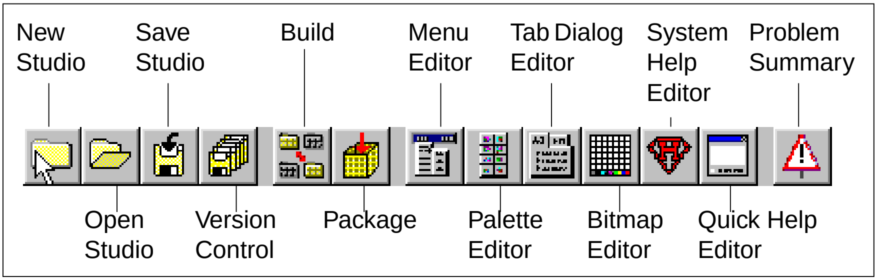


The Content Editor is the first window that opens when the Developer Studio is started. The Content Editor has the following capabilities:

- Create a new studio project.
- Load an existing studio project.
- Add user-created content files to the project.
- Create file dependencies between content files.
- Open the other DesignGuide Developer Studio editors.
- Save and build the current studio project.

- Package the DesignGuide for easy distribution and installation.
- Organize and store previous versions of the studio project as a means of backup.

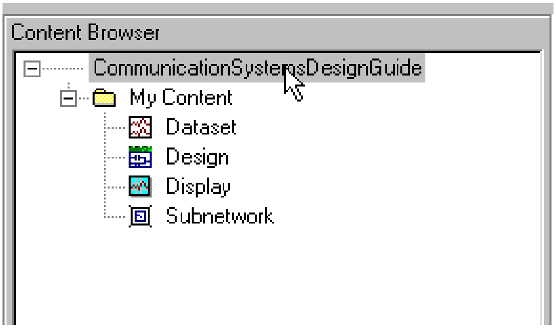
In the following, observe the Content Editor toolbar and the actions associated with each button. Each Content Editor component can also be called using the corresponding menu item. (Refer to Chapter 2, Content Editor.)



Creating a New Studio Project

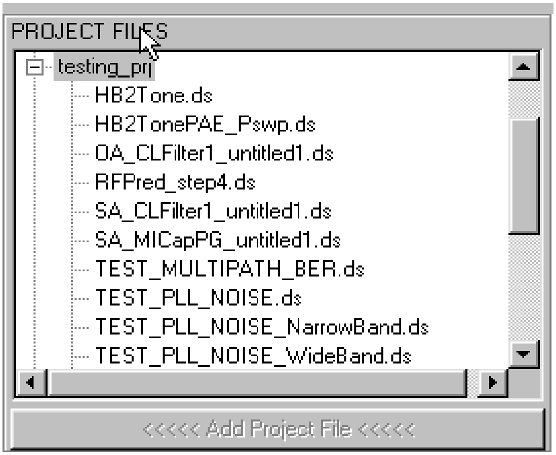
The Content Browser box (shown here) contains the current studio project. To create a new Studio Project

:



1. Select **File > New.**
2. Type in the name of the new studio project (or keep the default).
3. Select **Create Project**

The new studio project appears in the *Content Browser* area with one group, *My Content*, already created.


PROJECT FILES
testing_prj
HB2Tone.ds
HB2TonePAE_Pswp.ds
OA_CLFilter1_untitled1.ds
RFPred_step4.ds
SA_CLFilter1_untitled1.ds
SA_MICapPG_untitled1.ds
TEST_MULTIPATH_BER.ds
TEST_PLL_NOISE.ds
TEST_PLL_NOISE_NarrowBand.ds
TEST_PLL_NOISE_WideBand.ds
<<<<< Add Project File <<<<<

Adding Content

To add content files to the studio project:

[1] Select a content type (**Dataset, Design, Display, Subnetwork)** from *Content Browser.*

4. Select a project directory from the *Project Files* box in the lower right.
5. Select the content files to be added and choose the *Add Project File* button.
6. The content files are then added to the *Content Browser* box.

To remove content, select the content file in the *Content Browser* box and choose the **Remove** button.

Building a Project

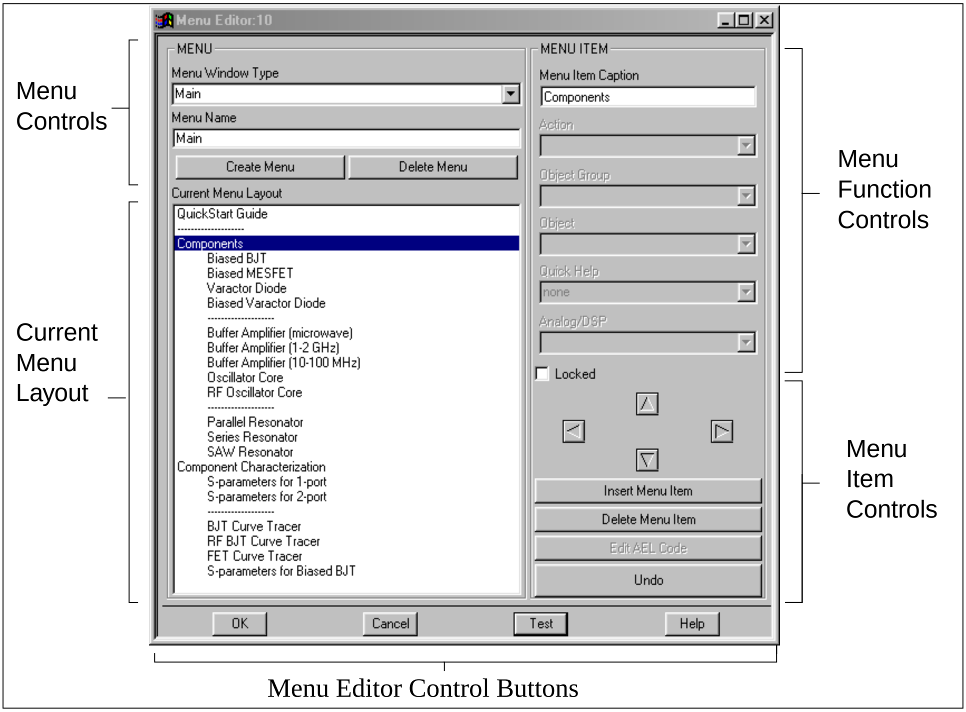
To build the current DesignGuide for use on the same computer, choose the **Build** button on the Content Editor. This will create the DesignGuide for use the next time ADS is loaded.

Packaging a Project

The Package utility packages the DesignGuide for installation on other computers. Choose the **Package** button on the Content Editor (this will also automatically build the DesignGuide) to create a *.deb* file for installation on other computers.

For more detailed information, refer to Chapter 2, Content Editor.

Menu Editor



Menu Window Type: Main

Menu Name: Main

Current Menu Layout:

```
QuickStart Guide
--------------------
Components
        Biased BJT
        Biased MESFET
        Varactor Diode
        Biased Varactor Diode
        --------------------
        Buffer Amplifier (microwave)
        Buffer Amplifier (1-2 GHz)
        Buffer Amplifier (10-100 MHz)
        Oscillator Core
        RF Oscillator Core
        --------------------
        Parallel Resonator
        Series Resonator
        SAW Resonator
Component Characterization
        S-parameters for 1-port
        S-parameters for 2-port
        --------------------
        BJT Curve Tracer
        RF BJT Curve Tracer
        FET Curve Tracer
        S-parameters for Biased BJT
```

Menu Item Caption: Components

Quick Help: none

□ Locked

To open the menu editor, select **Tools > Menu Editor** from the main window, or push the Menu Editor button on the toolbar. The menu editor has the following capabilities:

- Create menu cascade structures for any ADS window.
- Attach actions and file objects to menu items.

- Control menu attributes such as license locking, analog/DSP mode, and quick help files.

Menu Editor Components

The Menu Editor is composed of the following components (as shown in the preceding figure).

- **Menu Controls**. Allow the creation, deletion and renaming of any number of menus for use in ADS windows.
- **Menu Item Controls.** Allow the creation, deletion and renaming of individual menu items, along with the assigning of actions, objects, and other attributes to each menu item.
- **Current Menu Layout.** Displays the current menu layout with the menu items arranged in a menu/sub-menu hierarchy.
- **Menu Editor Control Buttons.** Control buttons to save the menu information, close the editor, test the current menu structure, and open the menu editor documentation.

Creating a Menu

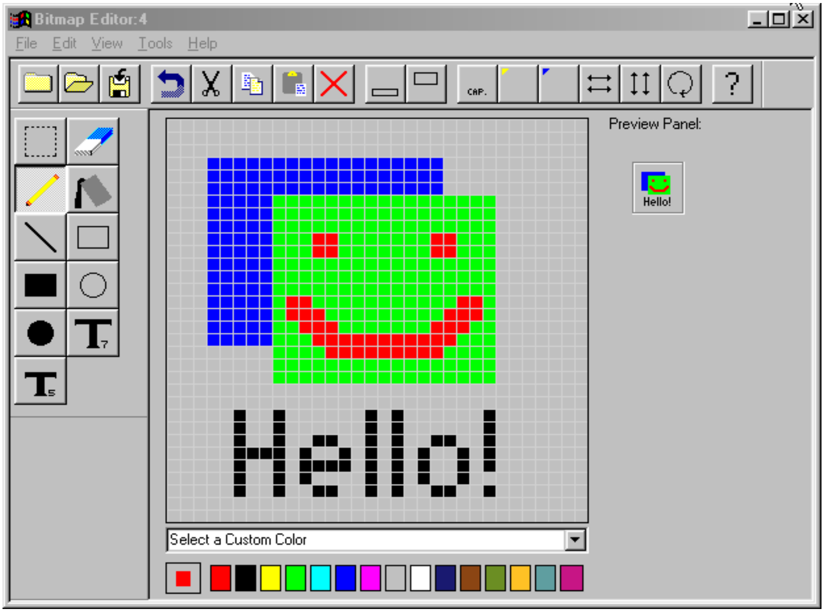
The Menu Editor has two default menus already created, one each for the Schematic and Layout windows. To create a sample menu:

1. Push the **Insert Menu Item** button on the lower right three times. This will add three new menu items to the current menu.
2. Double-click each menu item to change its name.
3. For each menu item, use the left and right arrow buttons to move it to the desired depth of indentation.
4. Select an action from the *Action* list for each menu leaf.
5. For each leaf item, select an **Object Group** from the list, if applicable.
6. Select or input an **Object** for each leaf from the list or text box.
7. Choose **OK** at the bottom of the Menu Editor.

After the studio project has been saved and built, this new menu will appear in the ADS Schematic window. Or pushing the *Test* button on the Menu Editor will create a window showing the menu and its accompanying actions and objects.

For more detailed information, refer to Chapter 3, Menu Editor.

Bitmap Editor



The Bitmap Editor is functionally independent of the rest of the DesignGuide Developer Studio and can be used without using any other Developer Studio component. When used in connection with the Developer Studio, the bitmap editor has the following capabilities:

- Save 32X32 bitmaps for use as palette buttons or elsewhere.
- Create custom bitmaps using a wide selection of drawing and editing tools in 15 colors.
- Add custom captions to new or existing bitmaps.

- Add ADS Smart Component, or Analog/RF color corners to custom or existing bitmaps.

Saving a Custom Bitmap

To save a user-created bitmap:

1. Select **File > Save As.**
2. Type in a name for the new bitmap. Bitmaps are saved in standard .bmp format, making them compatible with any other application that uses bitmaps.

Loading a Bitmap

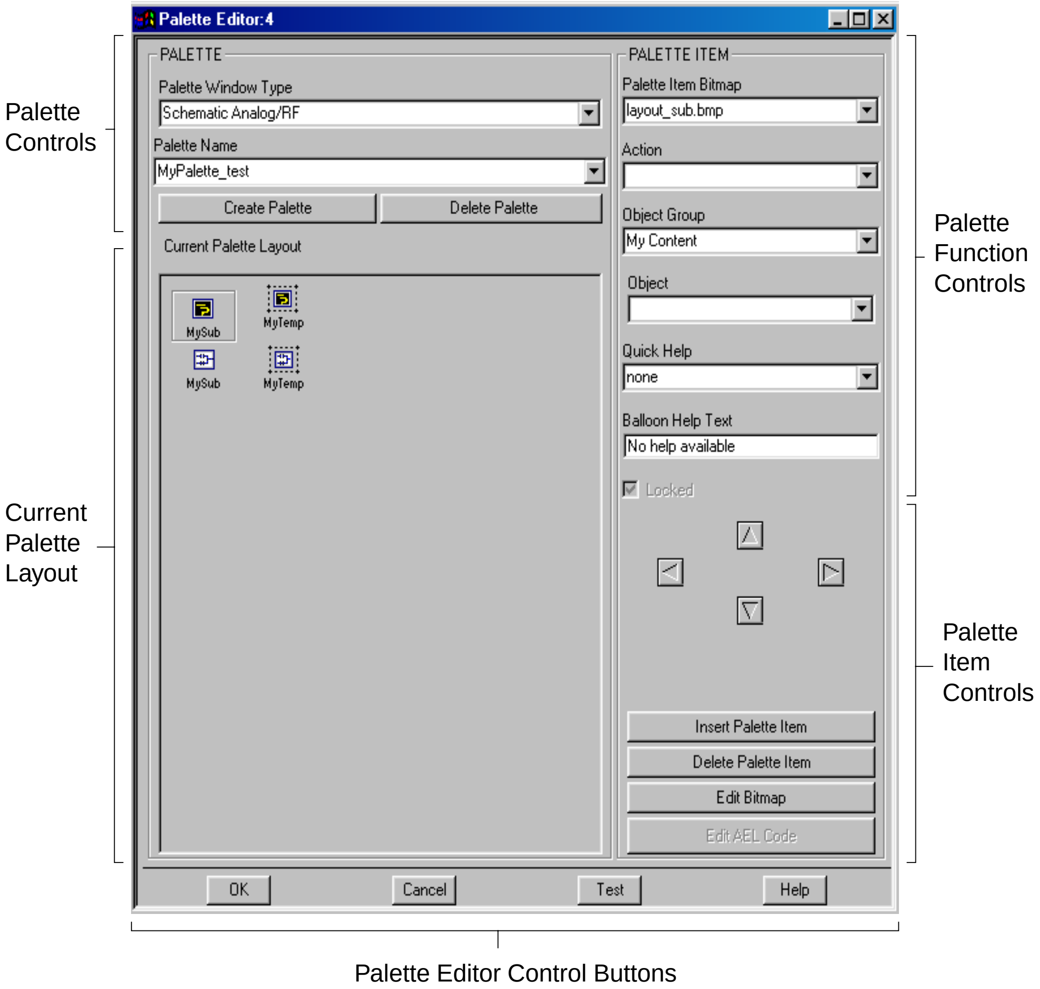
To load a bitmap:

1. Select **File > Open** or choose the **Open** toolbar button.
2. Select the file to open.

Note The DesignGuide Developer Studio Bitmap Editor only works with bitmaps that are 32X32, so larger or smaller bitmaps cannot be opened or edited.

For more detailed information about the Bitmap Editor, refer to Chapter 5, Bitmap Editor.

Palette Editor



Palette Window Type: Schematic Analog/RF

Palette Name: MyPalette_test

Palette Item Bitmap: layout_sub.bmp

Action:

Object Group: My Content

Object:

Quick Help: none

Balloon Help Text: No help available

☑ Locked

The Palette Editor has the following capabilities:

- Design any number of custom palettes for use in ADS windows.
- Add custom bitmaps to palette buttons.

- Attach actions and content objects to palette buttons, including custom AEL functions.

Palette Editor Components

The Palette Editor consists of the following components (as shown in the preceding figure).

- **Palette Controls** allows for the creation and deletion of palettes in the analog/RF, DSP, or layout window.
- **Current Palette Layout** displays the bitmap image of each palette item as it would appear in an actual palette.
- **Palette Function Controls** controls the action, objects, quick help windows, and balloon help texts of palette items.
- **Palette Item Controls** control buttons to save the palette, cancel, test the palette, and display help options.

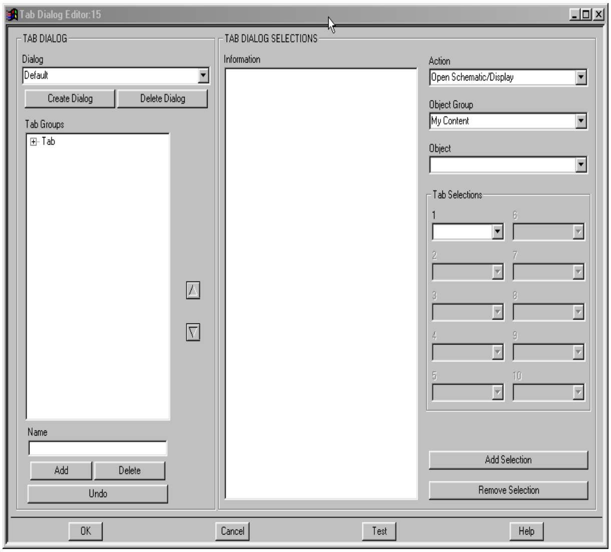
Creating a Palette

The palette editor has a default palette in the analog/RF window. To create a palette and add items:

1. Push the **Insert Palette Item** button.
2. Press any of the bitmap icons shown in the window.
3. Chose the **Insert** button followed by the **Close** button.
4. Select an action from the *Action* list.
5. Select an object from the *Object* list. (Object must be registered in the content editor/main window).
6. Select a quick help text name from the *Quickhelp* list (Quick help texts must be registered in the Quick Help editor).
7. Choose **OK** on the bottom of the editor.

After saving the project, choose the **Build** button. Open a Schematic window and select the palette you just created in the dropdown palette list. Press the palette button.

Tab Dialog Editor



Dialog: Default

Action: Open Schematic/Display

Object Group: My Content

The Tab Dialog Editor is a powerful tool that allows a number of different content files to be tied into the same menu or palette action, allowing the user to set certain parameters through a separate dialog box.

Creating a Tab Dialog

To create a Tab Dialog item:

1. Type in the desired name in the **Dialog** text area in the upper left.
2. Select **Tab** from the *Tab Groups* list.
3. Choose the **Add Tab Group** button to add the desired number of tab groups.
4. For each Tab Group, expand its button list, and select the first button.
5. Add the desired number of buttons by choosing the **Add Button Item** button.
6. Rename each Tab Group and button item using the *Name* text area at the bottom.

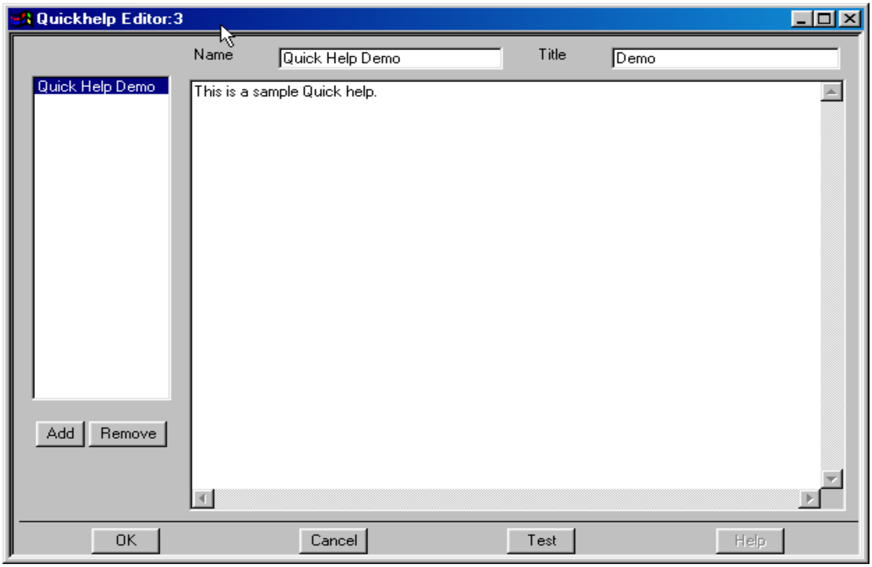
7. Add a Selection item for each content object you want controlled by the Tab Dialog item. (For more information on Selection items, refer to Chapter 6, Tab Dialog Editor.
8. Choose **OK**.

Using a Tab Dialog Item

Once created, the Tab Dialog item can be attached to a menu item by doing the following:

9. Open the Menu Editor.
10. Select the desired menu and menu item.
11. Select **Open Tab Dialog** from the *Action* list.
12. Select **Tab Dialog** from the *Object* list.
13. Choose **OK**.

Quick Help Editor



Name Quick Help Demo

Title Demo

Quick Help Demo

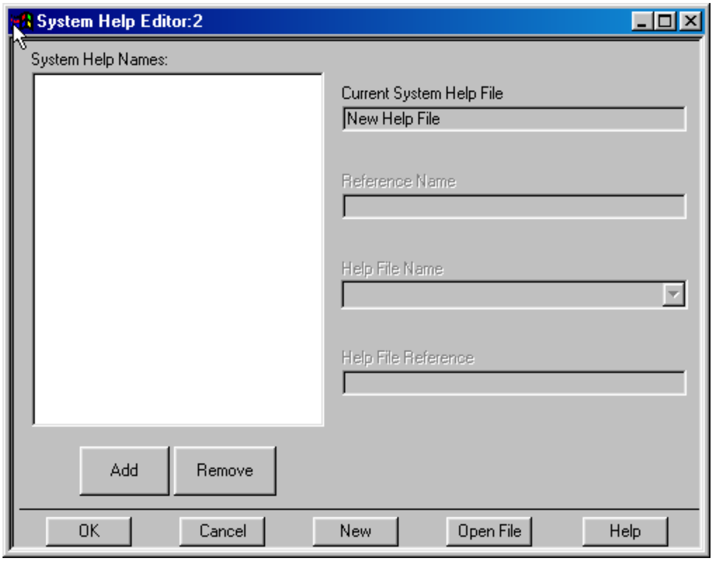
This is a sample Quick help.

The Quick Help Editor provides an easy way for developers to create a quick help file, that can be activated when a menu or palette item is selected.

To create a quick help file:

1. Choose **Add.**
2. Select the new file from the file list on the left.
3. Change the name (if desired) in the *Name* text box on top.
4. Type in a title in the *Title* text box.
5. Type in the desired text in the large text area.
6. When finished, press **OK**.

System Help Editor



Current System Help File

New Help File

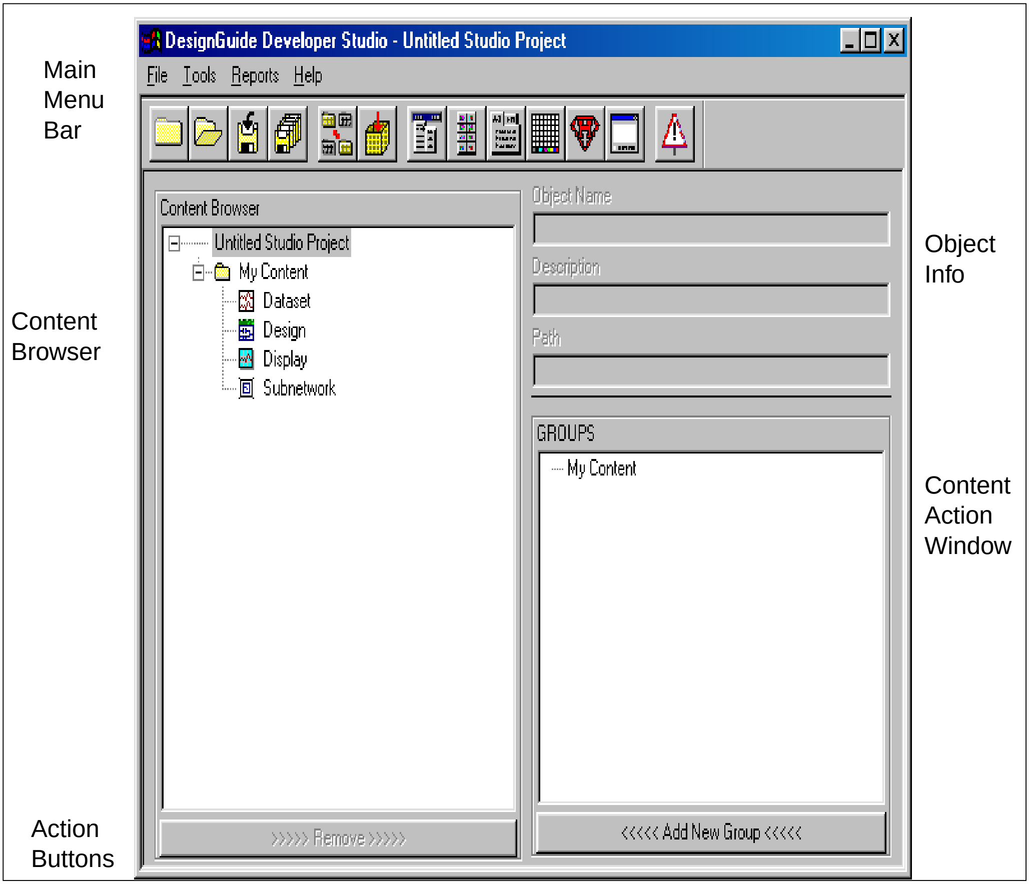
The System Help Editor allows developers to link documentation files to menu and palette items, allowing users to display documentation without much trouble. System help linking has the following process:

1. Copy all the documentation (*.html*) files into the */doc/* subdirectory of the studio project directory. Make note of all the HTML references within each file that need to be indexed.
2. Choose **Add** to create a new Help System index.
3. Put an HTML reference into the *Help File Reference* text box.
4. Select the file in which the reference is found with the *Help File Name* drop down list.
5. Change the **Reference Name** to the desired name. This is the name that will appear in the Menu and Palette Editor when help files are used.
6. Repeat Steps 2 through 5 for each HTML reference to be linked.
7. Choose **OK**. This will copy the *.html* files to the proper place in ADS and create an index file.

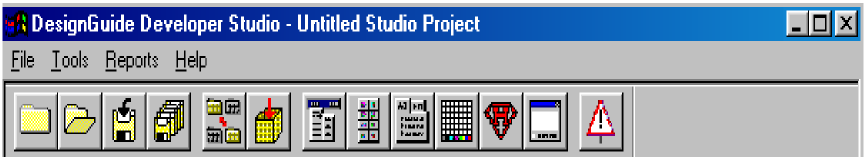
Note While HTML references do not need to be unique, *Reference Names* should be unique across all help files, as ADS will look through all help file indexes for the given reference name.

Chapter 2: Content Editor

The DesignGuide Developer Studio initially starts with the Content Editor. The Content Editor is the central organizational tool used to organize, register all the content, and construct the DesignGuide.



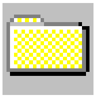
Main Menu Bar



The main menu bar is used to navigate through the Developer Studio while developing DesignGuides. Some menu items are attached to tool bar buttons for ease of use.

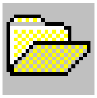
File Menu

New



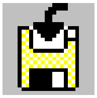
Create a new Studio Project.

Open



Open an existing Studio Project.

Save



Save the currently opened Studio Project.

Save As. Prompts for a filename before saving the active file.

Delete. Delete an existing Studio Project.

Build

Creates the necessary directory structure inside ADS to run your DesignGuide. All registered content will be copied into this structure. Additionally the necessary files will be created to run your menus and palettes or any other objects created in the DesignGuide Studio.

Unbuild. The opposite of build. It will remove all directories and files from a previous build.

Package

Creates the *xxx.deb* file for your DesignGuide, copying all the design files, datsets, data display pages, and so on, to the *studio_files/<project name>* directory. Then all files that went into the *xxx.deb* file are deleted, and this file is now the archive file to be transportable and installable in ADS.

Version

Version control is an easy way to back up your Studio Projects. Refer to the section "Version Control" on page 2-11.

Preferences. Summons a dialog box allowing you to specify certain variables used in the build and package or your DesignGuide.

Recent Files. List of the most recently used Studio Projects.

Exit. Quits the DesignGuide Developer Studio.

Tools Menu

Menu Editor

Opens the Menu Editor utility.

Palette Editor

Opens the Palette Editor utility.

Tab Dialog Editor

Opens the Tab Dialog Editor utility.

Bitmap Editor

Opens the Bitmap Editor utility.

System Help Editor

Opens the System Help Editor utility.

Quick Help Editor

Opens the Quick Help Editor utility.

Note See the appropriate chapters for more details on the editors.

Reports Menu

Problem Summary.

Displays a spreadsheet of problems within the Studio Project.

Content Summary. Displays a summary from the Content Editor.

Menu Summary. Displays a summary from the Menu Editor.

Palette Summary. Displays a summary from the Palette Editor.

Dialog Summary. Displays a summary from the Dialog Editor.

Help Menu

Getting Started. Opens the Getting Started documentation.

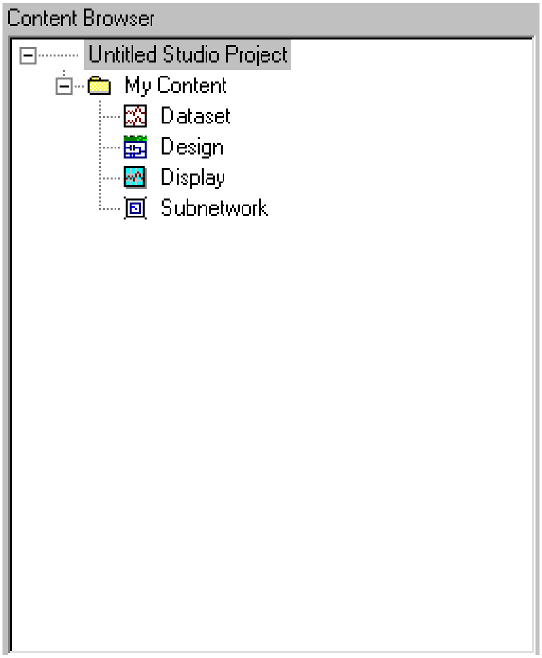
Reference Manual. Opens the Technical Reference Manual.

Style Guide. Opens the Agilent DesignGuide Style Guide.

Tutorial. Step-by-step tutorial using the DesignGuide Studio to create our example studio project.

About DesignGuide Studio. Informative synopsis of the DesignGuide Developer Studio

Content Browser



The Content Editor uses a hierarchal organization that simplifies the addition of content and makes viewing added content easy to see and understand. Content is viewed inside the Content Browser.

Content Hierarchy

Since the Content Browser uses a structural tree to organize the content the focus inside the browse window is very important. Where you are inside the window will change the result of pressing an Action Button. The hierarchal order is as follows:

Project Name

Group Name

Content Type

Content File

Dependent File

There are five different possible levels inside the Content Browser:

- **Project Name**. Name of the currently opened Studio Project.
- **Group Name**.

  Content can be divided into groups at your convenience.

- **Content Type**. There are five different content types that can be added to a Studio Project.
  - **Dataset.**

    Any file inside the */data* directory of an ADS project.

  - **Design.**

    Any .*dsn* file inside the */networks* directory on an ADS project.

  - **Display**.

    Any .*dds* file inside the root directory of an ADS project.

- **Subnetwork**.

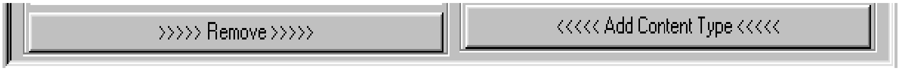
  Any *.dsn* file inside the /networks directory on an ADS project.

- **Template**.

  Any *.dsn* file inside the /networks directory on an ADS project.

- **Content File.** Added content files are listed under the Content Type of which they belong. The default object name is the file name with no extension.
- **Dependent File.** Any Content File may also take on Dependent Files. Dependent files must first be included as regular Content Files and then added as Dependent Files. All Dependent Files will be denoted by the icon relative to the Content Type they belong to.

Action Buttons



The Content Editor has only two action buttons. In essence, one is to add content while the other removes it. These buttons will be explained in greater detail further on in the manual.

- **Add.** Depending on the focus inside the Content Editor, adds content.
- **Remove**. Depending on the focus inside the Content Editor, removes content.

Adding Groups

1. Change the focus to be on the Project Name hierarchal level. The *Add* action button will change to *Add New Group*.
2. Click **Add New Group** and a new default group will appear. The focus will automatically go to the *Object Name* text box.
3. Change the group name.

If you don't change the name and you try to add another group, it will prompt you to change the first group name.

Adding Content Types

1. Change the focus to be on the *Group Name* hierarchal level. The *Add* action button will change to *Add Content Type*.
2. Highlight the **Content Type** you want to add inside the window titled *Content Types*.
3. Click **Add Content Type** and this Content Type will be added to the highlighted group.

Adding Content File

1. Change the focus to be on the *Content Name* hierarchal level below the group you want to add files. The *Add* action button will change to *Add Project File*. The window titled *Project Files* will now display all the ADS projects on your hard drive.
2. Browse to the file you want to add and click on **Add Project File** or double-click the file itself. The file will be added to the highlighted content type in that group.

Adding Dependent Files

3. Change the focus to be on the *Content File* hierarchal level, specifically on the file to which you want to add dependent files. The *Add* action button will change to *Add Dependent File*. The window titled *Dependent File List* will now display all the content yet added to your project.
4. Browse to the file you want and click *Add Dependent File* or double-click the file itself. The dependent file will be add to the *Content File*.

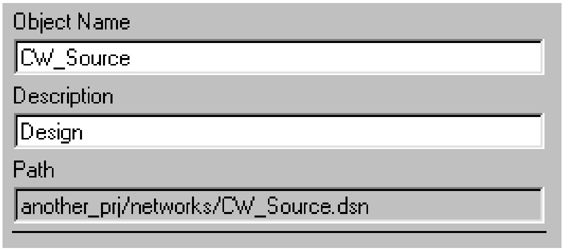
Removing Content

Removing content is a simple procedure.

1. Highlight any object beneath the *Project Name* hierarchal level inside the Content Browser.
2. You will notice that the caption on the *Remove* action button will change according to which object you want to remove. Click the **Remove** action button.
3. All objects from the selection level down the hierarchal tree will be removed.

Note Projects cannot be removed using the *Remove* action button. To remove a project, select **File** > **Delete** from the main menu bar,

Object Info



Object Name

CW_Source

Description

Design

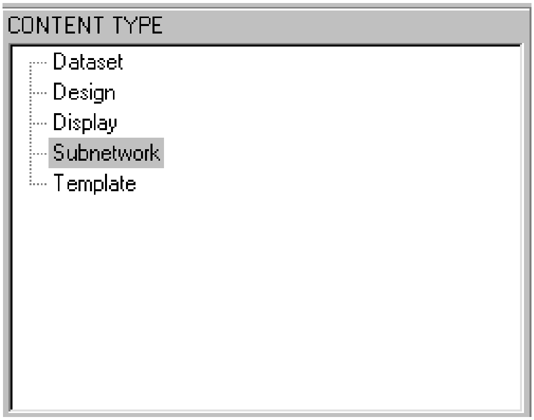
Path

another_prj/networks/CW_Source.dsn

Object Info displays information about a *Content File*. Something will be displayed only when the focus is on the *Content File* hierarchal level. Three items are shown:

- **Object Name.** All of the Developer Studio Editors will use the *Object Name* as a reference to added content. The content file minus the extension is the default *Object Name*.
- **Description**. A Description of the added content. The Developer Studio does not use the Description for anything, but can be used to help organize studio projects. By default, the description is the Content Type.
- **Path.** Starting in the $HOME$ directory for ADS shows the Path where the source file resides.

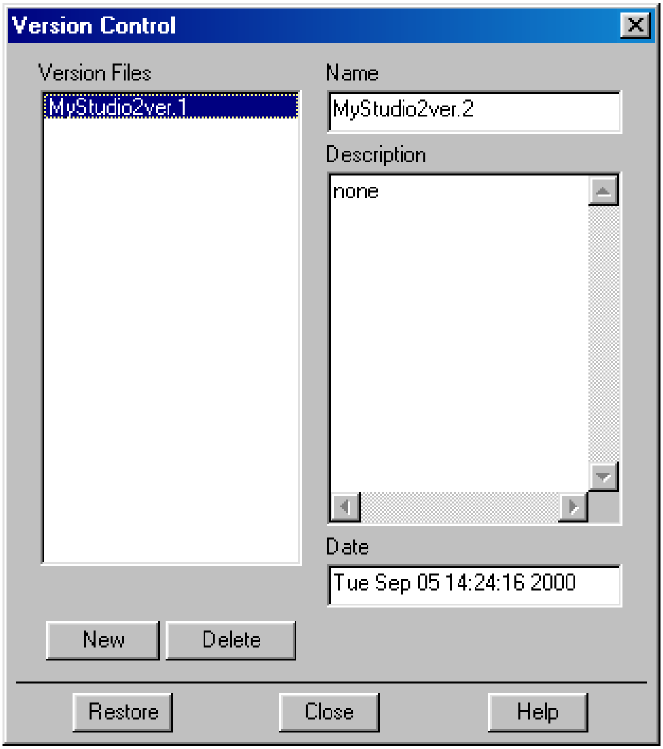
Content Action Window



The Content Action Window dynamically changes depending on the focus inside the Content Browser. There are five states the Content Action Window can be in:

- **Groups.** The Content Browser focus is on the *Project Name* level. The Content Action Window will display a list of the added Groups.
- **Content Type**. The Content Browser focus is on the *Group Name* level. The Content Action Window will display Content Types.
- **Project Files**. The Content Browser focus is on the *Content Type* level. The Content Action Window will display a directorial list of all ADS projects in the $HOME$ directory.
- **Dependent Files**. The Content Browser focus is on the *Content File* level. The Content Action Window will display a list of all content in the current studio project.
- **Nothing**. The Content Browser focus is on the *Dependent File* level. The Content Action window will display nothing.

Version Control



Version Files: MyStudio2ver.1

Name: MyStudio2ver.2

Description: none

Date: Tue Sep 05 14:24:16 2000

The Version Control creates backups of your studio project files (i.e. files with the *.stu* extension).

Creating a New Version

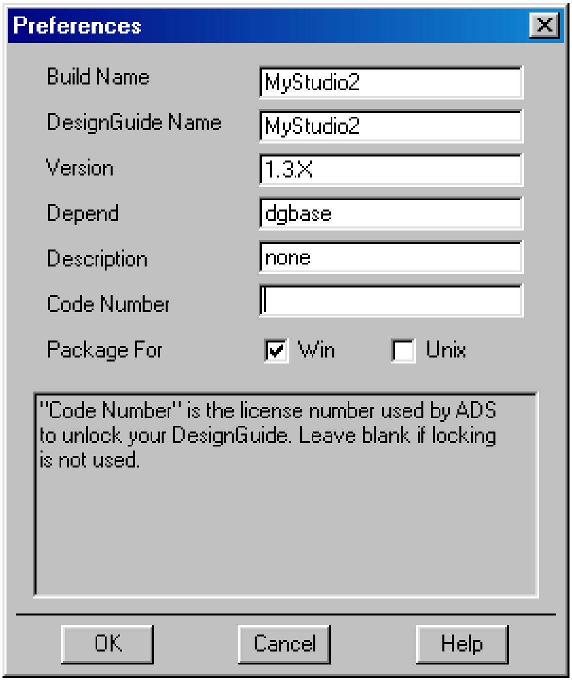
1. With your studio project open, click **File >Version.**
2. Click **New.**

A new file will appear in the *Version Files* list. Default information will appear for *Name*, *Description*, and *Date*. Any of this information may be changed to suit this version.

Restoring an Older Version

1. Have your studio project open and click **File >Version.**
2. Highlight the version you want to restore.
3. Click **Restore**.
4. The older version will be restored and loaded into the Content Editor.

Preferences


Preferences

Build Name MyStudio2
DesignGuide Name MyStudio2
Version 1.3.X
Depend dgbase
Description none
Code Number
Package For ☑ Win □ Unix

"Code Number" is the license number used by ADS to unlock your DesignGuide. Leave blank if locking is not used.

The Preferences box displays some variables that you can change. As you click in each text box, a paragraph at the bottom of the window will describe what each variable means.

Preference Variables

There are seven user-defined variables that can be changed according to each DesignGuide:

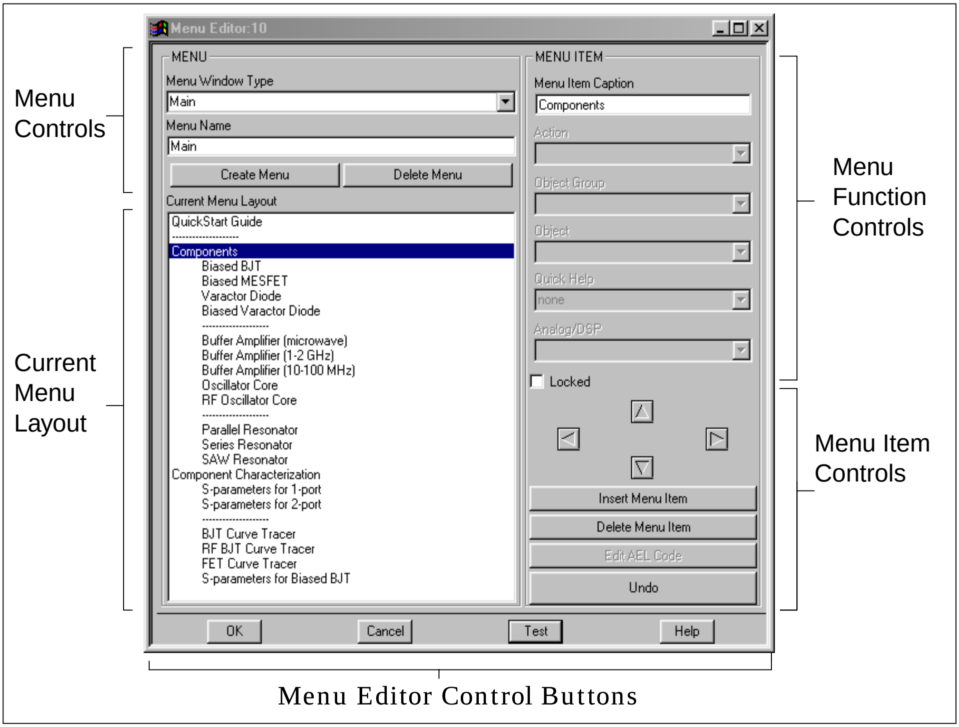
- **Build Name**. Refers to the name of the directory where your DesignGuide will be built. It is used as the project name in the *studio_file* directory and in the *$HPEESOF_DIR/DesignGuides/Projects* directory.)

  Note. The next five variables are used to make the *control* file for your package. The *control* file defines information about your DesignGuide.

- **DesignGuide Name**. Name of the Debian archive (package) file. Also used as a default name for some of the DesignGuide Developer Studio Editors.
- **Version**. DesignGuide Version. Default is 1.5.X, where X can be changed to any number.
- **Depend.** Some DesignGuides depend on other DesignGuides to function. A list of dependent DesignGuides may be placed here. Default is *dgbase*. All listed DesignGuides must be separated by a comma.
- **Description**. A short description of your DesignGuide.
- **Code Number.** Used for code locking and licensing purposes. Leave blank if no code locking is to be used.
- **Package For.** Choose either **Win** or **Unix**. The only change made to your DesignGuide is the bitmaps. If **Unix** is checked, the bitmaps will be converted to an *.xpm* format instead of a *.bmp* format

Chapter 3: Menu Editor

The DesignGuide Studio Menu Editor is a powerful tool that allows users to dynamically create menus with a large variety of actions to accompany any ADS window. Using user-created content, the menu editor can provide convenient access to a variety of tools and functions, making the design process much easier.



Menu Window Type: Main

Menu Name: Main

Current Menu Layout

- QuickStart Guide
- -------------------
- Components
  - Biased BJT
  - Biased MESFET
  - Varactor Diode
  - Biased Varactor Diode
  - -------------------
  - Buffer Amplifier (microwave)
  - Buffer Amplifier (1-2 GHz)
  - Buffer Amplifier (10-100 MHz)
  - Oscillator Core
  - RF Oscillator Core
  - -------------------
  - Parallel Resonator
  - Series Resonator
  - SAW Resonator
- Component Characterization
  - S-parameters for 1-port
  - S-parameters for 2-port
  - -------------------
  - BJT Curve Tracer
  - RF BJT Curve Tracer
  - FET Curve Tracer
  - S-parameters for Biased BJT

Menu Item Caption: Components

Quick Help: none

☐ Locked

Menu Controls

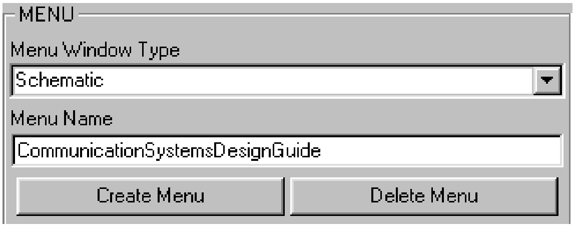
Current Menu Layout

Menu Function Controls

Menu Item Controls

Menu Editor Control Buttons

Menu Controls



Menu Window Type
Schematic

Menu Name
CommunicationSystemsDesignGuide

The menu controls are used to display the current menu name and type, as well as provide means to create and delete menus.

Menu Types

There are three main menu types within a studio project:

- **Main.** Appears in the DesignGuide menu cascade in the ADS main window.
- **Schematic**. Appears in the DesignGuide menu cascade in a project schematic window
- **Layout.** Appears in the DesignGuide menu cascade in a project layout window

Initially, a newly created studio project has a Schematic and Layout menu already created with one default menu item. These default menus (and any additional user-created menus) are automatically compiled when the studio project is built, and automatically placed in their respective windows when ADS is opened.

Creating and Deleting Menus

To delete a menu:

1. Select the menu to delete from the *Menu Window Type* drop down list.
2. Choose the *Delete Menu* button.
3. Choose **OK** in the dialog box. Once deleted, that menu will no longer appear in the DesignGuide cascade in ADS after the studio project is rebuilt.

Note If no menus are desired, all of the main menus can be deleted. In this case, all of the menu item dialog items are turned off until a new menu is created (or *Undo* is chosen).

Creating a New Menu

Choosing the *Create Menu* button will add a new menu to the studio project, according to the following conditions:

- *One or more standard menu types does not exist or has been deleted***.** The new menu is the same standard type that is missing. If more than one standard menu is missing, the first one in the list that's missing (*Main*, *Schematic*, and *Layout*) will be replaced.

  Note Replacing a previously deleted default menu by choosing the *Create Menu* button does not replace any of the menu information that the default menu possessed prior to deletion. To retain this prior information, the *Undo* button will need to be chosen to return the previous menu.

- *All three default menus are present*. A custom menu is created.

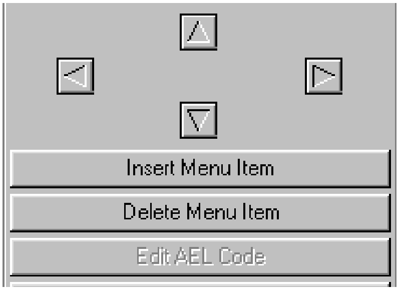
  Note At this time, the DesignGuide Studio does not have the capability to place custom menus in arbitrary locations. Therefore any user-created custom menus will not be built and used within ADS. Only the three default menus, if present, will be built and executed when ADS is run.

Menu Names

The Menu Name text area contains the name of the selected menu, which defaults to the studio name chosen in the Content Editor. This is the name that will appear in the DesignGuide cascade in each ADS window. To change the name, type in a new name and the menu editor will store it with the other menu information.

Menu Item Controls

The Menu Item Controls, as shown here, allow you to control the number, name, and format of menu items within a menu.



Adding Menu Items

Each default menu starts with just one menu item. To add additional menu items:

1. Select the menu item that's immediately above and *at the same indentation level* of the desired location of the new menu item.
2. Choose the **Insert Menu Item** button. The menu item will be added below the selection location, at the same depth level. If no menu item is selected, the new item is added at the bottom.

Note For convenience, the new menu items are automatically configured to the action and the group of the currently selected menu item. This makes it easier to create several menu items in a row that perform the same action.

Deleting Menu Items

To deleting a menu item:

1. Select the unwanted menu item.
2. Choose the **Delete Menu Item** button.

In the case where the deleted menu item had sub-menus attached to it, all the sub-menu items are automatically shifted over to become sub-menus to the menu item directly above the deleted one. In a case where a menu item was deleted mistakenly, the *Undo* button can be pushed to undo the delete.

Note Menu items that are the only one remaining in a menu cannot be deleted using the *Delete Menu Item* button. To remove the menu, the *Delete Menu* button must be used.

Changing Names of Menu Items

There are two ways to change the name of a menu item:

- Type in the new name in the *Menu Item Caption* text box in the upper right of the Menu Editor.
- Double-click on a menu item and type in the new name in the dialog box.

Note Menu names can consist of any combination of characters and spaces, although it must not be empty. Leading and trailing white space will be removed once the new name has been entered. Menu names need not be unique within a menu, although to avoid confusion when the menu is used, unique names are preferable.

Menu Separators

Menu separators, while not a vital component to a menu, can be added to allow users to visually differentiate between different sections of a menu. There are two methods to add a separator:

- Type in a '-' character into the *Caption* text box.
- Double-click on a menu item and enter a '-' character into the dialog box.

The separator will appear as '------------------' in the Layout window. Menu separators cannot have actions or objects associated with them. Nor can they have sub-menus below them.

Creating Sub-Menus

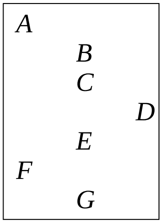
Creating sub-menus is an important part of menu creation. Using the arrow buttons on the right side of the Menu Editor, menu items can be moved left or right. Following are some important points about creating a menu hierarchy:

- Concurrent menu items at the same level will be in the same cascade menu.
- A menu item one level to the right of the one immediately above it is part of a sub-menu; the menu items that will appear when the parent menu item is selected.
- Parent menu items cannot have actions or objects associated with them. If an action or object has been assigned to a menu item previously, and the menu item later becomes the parent of a sub-menu, the action and object will be cleared.
- Menu separators can also be moved left and right, with the restriction that separators can never have sub-menus on their own. If a menu item immediately below a separator is moved to the right, an error message will occur. Likewise, a separator moving to the left with menu items directly below it will bring the same error message.
- Menu items within a sub-menu can have sub-menus of their own. There is no limit to the number of sub-levels within a menu, although having more than three levels of menu items can be difficult and inconvenient to you.

Moving Items Up and Down

The up and down arrows on the right side of the Menu Editor allow menu items to be moved up and down among menu items of the same depth. For instance, given the menu in the example shown here, moving G up would exchange places with E, rather than F, as G and E are at the same depth. By the same token, moving A down would switch places with F, since they are at the same level, while D would be unable to move up or down, without first moving to the left

.

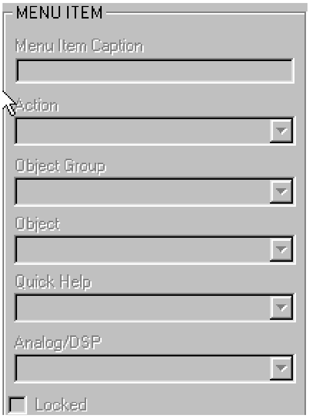


Note As before, menu separators cannot have sub-menus associated with them, so while separators can be moved up and down as with any menu item, moves that result in separators having sub-menus will not be allowed by the Menu Editor.

Undo

The Menu Editor keeps a backup of the last five changes made to the current menu. The Undo button can be used up to five times in succession to return the current menu to its previous state. These changes include the creation and deletion of menus, as well as any changes in the number or format of menu items.

Note The backup menus are reset any time a new menu is selected by choosing the *Menu Window Type l*ist. This is to prevent any accidental changes to a previously selected menu while another menu is on the screen. If changes need to be undone that are more than five steps away, or before the current menu was selected, the Content Editor's version control system will need to be used to bring back a previous version of the studio project.



Menu Function Controls

The Menu Function Controls allow you to assign *actions* and *objects* to menu items, as well as control other areas such as quick help and user accessibility.

Actions and Objects

Menu items would be fairly useless without accompanying actions. An *Action* is the event performed by ADS when the menu item is selected, which can be displaying a specific design, opening a smart component, or running a block of AEL code. The following actions are supported by the Menu Editor:

- **Open (Schematic/Layout)/Display.** Opens both a *.dsn* file and the accompanying display file (*.dds*) for the given object name to the ADS Schematic or Layout window.
- **Open (Schematic/Layout).** Opens a *.dsn* file to the ADS Schematic or Layout window.
- **Open Display.** Opens a *.dds* display file.
- **Open Tab Dialog.** Opens a Tab Dialog object, created using the Tab Dialog Editor.
- **Open Help File.** Opens either a quick help window, or an HTML documentation file.
- **Place Subnetwork**. Opens a Subnetwork *.dsn* file for immediate placement on the ADS Schematic or Layout window.
- **Place Template.** Opens a Template *.dsn* file for immediate placement on the ADS Schematic or Layout window.
- **Place Component.** Opens a user-defined component for placement on the ADS Schematic or Layout window.
- **Show Palette.** Displays a user-created palette, created using the Palette Editor.
- **Execute AEL Function.** Executes a user-defined block of AEL code**.**
- **Toggle Quick Help.** Toggles all the quick help files in the menu on and off.

All eleven actions can be performed within a Schematic or Layout window, although the Main window menu is limited to opening or toggling help files, or executing AEL functions.

To assign an action:

1. Highlight the desired menu item.
2. Open the **Action** drop-down list and choose an action.

Selecting an Object

To assign an object to a menu item:

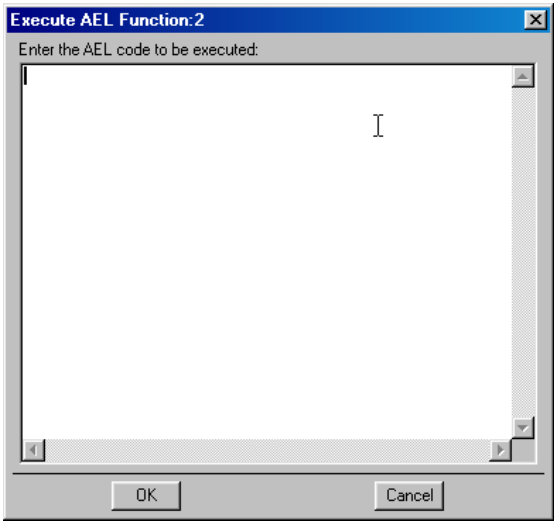
1. Assign an action to the menu item.
2. Choose an **Object Group** from the *Group* drop-down list (not necessary for some actions).
3. Open the **Object** drop-down list box. The list of object corresponding to the selected action and group will be displayed.
4. Select an object from the list.

Certain actions have different methods of obtaining objects, and do not require a group to be selected:

- **Place Component.** When this action is selected, the Object list becomes a text area where you can enter in the component name, without having to choose a group.
- **Execute AEL Function.** When this action is selected, a dialog box opens to allow you to type in a block of AEL code to be executed when the menu item is activated.
- **Open Help File.** Instead of choosing a group, the Group list contains two options for help files, Quick Help, or Help System. Both options get the list of possible objects from the Quick Help Editor and Help System Editor respectively.
- **Toggle Quick Help**. This action requires no group or object, thus the lists will be deactivated when this action is selected.

Editing AEL Code

When the *Execute AEL Function* action is selected and AEL code is inserted as the *object*, the code can then be edited by choosing the *Edit AEL Code* button. This brings up the AEL code dialog box again, allowing you to change the AEL code.



Note The AEL dialog box checks to see that the AEL code put in by you is syntactically correct. However, it is your responsibility to make sure that function and variable names are spelled correctly and that AEL code performs the desired function. Any errors in code may not be obvious until after the menu is built and executed, in which case the menu will have to be re-edited and rebuilt.

Other Menu Attributes

Following are descriptions of some additional menu attributes.

Quick Help

In addition to assigning quick help files as objects, additional quick help files can be assigned to menu items to be displayed along with the results of other actions. To assign a quick help file to a menu item, select a file name from the Quick Help file list, which will show all the quick files currently in the studio project. When assigned, the quick help file will appear on the screen when the menu item is activated along with the normal results of the menu item action.

Analog/DSP

This element of the Menu Editor only applies to Schematic menus. The Menu Editor has the capability of controlling the sensitivity of menu items based on the mode of the Schematic window. If the Analog/DSP list item is set to both (the default value), then that menu item will be sensitive, regardless of what mode the Schematic window is in. Selecting Analog RF, or DSP, however, designates that only in that specific Schematic window mode will the menu item be sensitive. The Analog/DSP settings have no effect in Main, Layout, or Custom windows.

Locking

The Locking check box allows the action and object associated with this menu item to be *locked*, or unable to be activated unless you have the appropriate license number. When checked, the menu item will call a special *check_License* function to check whether the code number, shown in the Content Editor, is an appropriate license. If the license test does not succeed, the action is not performed.

Menu Editor Control Buttons

- **OK.** Saves the current menu in the current studio project and exits the Menu Editor.
- **Cancel.** Exits the Menu Editor without saving the current menu
- **Test**. Opens a test window. See section below on *Menu Testing*.
- **Help**. Opens this help documentation.

Miscellaneous Information

Following are details on miscellaneous features.

Menu Testing

The Menu Editor provides a testing function to examine the menu structure and the attached actions and objects without having to build the entire studio project. At any time in the design process, the Test button at the bottom of the Menu Editor can be pushed to bring up a test window. The test window contains a text area and a menu bar. The menu bar contains all the currently created menus with the submenus properly structured. When chosen each test menu item prints to the text area a list of the associated action, group, and object, along with quick menu, analog, and lock information. This feature provides an easy way to look over the designed menu and make some immediate changes without having to build the entire project to look at the menus within ADS.

Menu Summary

A complete and printable summary of all the menus and menu items, along with their associated action, objects, and other attributes can be created by calling the menu summary from the Content Editor menu bar. This will display a summary, in spreadsheet format, of all the relevant information associated with that studio project's menus.

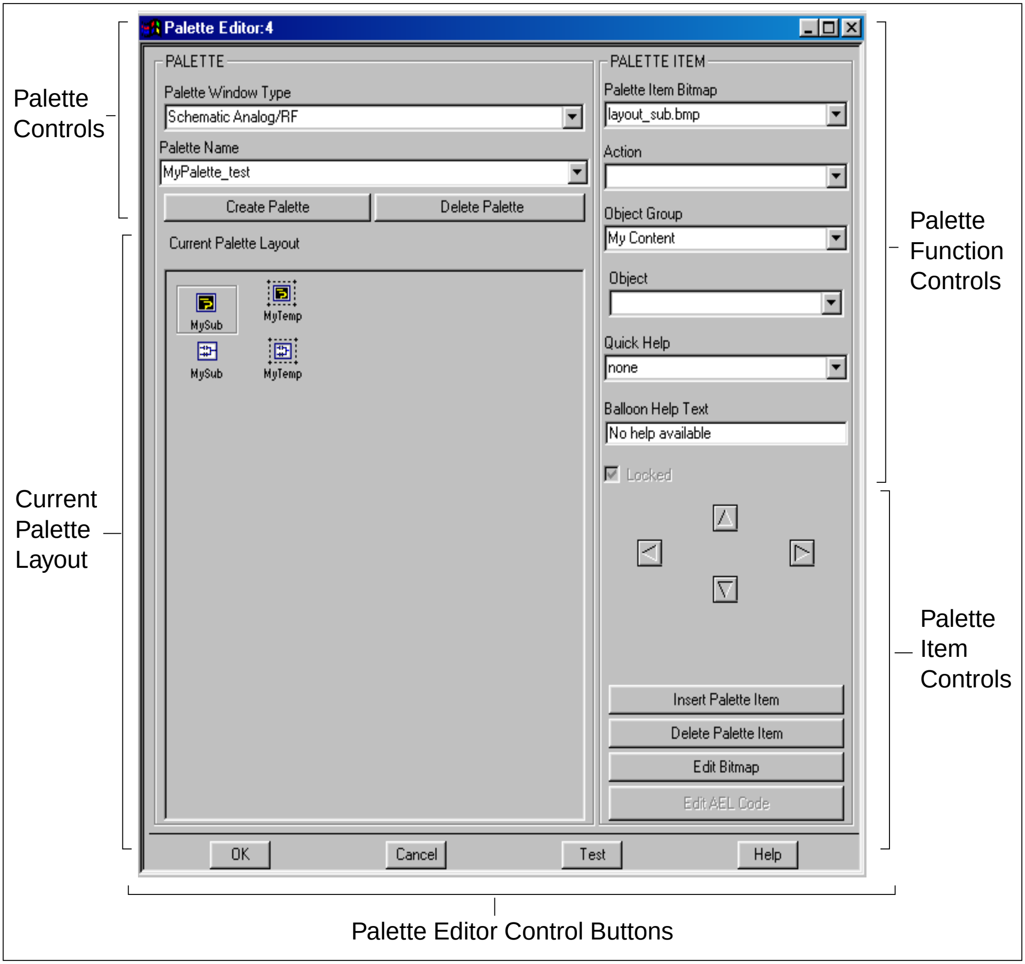
Problem Summary

As part of the problem summary, the menu editor provides a list of menu items (without submenus) that have no action or object associated with them. This is designed to help developers quickly recognize which parts of the menu are incomplete, or need fixing without having to go through each menu item individually. The Problem Summary is called through the Content Editor’s menu bar.

Chapter 4: Palette Editor

The DesignGuide Studio Palette Editor is a tool that enables you to dynamically create palettes with many different actions. Content that you've created can now be easily placed as actions to any palette item and any palette

.



Palette Editor:4

PALETTE

Palette Window Type: Schematic Analog/RF

Palette Name: MyPalette_test

Current Palette Layout

PALETTE ITEM

Palette Item Bitmap: layout_sub.bmp

Action:

Object Group: My Content

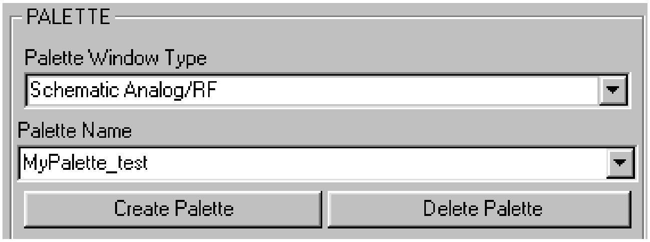
Object:

Quick Help: none

Balloon Help Text: No help available

☑ Locked

Palette Controls



PALETTE

Palette Window Type: Schematic Analog/RF

Palette Name: MyPalette_test

The palette controls are used to create, delete, or load palettes in the three window types.

Palette Types

There are three palette types:

- Schematic Analog/RF
- Schematic DSP
- Layout

All studio projects start with a default palette, *MyPalette_StudioName*, where StudioName is the name of the current studio project. The palette type is *Analog/RF*.

Note There is no default palette item, only a palette.

Creating, Deleting, Changing Palettes

The following sections provide details on creating and modifying palettes.

Creating Palettes

Palettes can be created by clicking on *Create Palette*. This will bring up a dialog box asking for the palette name. After pressing **ENTER** or clicking **OK,** a new palette will be created with no palette items.

Note Duplicate palette names are not permitted. However, duplicate palette names in different window types (Analog/RF, DSP, Layout) are permitted. For example, a palette named *MyPalette* with an Analog/RF window type and a palette named *MyPalette* with a DSP window type may both exist. A palette named *MyPalette* with an Analog/RF window type and another palette named *MyPalette* with an Analog/RF window type may not exist. Blank palette names are not permitted.

Deleting Palettes

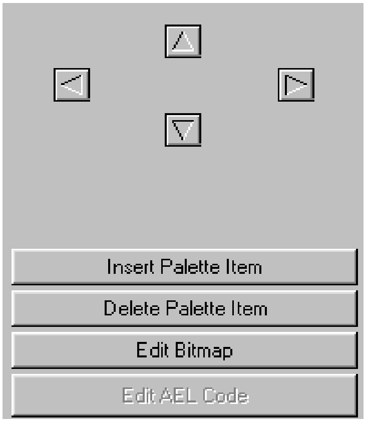
A palette may be deleted by clicking *Delete Palette*. A dialog box will ask if you are sure you want to delete the palette. Once a palette has been deleted, all information is lost. After all palettes in a window type are deleted, *Delete Palette* will be grayed out and the only option will be to select another window type or create a palette.

Change Palette Names

A palette name may be changed by clicking *Change Palette Name*. A dialog box will prompt you to enter the new name. Type the name and either press **ENTER** or click **OK**. The palette name will change to the new name.

Note Duplicate and blank names are not permitted. Refer to the "Creating Palettes" on page 4-2 section.

Palette Item Controls



The palette item controls allow you to control the number and order of palette items as well as edit palette item bitmaps.

Inserting, Deleting, Editing Palette Items

The following sections provide information on making changes in palette items.

Inserting Palette Items

To insert a palette item click *Insert Palette Item*. This will bring up the bitmap selection window. After selecting the bitmap to use, the bitmap will appear in the *Current Palette Layout* window. The palette item bitmap name will appear in the *Palette Item Bitmap* drop-down combo box. The default selection choices include My Content for the *Object Group*, none for the *Quick Help*, and *No help available* for the *Balloon Help Text*. As the number of palette items grows a scroll bar will appear enabling you to scroll and view all palette items.

Deleting Palette Items

To delete a palette item click the palette item. A grey box outlines the current palette item selected. Click on *Delete Palette Item*. A dialog box will ask if you want to continue. To delete the palette click *Yes*, or else click *No*. If the palette item is deleted, the corresponding bitmap will be also be deleted.

Note It is strongly recommended that the same bitmap is not used for multiple palette items, since deleting one of the palette items will delete the bitmap both palette items were sharing. Also, be aware that once a palette item is deleted, the information is lost. When you want to delete multiple palette items, start with the palette items on the bottom. On large palettes deleting palette items near the top of the list results in an increase of time needed to update the *Current Palette Layout* window.

Changing Bitmaps

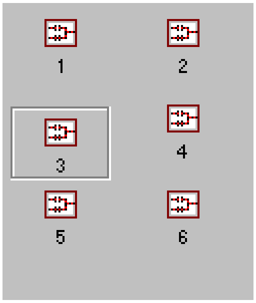
A palette item bitmap can be changed be using the drop-down combo list labeled *Palette Item Bitmap*. The list shows all bitmaps found in the bitmap directory. To change the bitmap, move the pointer to the bitmap name that you want to change the palette item to and left-click on the name. The bitmap will change in the *Current Palette Layout* window to reflect the new palette item bitmap.

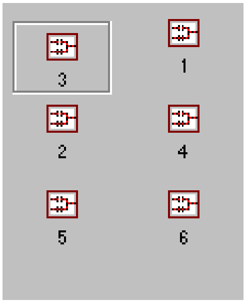
Note Although it is possible to have two palette items share the same bitmap, it is not recommended. (Refer to the "Deleting Palette Items" on page 4-4 section.)

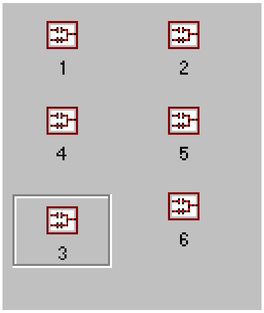
Moving a Palette Item

A palette item can be moved using the four directional arrows located below the lock checkbox. The palette item list is in a linear order starting at the top left-hand item and going across the row, then down the rows. Moving an item up or down will move the item up or down one row, and will cause the two items in the middle to slide down. Moving an item right or left will swap the position of the two items. Look at the following example. From left to right, we see the original palette, the palette moved up, and the palette moved down

.

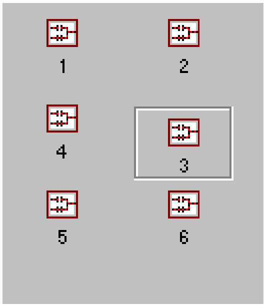


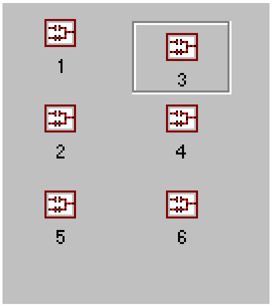




In the next example, we first see the palette moved right, then the palette moved left

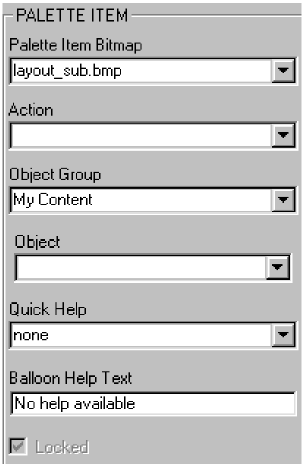
.





Note All screen shots were taken after moving the palette item from the original position.

Palette Function Controls



PALETTE ITEM

Palette Item Bitmap: layout_sub.bmp

Action:

Object Group: My Content

Object:

Quick Help: none

Balloon Help Text: No help available

☑ Locked

The palette function controls allow you to change the bitmap, actions, objects, quickhelps, and balloon text help for any palette item.

Assigning Actions, Objects, and Object Groups

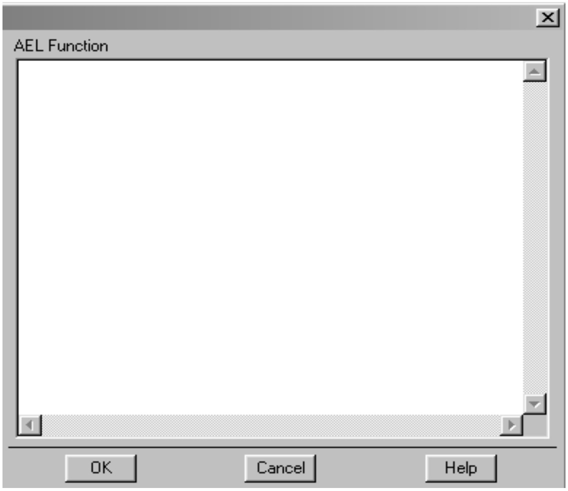
The following sections provide details on making assignments.

Assigning an Action

For the palette item to do something an action, object group, and object need to be assigned. To assign an action click on the palette item. Click on the drop-down list labeled *Action* and select the action you want the palette item to perform. Once an action has been assigned, any palette item without an action that has been selected will automatically have the same action as the first palette item with the exception of *Execute AEL Function*.

For example, Five palette items exist: A, B, C, D, and E. Suppose A has an action already assigned, *Place Template*. All other palette items have no action assigned. You select palette item B and assign it the action *Place Subnetwork*. Next you click on palette item C. Since palette item C had no previous action, the action *Place Subnetwork* gets assigned. Next you select palette item A. Since palette item A has an action already assigned nothing is changed. Palette item D is selected. Its action changes to *Place Subnetwork*. You want palette item D to execute an AEL function so you change the action to *Execute AEL Function*. Finally you click on palette item E. *Execute AEL Function* will not be the selected action. Instead it will be the previous action, or *Place Subnetwork*.

Execute AEL Function



The *Execute AEL Function* is different from other actions and needs more explanation. If *Execute AEL Function* is selected, a dialog box will appear with an editable text field. Type in the AEL code you wish to execute and click **OK**.

Note The button labeled *Edit AEL Code* will only be active if the current palette item selected has the action *Execute AEL Function*. Clicking this button will bring up the *Execute AEL Function* dialog window with the AEL code to edit. The object of an *Execute AEL Function* action is your defined AEL code.

Assigning an Object Group

When a palette item is inserted, the default object group assigned is *My Content*, or the first Object Group defined in the Content Editor. To change the object group. use the drop-down list labeled *Object Group*. Left-click the name of the new object group you wish to use.

Assigning an Object

To assign an object, both an action and an object group need to be assigned. Click the drop-down list labeled *Object* and left-click the desired object. The only exception is the object associated with the *Place Component* action. When the action assigned is *Place Component,* an edit line is used to show the object, not a drop-down list. Type in the component you wish to place, without the extension.

Assigning the QuickHelp

The Quickhelp is assigned for palette items using the drop-down list labeled *Quick Help*. Select the quickhelp name you wish to call when the palette button is pushed.

Note To select a QuickHelp topic, you must first use the Quickhelp Editor to register Quickhelp names with corresponding Quickhelp dialog boxes.

Assigning Balloon Help Text

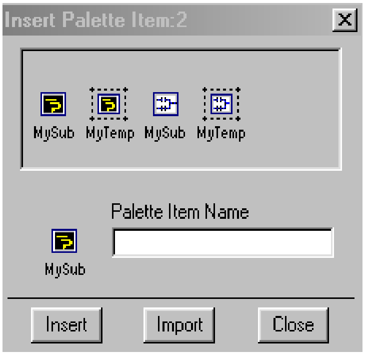
The balloon help is the text that pops up in a little balloon when the pointer is left on a palette item for a few seconds. To assign a balloon help to a palette item, remove the current text in the edit field labeled *Balloon Help Text* and insert the new balloon help text.

Lock Feature

To lock the palette item, check the box labeled *Locked*. To unlock, uncheck the box. The lock feature will check to see if a valid license exists before the action is performed. If the palette item is locked and a valid license exists, the button will work. If a valid license does not exist, the button will not work. If the palette item is not locked, the button will work regardless of the license status.

Note The license code number is found in the *File* > *Preferences* menu item.

Bitmap Selection Window



Note The Changing Palette Item Name capability is on Windows version of DesignGuide Developer Studio only.

The bitmap selection window provides a way to quickly browse the contents of the bitmap directory to find a suitable bitmap. Once a bitmap is found there are two ways to insert it in the palette. The first is as a template and the second is a copy.

Note When the *Bitmap Selection Window* first opens, it searches the *$HOME\studio_files\ProjectName\bitmaps\palette* directory, where ProjectName is the current project, and removes all bitmaps that are not 32 x 32 with 16 colors. It places the bitmaps in the *$HOME\studio_files\ProjectName\bitmaps* directory

---

Template vs. Copy

The main difference between a template and a copy is that a copy is an exact copy of the bitmap, while the template inserts a user-defined text as a caption. In both a copy and a template. a copy of the bitmap is created so even though the bitmaps look the same, they are different bitmaps. This can be seen in the palette item bitmap.

Inserting a Template

Any bitmap image may be inserted as a template. To insert a bitmap:

1. Left-click on a bitmap.
2. Type in the template text in the *Palette Item Name* edit box.
3. Left-click the *Insert* button.
4. A copy of the bitmap with a new caption is created and inserted as a palette item.

Note By default, when a project is first created, only the bitmaps in the template directory will show in the bitmap selection window. To add custom bitmaps that get added to all studio projects place them in the *$HPEESOF_DIR\designguides\projects\dgDeveloperStudio\ui\bitmaps\templates* directory. To add custom bitmaps that only get added in the current studio project add the bitmaps to the *$HOME\studio_files\ProjectName\bitmaps\palette* directory, where the ProjectName is the current project. Keep template names to about six characters since any more will not fit on the bitmap.

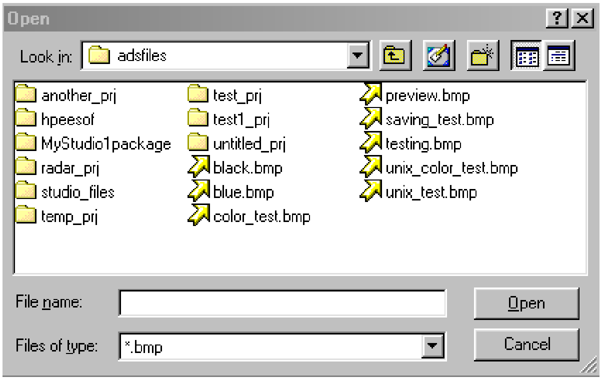
Inserting a Copy

Any bitmap may be inserted as a copy. To insert a bitmap as a copy:

1. Left-click on a bitmap.
2. Left-click the **Insert** button.
3. This will create an exact copy of the bitmap.

Note The maximum number of copies of bitmaps that can exist is 99. This is true for both template and copies, however inserting as a template will rename the bitmap to the template name. Also, the first palette item inserted as a copy will not be a copy, instead it will be the original bitmap. No copy will be made.

Import Button



Look in: adsfiles

File name:

Files of type: *.bmp

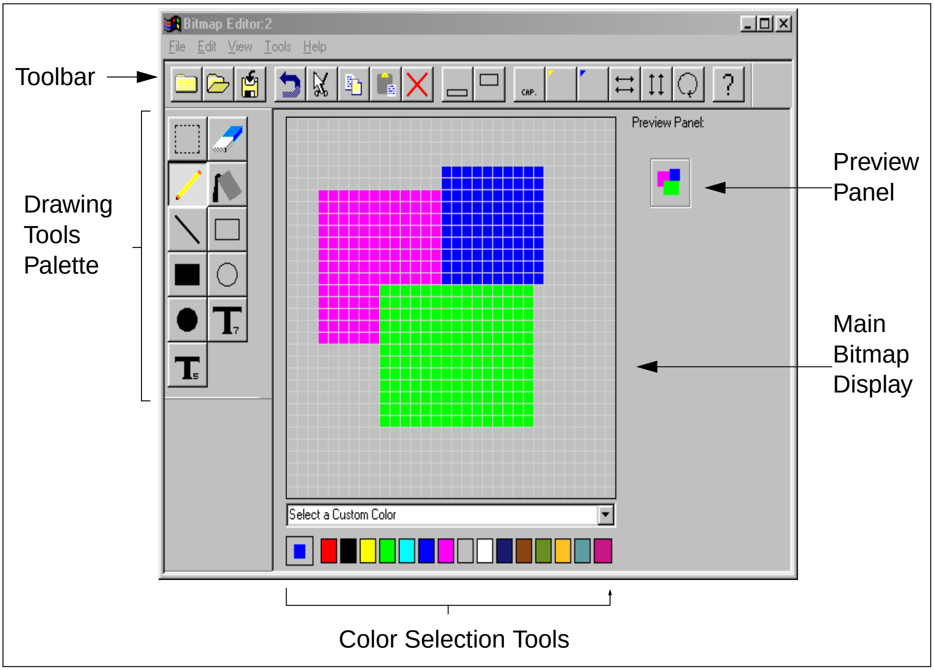
Note--Importing bitmaps only works on the PC version of DesignGuide Developer Studio. In UNIX, bitmaps must be copied by hand into the $HOME/studio_files/<current project>/bitmaps/palette directory.

The import button opens a dialog box that can be used to search for a bitmap to insert. When *Open* is chosen, the bitmap is copied into the bitmap directory. Since only bitmaps from the bitmap directory can be added as a palette item, it is necessary to make a copy. The import window will remember which directory you last imported a bitmap and will open to that directory. If you haven't imported a bitmap yet, the default directory is the bitmap directory for the current studio project.

Note Only bitmaps that are 32 x 32 with 16 colors may be imported.

Chapter 5: Bitmap Editor

The Bitmap Editor is a fully functional tool for creating and editing bitmaps, which can be used independently of the other DesignGuide Studio tools. The Bitmap Editor can create or edit any 32 by 32 bitmap for use as a custom palette button within the DesignGuide Studio itself, or for other applications that use bitmaps. The Bitmap Editor uses the standard .bmp format, so bitmaps created using the editor are compatible with any other application that uses bitmaps.



Drawing Tools Palette

Following are the buttons in the Drawing Tools Palette.

Selection

The Selection mode allows a rectangular section of the bitmap to be selected using a simple mouse drag. After selection is completed, the selection is shown on the screen with a thin dotted black line. The selection will only be in effect while the bitmap editor is in Selection mode. A switch to any other mode will remove the selection marker from the screen. Many of the toolbar commands such as cut, copy, rotate, and flip require the bitmap editor to be in selection mode, and will only function after a selection has been made.

Moving a selection

The Bitmap Editor provides a quick method of moving a selection from one place to another without having to use the Cut and Paste buttons. Moving can be done in the following steps

1. Draw a selection box around the bitmap section to be moved.
2. Put the mouse cursor within the selection box, and hold down the left mouse button.
3. An image will appear below the mouse cursor showing the current location of the selection. Move the cursor and the image to the desired destination.
4. Release the mouse button; the selection will be automatically cut from the old location and placed in the selected destination.

Erase

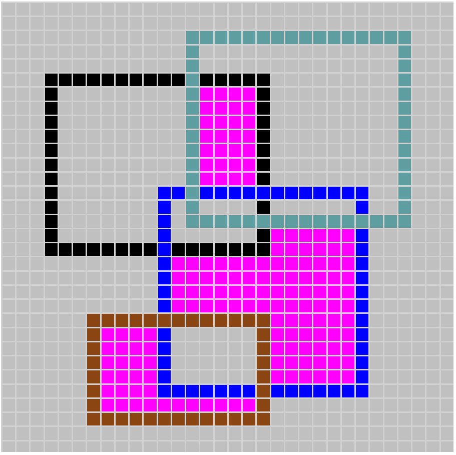
The Erase mode allows individual pixels on the bitmap to be replaced with the background color. Similar to Draw mode, Erase mode allows you to either click on individual pixels, or hold down the mouse button and drag, changing the individual pixels to the background color.

Draw

In Draw mode, you can click or drag the mouse on the screen to change individual pixels to the selected foreground color.

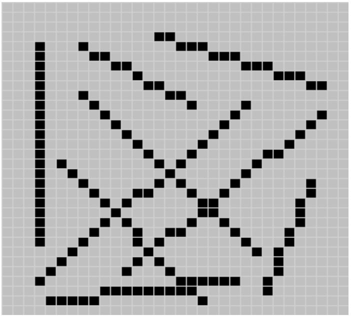
Flood Fill

In Flood Fill mode, clicked on a pixel with the mouse, causes every square of that color within the same shape to be filled with the selected foreground color. The flood proceeds in all four cardinal directions, and stops whenever it reaches a color that is different from the original selected color. The following figure shows a demonstration of the flood fill action



Line

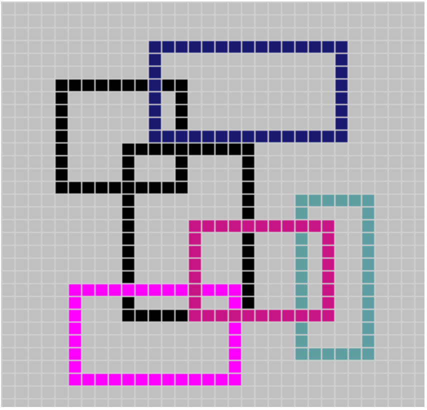
Line mode allows you to draw a line, one pixel wide, connecting any two points. Simply hold down the mouse button over the desired starting point, and drag the mouse to the desired end point. A line image will appear after dragging, showing the location of the line. When the line is in the right place, release the mouse button, and the line will be filled in with the selected foreground color.The following figure shows how the line tool works.



Rectangle

Rectangle mode allow you to draw entire rectangles quickly and conveniently, without having to draw each side separately using the line tool. After selecting Rectangle mode, move the mouse cursor to any corner of the desired location and hold down the left mouse button. Dragging the mouse will display a rectangle image which then can be sized appropriately.

Releasing the mouse button will draw a rectangular border in the selected foreground color, as shown here.

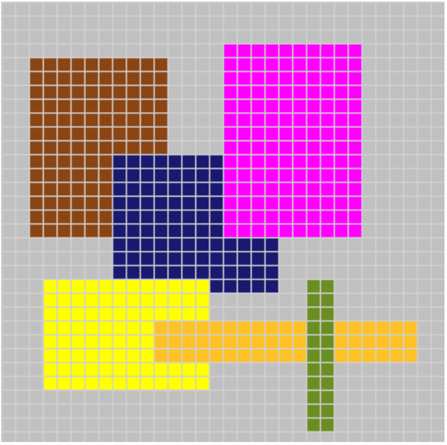


Filled Rectangle

Filled Rectangle mode works the same way as Rectangle mode, only the entire rectangle will be filled with the selected foreground color, overwriting any other colors within the boundaries of the rectangle.

The following figure shows how the filled rectangle mode works

.

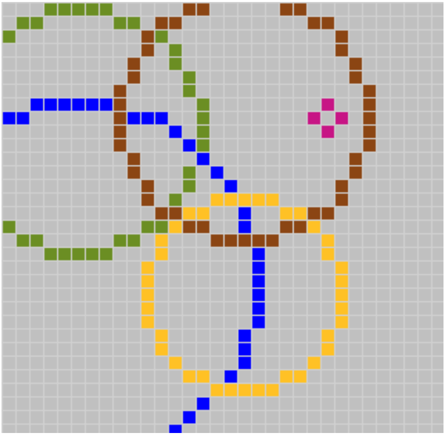


Circle

The Circle mode allows a user to add an unfilled circle to the bitmap. After selecting Circle mode, click and hold the left mouse button where the center of the circle will be, and then move the mouse outward. An image will appear, showing the boundaries of the circle as the cursor is moved. Once the circle is the proper size and shape, releasing the mouse button creates the circle using the selected foreground color.

The following diagrams show the use of the Circle too:

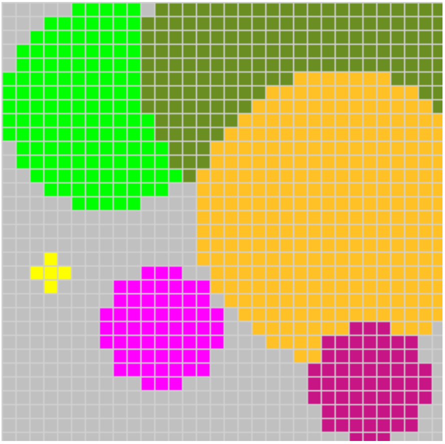
:



Filled Circle

● Filled Circle mode works the same way as Circle mode, only the completed circle is filled completely with the selected foreground color.

The following figure demonstrates its use



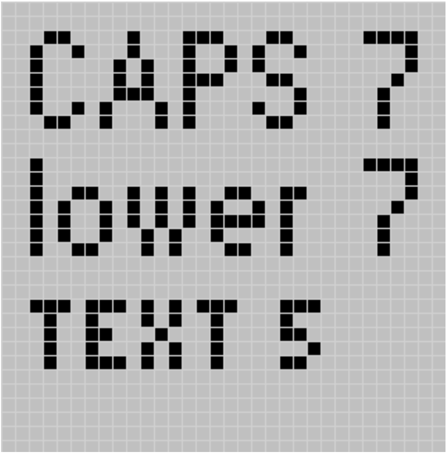
Text (7)

The larger text tool of the Bitmap Editor allows you to place text anywhere on the bitmap. Once selected, the mouse cursor shows a bracket image, representing the left side of the text. Once the location is selected, and the mouse button is pushed, a dialog box asks for the text to be inserted. The Text 7 tool is capable of upper and lower case letters and numbers, although blank spaces will be inserted in place of any other symbols. The text will be printed in the selected foreground color, and will start in the location specified by the mouse. In the case the text is too long to fit within the borders of the bitmap, the tool will cut off the text at the border and any remaining text will not be printed.

Text (5)

The smaller text tool of the Bitmap Editor draws characters that are 5 pixels high at any place on the bitmap. When the tool is selected, the mouse cursor becomes a bracket image representing the left border of the text. As with the Text-7 tool, once the mouse button is pushed, a dialog box prompts you to input the desired text. Unlike the Text-7 tool, the 5-pixel font has only upper case letters, and will print the inputted text in all upper-case regardless of the case used when inputting.

The following figure shows the results of both text tools.



Menu/Toolbar Items

The following provides details on the menu and toolbar commands.

File Menu

New

Creates a new blank bitmap without saving the previous bitmap. The blank bitmap is automatically gray, regardless of the selected background color.

Open

Opens an existing bitmap. Selecting this menu item opens a file selection dialog box, allowing you to selected the bitmap to open.

Note Since the bitmap editor only handles bitmaps that are 32 X 32, attempting to open a larger bitmap will result in an error message.

Browse

Consists of four separate menu items: Analog, DSP, Layout, and Template. Each item opens a special browse window to look through and select from specially defined preexisting bitmaps.

Save. Saves the current bitmap under its current name. If no name has been selected for this bitmap, a file dialog box is opened allowing you to input the new bitmap's name.

Save As. Opens a file dialog box to allow you to save the current bitmap under a user-defined name.

Exit. Closes the bitmap editor without saving the current bitmap.

Edit Menu

Undo

Redraws the current bitmap as it was before the last drawing command was performed.

Note Undo can only be used once in succession as only the most recent drawing command is remembered. When undoing a paste command, the paste selection remains in the clipboard, and can be pasted again immediately.

Cut

This is only applicable in Selection mode, and after a selection has been made. Copies the selected section to the clipboard and replaces the selection with background color.

Copy

This is only applicable in Selection mode, and after a selection has been made. Copies the selected section to the clipboard while leaving the original selection unchanged.

Paste

This is only applicable in Selection mode, and after having selected a section and either used the Cut or Copy command beforehand. Displays a rectangular image under the mouse cursor showing the size of the clipboard. Once the left mouse button is pressed, the selection is copied onto the bitmap in the designated location. If the selection reaches beyond the boundaries of the bitmap editor, the extraneous pixels are eliminated.

Clear Selection

This is only applicable in Selection mode, and after a selection has been made. Replaces the selected area with background color.

Clear All. Replaces the entire bitmap area with background color.

View Menu

Redraw Bitmap Preview. Refreshes the bitmap preview window to the side of the main bitmap display.

Redraw View. Refreshes the main bitmap window. Use if visual artifacts begin to appear on the screen.

Caption Guide Display.

Toggles the Caption Guidelines on and off. When activated, the bitmap window shows the approximate space designated for captions, for aid in bitmap design.

Image Guide Display

Toggles the Image Guidelines on and off. When activated, the bitmap window shows the approximate space designated for the main bitmap image, used as an aid in bitmap design.

Tools Menu

Add Caption

Allows you to input a caption to be placed in the caption area at the bottom of the bitmap window. The caption will be centered and printed in the selected foreground color. Captions that extend the boundaries of the bitmap window will have the left-most and right-most sections cut off.

Add Analog/RF Corner Tab

Draws a yellow corner on the bitmap, used to designate an Analog/RF item.

Add Smart Component Corner Tab. Draws a blue corner on the bitmap, used to designate a Smart Component.

Flip Selection Horizontal

This is only applicable in Selection mode, after a selection box has been drawn. Flips the selection across a vertical axis, creating a mirror image of the original image.

Flip Selection Vertical

This is only applicable in Selection mode, after a selection box has been drawn. Flips the selection across a horizontal axis, creating a vertically opposite image of the original.

Rotate Selection

This is only applicable in Selection mode, after a selection box has been drawn. Rotate also requires the selection box to be a square. Once the selection is made, Rotate redraws the selection after rotating the image 90 degrees clockwise.

Help Menu

User Manual

Brings up the help documentation for the bitmap editor.

About Bitmap Editor. Version and copyright information.

Color Selection Tools

The Selected Color Box at the bottom left of the bitmap editor shows the currently selected colors: the inner square is the foreground color, the outer square is the background color. The fifteen color boxes shown in Table 5-1are the available colors.

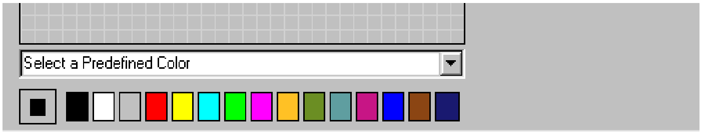


Table 5-1. Bitmap Editor Colors

Color	ADS Number
Red	1
Black	0
Yellow	2
Green	3
Light Blue	4
Blue	5
Purple	6
Gray	7
White	8
Dark Blue	13
Brown	29
Olive	27
Orange	82
Teal	63
Magenta	37

Clicking on one of the color boxes selects that color according to which mouse button is used:

- **Left Mouse Button**: Selected color is now the foreground color
- **Right Mouse Button**: Selected color is now the background color

Color Selection List

Another way to select a color is through the Color Selection List, found right above the color boxes. The List contains names for standard component items and the colors associated with them. This will assist in ensuring that custom bitmaps have the same color scheme and pattern as other ADS bitmaps. To select a color, simply choose the desired name from the list, and the foreground color will be adjusted automatically. The choices shown in Table 5-2 are available through the Color Selection List.

Table 5-2. Color Selection List Options

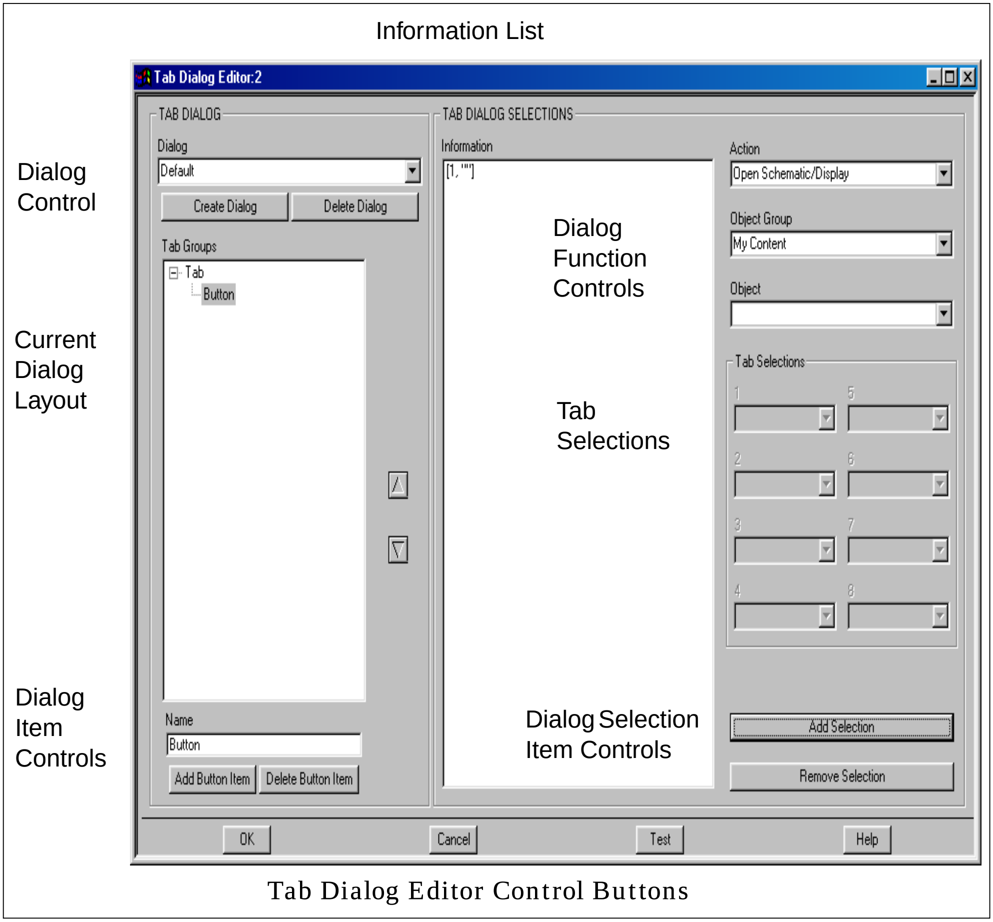
Name	Color
RF Symbol Outline	Black
RF Symbol Fill	White
RF Symbol Text	Black
RF Pins	Red
RF Graphics	Black
DSP Timed	Black
DSP Symbol Bodies	Black
DSP Any	Black
DSP Integer	Yellow
DSP Message	Teal
DSP Fix	Purple
DSP Float	Blue
DSP Graphics	Dark Blue
DSP Complex	Green
DSP Matrix	Orange

Bitmap Preview

The Bitmap Editor provides a bitmap preview window just to the right of the main layout grid. This window shows the created bitmap as it would appear as a palette item using a *.bmp* file. (This does not physically *create* the named *.bmp* file, just a temporary one for use as a preview. To create a .bmp file, use the *Save* or *Save As* commands in the Bitmap Editor menu.) The bitmap preview will be automatically updated each time any palette (Circle, Rectangle, Text, etc) or menu functions (cut, paste, rotation, etc) are performed. If you want to refresh the preview for any reason, select the *View* > *Refresh Bitmap Preview* from the menu.

Note The Bitmap Preview only exists on the Windows Version of the DesignGuide Developer Studio.

Chapter 6: Tab Dialog Editor



Tab Dialog Editor:2

Dialog: Default

Tab Groups:
- Tab
  - Button

Name: Button

Information: [1, ""]

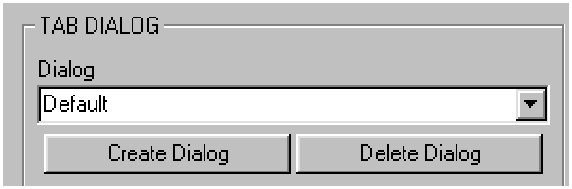
Action: Open Schematic/Display

Object Group: My Content

Object:

The Tab Dialog Editor is the tool you use to create custom tab dialog boxes. These tab dialog boxes may be attached to menus inside your DesignGuide to simplify the design process.

Dialog Controls



Tab Dialogs can be created, deleted and renamed using the Dialog Controls.

Creating Tab Dialogs

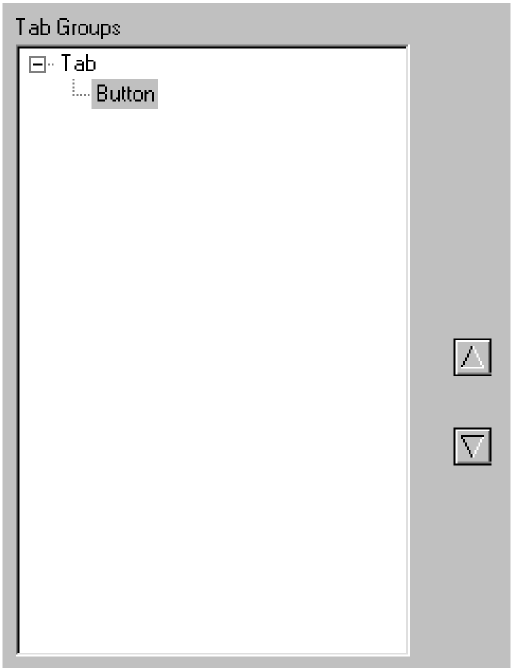
By simply clicking on the *Create Dialog* button, a default Tab Dialog will appear. You can change the name by highlighting the default name and entering the name that you want. After creating multiple Tab Dialogs you can browse through them using the pull-down menu.

Note The name that you give each Tab Dialog is what will appear inside the Menu Editor when you assign tab dialogs to menu items.

Deleting Tab Dialogs

Tab Dialogs are deleted by selecting the desired dialog name using the pull-down menu, then clicking on *Delete Dialog*.

Current Dialog Layout



Tab Dialogs are organized into small hierarchal tree list structures. There are only two levels to the structure:

- **Tab Names.** Appear on the top level
- **Button Names.** Appear on the second level.

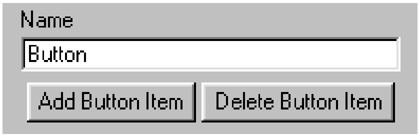
Reordering Tabs and Buttons

To change the order of the tabs or buttons:

1. Choose the tab or button you want to move.
2. Click on the up or down arrow depending on the direction you want to move the tab or button.
3. Continue arranging the tabs and buttons until you reach the desired order.

Note If selections have already been added to the information list, rearranging the tabs and buttons will not affect these selections. The combinations will change as you change the order of tabs and buttons.

Dialog Item Controls



Button

Adding, deleting, and renaming dialog items are done in this box.

Adding and Deleting Dialog Items

To add a dialog item:

1. Choose the tree level in the Current Dialog Layout corresponding to the item you want to add. The Add action button should change to either *Add Tab Group* or *Add Button Item*.
2. Click on the **Add** action button. A default item will appear in the Current Dialog Layout.

To delete a dialog item:

1. Choose the tree level in the Current Dialog Layout corresponding to the item you want to delete. The *Delete* action button should change to either *Delete Tab Group* or *Delete Button Item*.

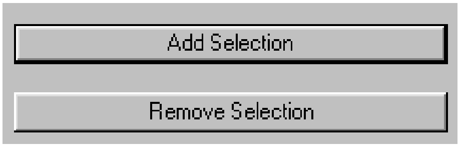
2. Click the **Delete** action button. The item will be deleted from the Current Dialog Layout.

Warning Deleting a tab will also delete all the buttons associated with it.

Dialog Item Names

As you select dialog items from the Current Dialog Layout window, their names will appear in the text box. By putting the cursor in the text box and typing the desired name, a dialog item's name may be changed. Button names are what will appear when choosing combinations.

Dialog Selection Item Controls



Selections must be added to the Information List before they can be made into a working tab dialog.

Adding and Removing Selections

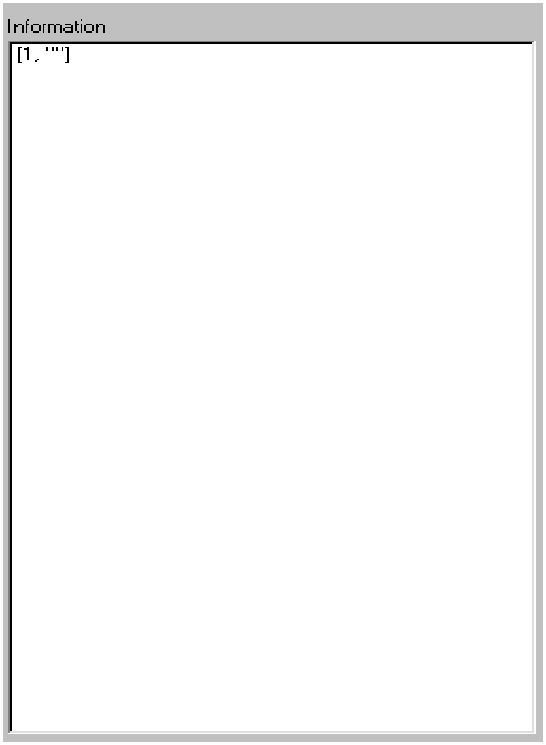
To add selections:

1. Click the **Add Selection** button until the desired amount of selections have been added. All added selections will initially default to button1 for all tabs.

To delete selections:

1. Highlight the selection in the Information List that you want to delete.
2. Click **Remove Selection**.

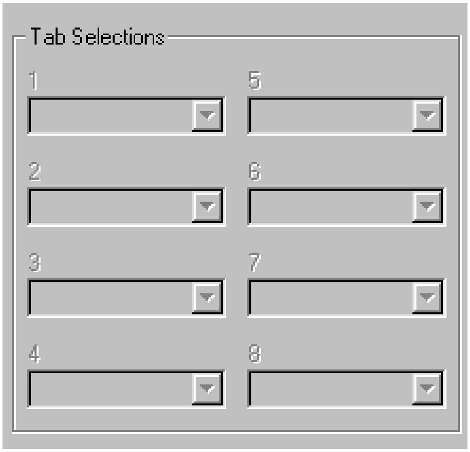
Information List



```
[1, '"']
```

The Information List displays all the added selections and combinations for the current tab dialog. The numbers correspond to the buttons and tabs that need to be chosen to activate the object. For example, if you have *1,3,2* on one of the selections, it refers to the first button on the first tab, the third button on the second tab, and the second button on the fourth tab. After this combination, each selection shows the resultant action.

Tab Selections



The logic for each selection is made in the Tab Selections pull-down menus.

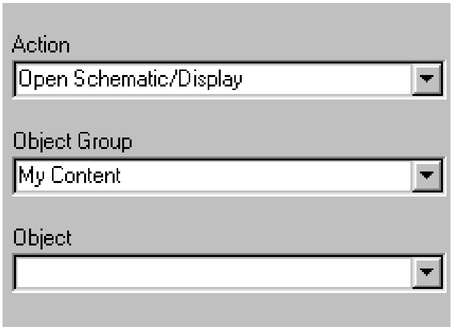
Adding and Changing Logic

To add or change logic:

1. Choose the selection from the Information List that you want to change. Notice that, once selected, all pull-down menus will conform to the highlighted selection.
2. Using the pull-down menus change the selection logic to fit your DesignGuide.

The Tab Selections pull-down menus are arranged according to your tabs and buttons. Pull-down menu number 1 corresponds to the first tab, pull-down menu number 2 corresponds to the second tab, etc. Likewise the first item on each pull-down menu refers to the first button of each corresponding tab, etc.

Dialog Function Controls



Action: Open Schematic/Display

Object Group: My Content

Object:

Each selection needs an Action, Object Group, and an Object.

Actions and Object

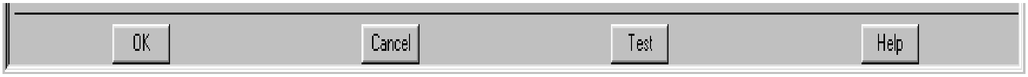
Actions and Objects help define what the Tab Dialog will do. Each selection, therefore, needs both an action and an object in order for it to do anything.

To change the action or object for a selection:

1. Choose a selection from the Information List.
2. Select the desired action from the Action pull-down menu.
3. Select the correct group from the Object Group pull-down menu.
4. Select the desired object from the Object pull-down menu.

Notice that the selection will change on the Information list as you make changes from the pull-down menus.

Tab Dialog Editor Control Buttons



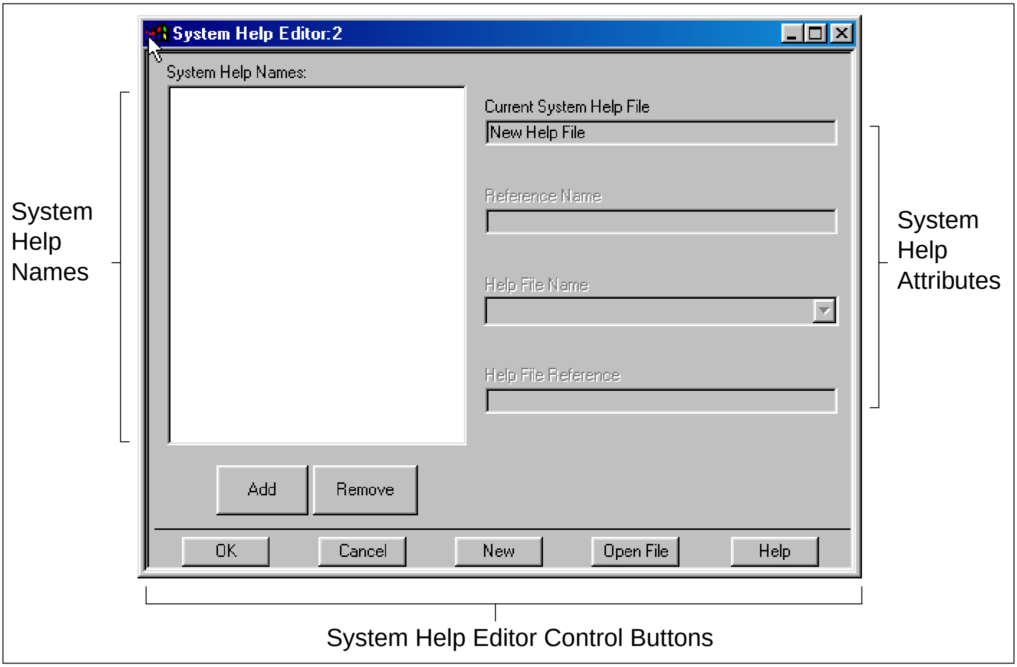
Tab Dialog Editor Control Buttons

- **OK.** Exits the Tab Dialog Editor.
- **Cancel.** Closes the Tab Dialog Editor.
- **Test.** Displays what the Tab Dialog will actually look like. Changing button selections will show the change in action.
- **Help.** Opens this help documentation.

Chapter 7: System Help Editor

The System Help Editor is a tool designed to help DesignGuide developers easily coordinate and link help files and other documentation into a custom DesignGuide

.



System Help Editor:2

System Help Names:

Current System Help File: New Help File

Reference Name:

Help File Name:

Help File Reference:

Using the System Help Editor

The System Help Editor creates a help index file, which allows direct reference to HTML files through the DesignGuide Developer Studio. This allows custom menu items to bring up a specified section of an HTML document as help for those who use a custom design guide.

A help index file requires the following three items:

- **Reference Name**. User-defined name used within the Developer Studio
- **Help File Name**. HTML (.html) file referenced

- **Help File Reference.** Reference used within the above HTML file to refer to the desired section of documentation.

Creating a Help Index File

To create a help index file, do the following:

1. Copy all the documentation (*.html*) files into the */doc/* subdirectory of the studio project directory. Make note of all the HTML references within each file that need to be indexed.
2. Choose **Add** to create a new Help System index.
3. Input an HTML reference into the *Help File Reference* text box.
4. Select the file in which the reference is found with the *Help File Name* drop down list.
5. Change the **Reference Name** to the desired name. This is the name that will appear in the Menu and Palette Editor when help files are used.
6. Repeat Steps 2 through 5 for each HTML reference to be linked.
7. Choose **OK**. This will copy the *.html* files to the proper place in ADS and create an index file.

Using Help References

After the help index file is created, help docs can be tied into a custom menu by the following process:

1. Open the custom menu using the Menu Editor.
2. Select the menu item to which the help file will be attached.
3. Choose **Open Help File** from the *Action* list.
4. Choose **Help System** from the *Group* list.
5. Open the **Object** list. The list of reference names created in the System Help Editor will appear.
6. Choose the reference name to be assigned to the menu item.

When the menu is built, selecting that menu item will automatically start your Web browser and bring the referenced file to the screen.

Removing Help References

To remove a help reference from the list, simply select the reference and choose the **Remove** button. Once the *OK* button is chosen, the help index file will be recreated without the deleted reference.

Note The System Help Editor does not have an Undo function, so deleted help references will need to be recreated if deleted by mistake.

System Help Editor Control Buttons

The Control Buttons on the bottom of the System Help Editor help you in saving and loading system help files, as well as creating a new system file.

Opening and Editing an Existing file

To open and edit an existing system help file:

1. Choose **Open File** from the System Help Editor Control Buttons.
2. Select an index file from the file dialog box.

Note Index files are found in the */doc/addons_indexes/hpeesofsim* subdirectory of the main ADS directory. They are simply text files without a file extension.

Opening a New Help File

By default, the System Help Editor opens to the associated help file for the studio project currently opened in the DesignGuide Developer Studio. Choosing the **New** button will clear the current help file, allowing you to create a new one.

If a separate help file has been opened using the *Open File* button, however, the *New* button will return the help editor to the current studio system help file, allowing users to retain changes that were made to the current studio help file while still able to open and edit outside help files as well.

The *Current System Help File* box displays whether the current help file is an outside file, or the current studio project help file.

Chapter 8: Quick Help Editor

The Quick Help Editor provides a way to display small help comments to you. These *quickhelp*s can be used in any palette or menu item and can help reduce the amount of on-line documentation needed for your projects. Following are descriptions of the various buttons, commands, and fields used in the Quick Help Editor.

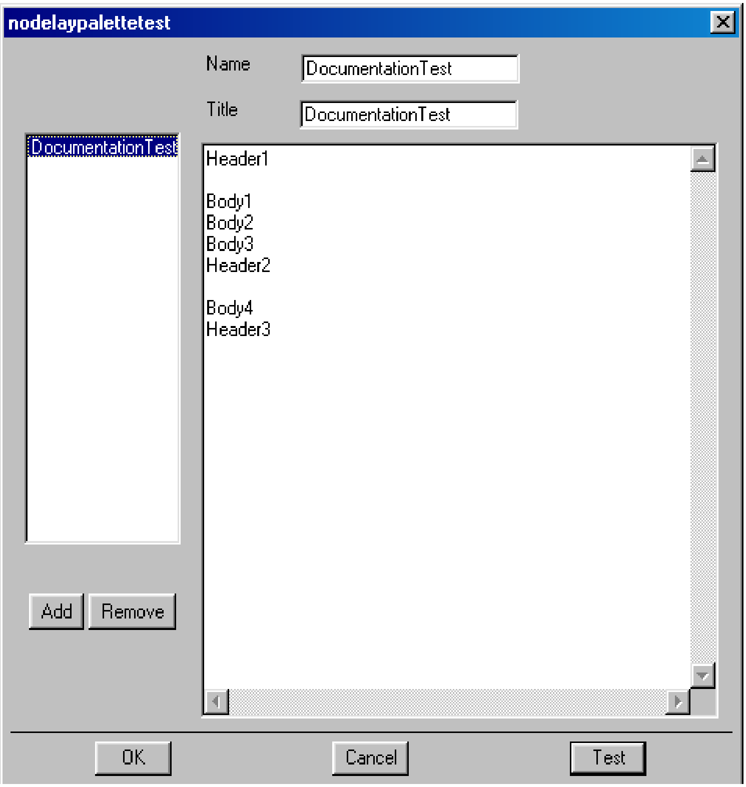
Add Button, Used to add quickhelp items. The default name is defaultX, where X is an integer representing the next unused integer. An item must be added before you can add text and give it a caption.

Remove Button. Removes the current quickhelp item selected. If no quickhelp item is selected, nothing will happen.

Name. Corresponds to the name of the quickhelp item. This name is used in the dropdown list of the palette and menu editors. All names must be unique. To change the name select the item you wish to change, highlight the text in the name field, and replace with the new name you wish to use.

Title. The string that appears at the top of the quickhelp dialog box. To change the title, select a quickhelp item, highlight the text in the title field, and replace with the new title.

Main Text Field. The dialog part of the quickhelp dialog box. The quickhelp dialog box can show text as either bolded or normal. To bold the text, place two returns after the line of text. To make the text normal, only use one return. View the example that follows. The Quick Help Editor appears as shown here.



nodelaypalettetest

Name DocumentationTest

Title DocumentationTest

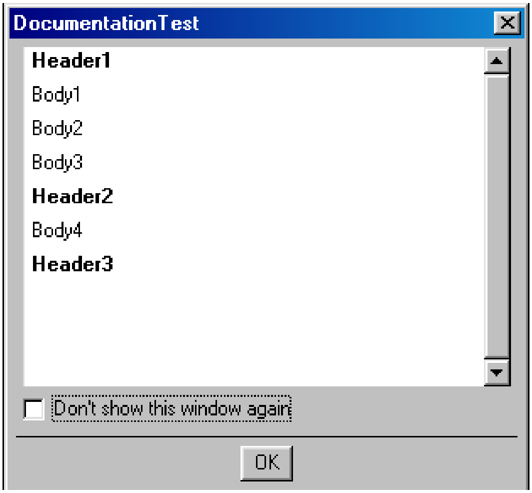
DocumentationTest

```
Header1

Body1
Body2
Body3
Header2

Body4
Header3
```

The Quick Help dialog box appears as shown here.



Test Button. The *Test* button will enable you to view the quickhelp dialog.

Note. The *Don't show this window again* checkbox is disabled in test mode. Checking it has no effect.

Quick Help Dialog

The quickhelp dialog is the dialog box that pops up when a menu or palette item is selected. It will only function if the palette or menu item has been assigned a quickhelp name. It is a scrollable table that will expand horizontally to fit longer lined text and scroll vertically to show all lines of text.

"Don't Show This Window Again" Checkbox

The checkbox will *turn off* the quickhelp dialog for that quickhelp name.

Note Any palette or menu item that shares the quickhelp name will also be disabled.

Quick Help Config File

The quickhelp information regarding a studio project is stored in the *adsfiles\hpeesof\config* directory with the file named after the project name. The format is the quickhelp name followed by a *=TRUE* or *=FALSE*. This corresponds to the *Don't Show This Window Again* Checkbox. All names default to TRUE which means all names will show the quickhelp dialog box when the palette or menu is selected. If the box gets checked, the corresponding name in the config file gets changed to *name=FALSE*, where *name* is the quickhelp registered name.

Note To manually turn on the quickhelp dialog box for a quickhelp name, open the corresponding *config* file and change the boolean value to TRUE, and to turn off the dialog box. change the value to FALSE.

Chapter 9: Lab Example Summary

This chapter provides a a basic summary of the steps used to develop DesignGuides, using some actual lab examples. The user-created DesignGuide example projects referenced in this chapter include the following:

- *dg_budget_pr*j (Budget DesignGuide)
- *dg_radar_prj* (Radar DesignGuide)
- *dg_rfic_models_prj* (RFIC Model DesignGuide)

To access these examples from the Advanced Design System Main window, select *File* > *Example Project* > *Training* > *DesignGuide*, and select the example project you want to open.

This chapter is organized as follows:

- Basic information before starting
- Tab Dialog Editor Example (Budget DesignGuide)
- Menu and Palette Editor Example (Radar DesignGuide)
- Advanced Topics (RFIC Model DesignGuide)
- Advanced Topics (Documentation)

The primary steps in creating a DesignGuide using the Developer Studio are as follows:

- Create DesignGuide structure, using the following:
    - Menu Editor
    - Palette Editor
    - Tab Dialog Editor
    - Bitmap Editor
    - Quick Help Editor
    - System Help
    - Report Problem Summary
- Associate actions to menu selections
    - Open Schematics / Data Displays

  - Display palettes
  - Open Tab dialogs
  - Insert templates, components, and sub-networks
  - Execute custom AEL routines
- Produce DesignGuides for PC or Unix
- Create a Revision History
- Create single file archive for easy distribution via worldwide web or e-mail

Basic Information Before Starting

The following is some basic data that will be helpful before you begin your development process.

Navigating Through the Developer Studio

To access the main dialog box for the Developer Studio, in the ADS Main window, select *DesignGuide* > *DesignGuide Developer Studio* > *Start DesignGuide Studio*.

For a basic introduction to navigating through the DesignGuide menus and commands, refer to the section, Developer Studio Overview, in Chapter 1, Getting Started with DesignGuide Developer Studio.

Important Files and Directories

Following is the DesignGuide directory structure in $HPEESOF_DIR:

$HPEESOF_DIR\designguides\ael\vapi_runtime.atf

$HPEESOF_DIR\designguides\projects\<designguide name>

ael

data

doc

networks

ui

The source files can be any ADS project directory. The ADS project must be located at $HOME. It cannot be in a different directory, There is no browse capability for the Developer Studio.

$HOME \ <ads_project_name>_prj

The archive files are located in the *studio_file* directory:

$HOME \ Studio_files \ <designguide name> \ <designguide name>.deb.

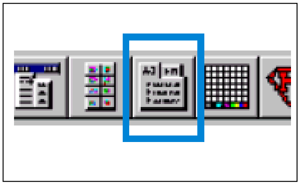
Creating Tab Dialogs

This section uses the example project *dg_budget_prj* (Budget DesignGuide) to provide a brief summary of the key steps in creating tab dialogs, including:

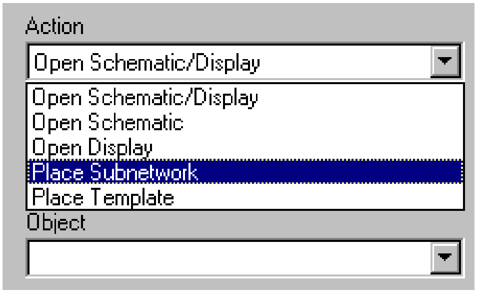
- Creating a tab dialog user interface for opening schematics and data displays
- Creating a menu for opening the tab dialog
- Showing the difference between a *build* and a *package*

This example begins with a collection of related and similarly named schematics, data displays, and datasets. There are no subcircuits or other dependent files. A matrix of user-selectable categories is determined, and each schematic represents one of the possible selection combination.The key actions in this example include:

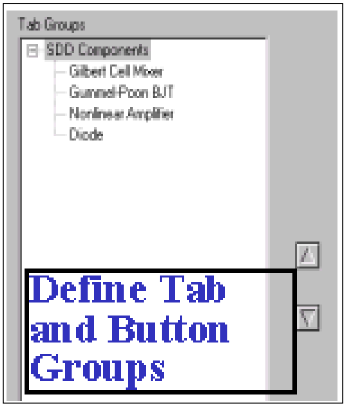
1. Choose the Tab **Dialog Editor** button on the dialog box toolbar.



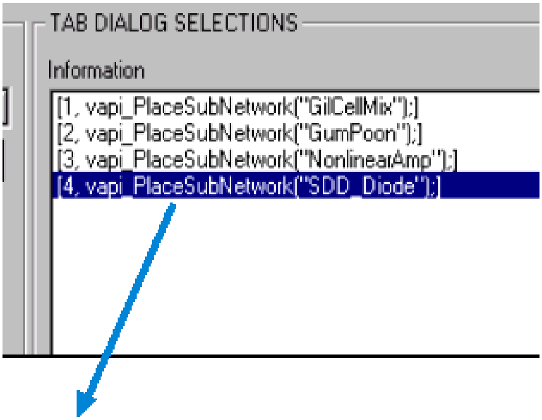
2. From the Action field of the dialog box, do any of the following:



3. Define the tab and button groups.

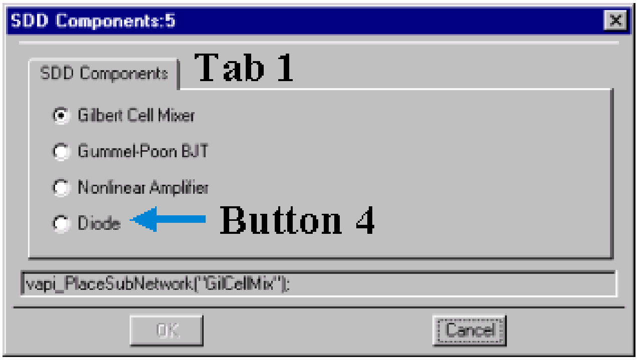


4. View the summary of tab definitions, as shown here.



[1, vapi_PlaceSubNetwork("GilCellMix");]
[2, vapi_PlaceSubNetwork("GumPoon");]
[3, vapi_PlaceSubNetwork("NonlinearAmp");]
[4, vapi_PlaceSubNetwork("SDD_Diode");]

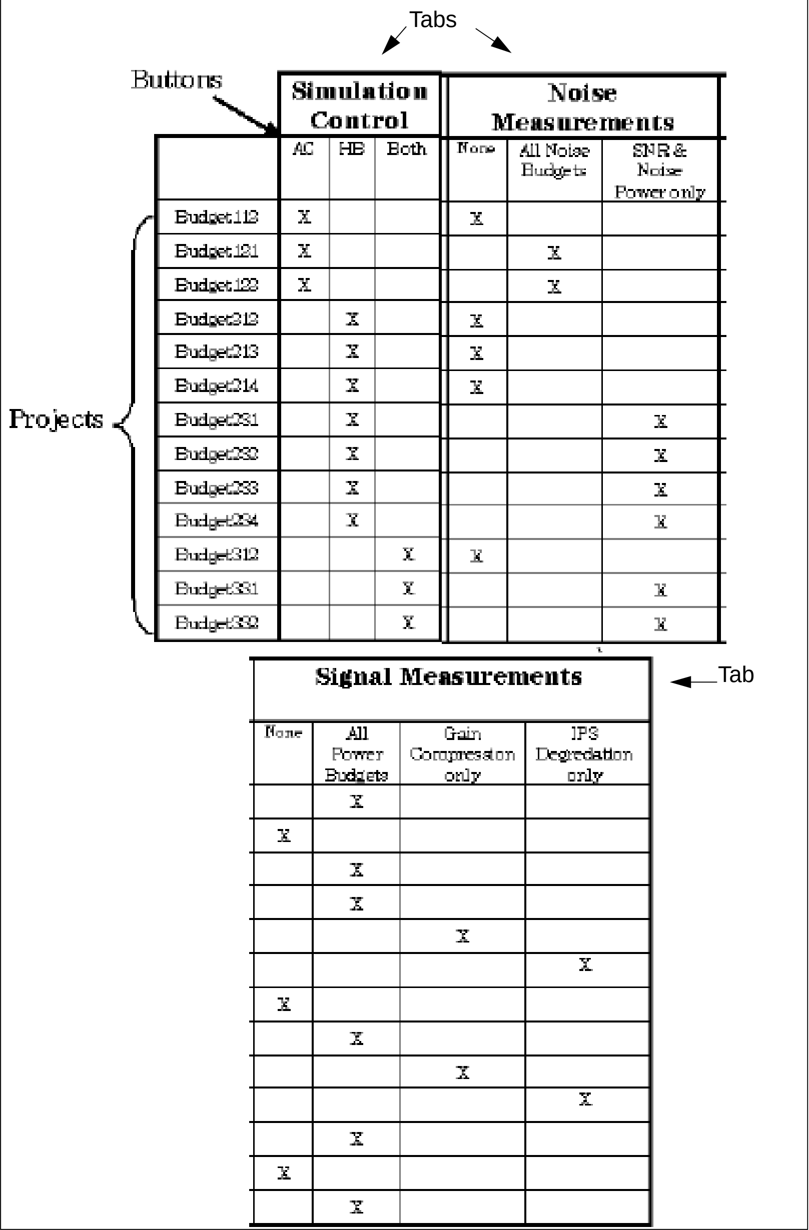
Button 4 from tab 1 is defined to place the SDD_Diode subnetwork.



● Gilbert Cell Mixer
○ Gummel-Poon BJT
○ Nonlinear Amplifier
○ Diode

vapi_PlaceSubNetwork("GilCellMix");

We recommend that you define and create a plan for your tab dialog structure, as shown in the following example

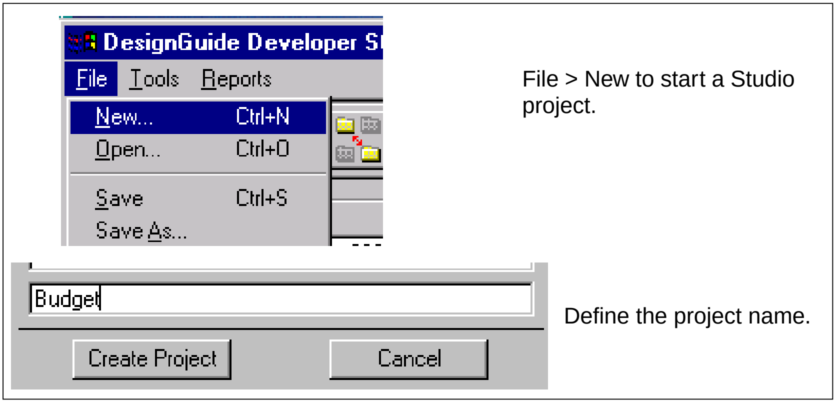

Tabs
Buttons
Projects
Tab

	Simulation Control			Noise Measurements		
	AC	HB	Both	None	All Noise Budgets	SNR & Noise Power only
Budget113	X			X		
Budget121	X				X	
Budget123	X				X	
Budget212		X		X		
Budget213		X		X		
Budget214		X		X		
Budget231		X				X
Budget232		X				X
Budget233		X				X
Budget234		X				X
Budget312			X	X		
Budget331			X			X
Budget332			X			X

Signal Measurements			
None	All Power Budgets	Gain Compression only	IPS Degredation only
	X		
X			
	X		
	X		
		X	
			X
X			
	X		
		X	
			X
	X		
X			
	X		

Starting a New Studio File. Following are the basic steps in starting a new studio file.

1. Place a project file (*dg_budget_prj* in this example) in the $HOME directory, using the **Copy Project** command from the Main window.
2. From the Main Window, select **DesignGuide** > **DesignGuide Developer Studio** > **Start DesignGuide Studio**.
3. From the DesignGuide Developer Studio window select **File** > **New**.
4. From the New Project dialog box, define the name of the project (for example, **Budget)**, then select the **Create Project** button.

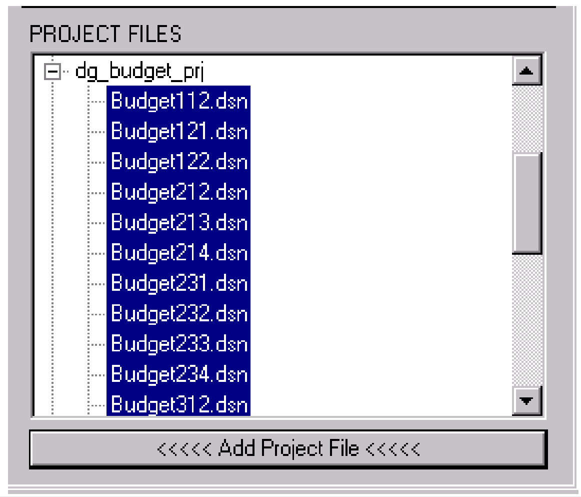


File > New to start a Studio project.

Define the project name.

Registering the Content. The Content Register is used to do the following:

- Define the location of designs, data displays, and datasets
- Map content from multiple ADS projects
- Define dependent files
- The *build* copies the content to one location.



Select design files in dg_budget_prj

Map the project by selecting Add Project File.

5. From the DesignGuide Developer Studio window, in the Content Browser, expand the folder hierarchy of **Budget** by selecting the **+** symbol by the name.

6. In the Content Browser window, expand the folder hierarchy of **My Content** by selecting the **+** symbol by the name.

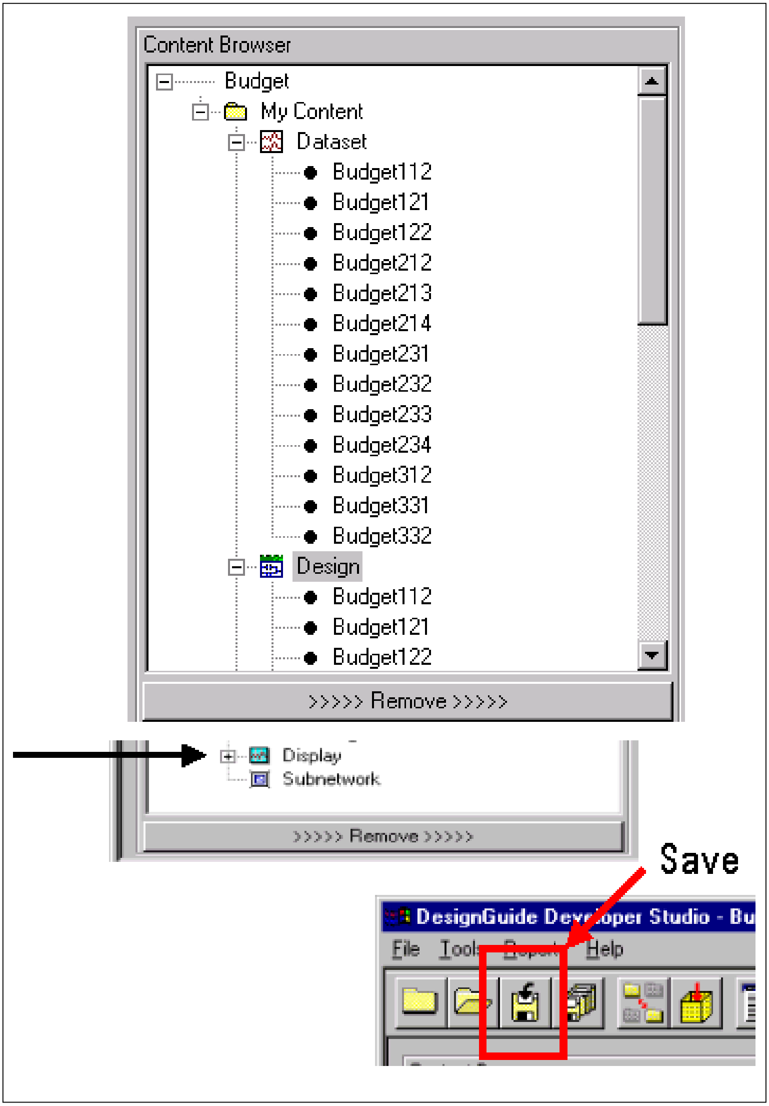
7. Select the **Design** icon in the Content Browser so that it is highlighted. The ADS project files are now listed.

8. In the Project Files window, expand the folder hierarchy of *dg_budget_prj* by selecting the **+** symbol by the name of the project.

9. The Windows convention of **Shift** and **Ctrl** work for file selection. Select all of the design files by selecting the first design file, scrolling down to the last design files, and hold the shift key down while selecting the last design file.

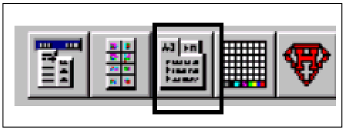
10. Select **Add Project File.**

11. Continue the same content registry process for the data displays and the datasets, as shown in the following illustration. It is advisable to save the Studio project. In the DesignGuide Developer Studio window, select **File** > **Save** or select the **Save** icon on the toolbar.


Content Browser
Budget
My Content
Dataset
Budget112
Budget121
Budget122
Budget212
Budget213
Budget214
Budget231
Budget232
Budget233
Budget234
Budget312
Budget331
Budget332
Design
Budget112
Budget121
Budget122
>>>>> Remove >>>>>
Display
Subnetwork
>>>>> Remove >>>>>
Save
DesignGuide Developer Studio - Bu
File
Tools
Help

Using the Tab Dialog Editor. The steps that follow show the use of the Tab Dialog Editor.

12. From the DesignGuide Developer Studio window toolbar, open the Tab Dialog Editor. Refer to the menu plan introduced previously in this section.



13. Define three Tabs by choosing the **Add Tab Group** button. If you expand one of the inserted tabs, and select this button, the **Add Tab Group** button changes to **Add Button Group**.
14. Define three buttons for tab 1 using the **Add Button Group** button.
15. Define three buttons for tab 2, and four buttons for tab 3.
16. Change the name of the tabs to the following:
    - Tab 1 to **Simulation Control**
    - Tab 2 to **Noise Measurements**
    - Tab 3 to **Signal Measurements**
17. Change the tab 1 button names to the following:
    - Button 1 to **AC**
    - Button 2 to **HB**
    - Button 3 to **Both AC and HB**

Note The name order of the buttons is important for proper mapping to actions.

18. Change the tab 2 button names to the following:
    - Button 1 to **None**
    - Button 2 to **All Noise Budgets**
    - Button 3 to **SNR and Noise Power only**

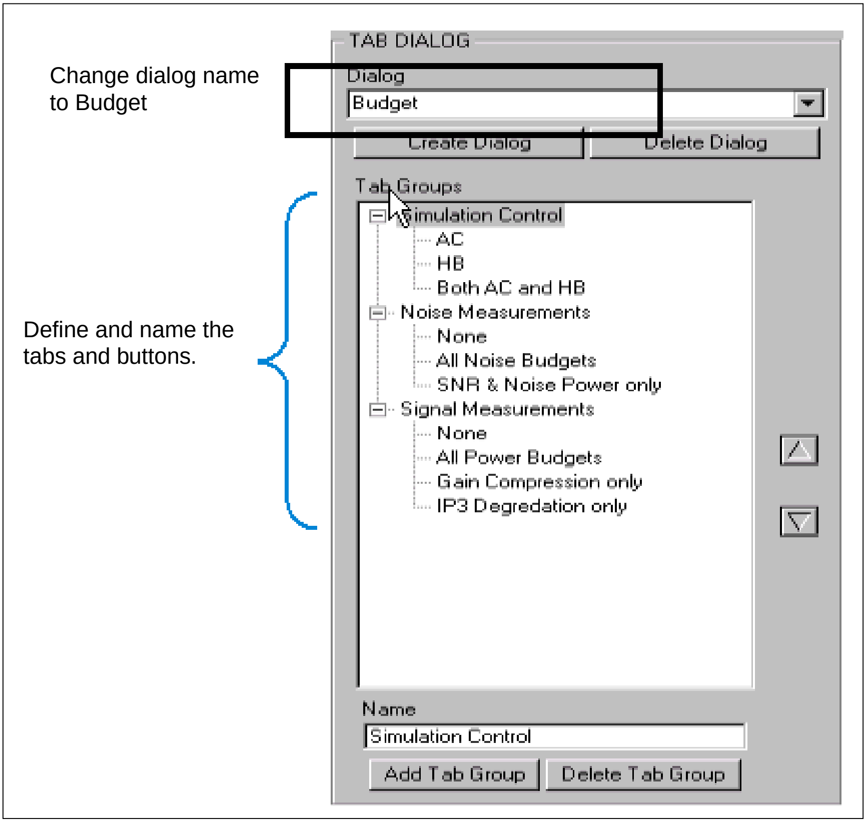
19. Change the tab 3 button names to the following:

    Button 1 to **None**

    Button 2 to **All Power Budgets**

    Button 3 to **Gain Compression only**

    Button 4 to **IP3 degradation only**



Dialog: Budget

- Simulation Control
  - AC
  - HB
  - Both AC and HB
- Noise Measurements
  - None
  - All Noise Budgets
  - SNR & Noise Power only
- Signal Measurements
  - None
  - All Power Budgets
  - Gain Compression only
  - IP3 Degredation only

Name: Simulation Control

Change dialog name to Budget

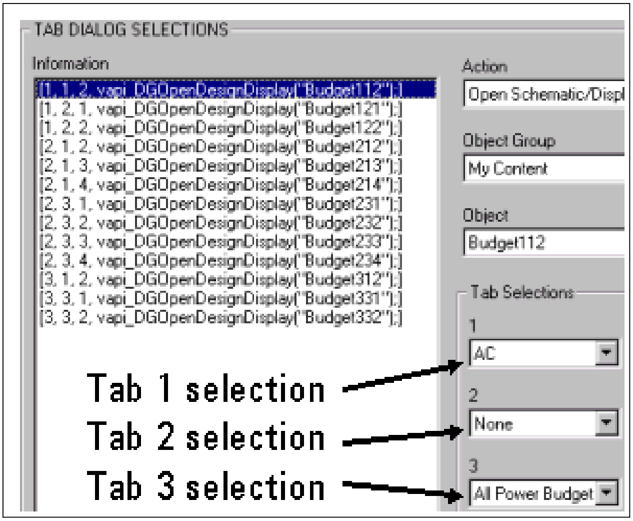
Define and name the tabs and buttons.

20. From the Tab Dialog Editor window, after the tabs and buttons have been named, add 13 tab selections (one for each design file) using the **Add Selections** button.

21. Repeat the following process for each of the 13 tab selections.

- Select a tab selection in the information window so that it is highlighted.
- Select the action to **open schematic/display**.
- Select the object group **My Content.**
- Select the object (start with **Budget112**).
- Select the combination of buttons that will define the action. For example, the object *budget213* refers to button 2 from tab 1, button 1 from tab 2, and button 3 from tab 3.
- Choose the **Test** button to see a preview of the tab dialog.

22. When all 13 definitions are complete, close the Tab Dialog Editor window.
23. Save your work.



Information

```
[1, 1, 2, vapi_DGOpenDesignDisplay("Budget112");]
[1, 2, 1, vapi_DGOpenDesignDisplay("Budget121");]
[1, 2, 2, vapi_DGOpenDesignDisplay("Budget122");]
[2, 1, 2, vapi_DGOpenDesignDisplay("Budget212");]
[2, 1, 3, vapi_DGOpenDesignDisplay("Budget213");]
[2, 1, 4, vapi_DGOpenDesignDisplay("Budget214");]
[2, 3, 1, vapi_DGOpenDesignDisplay("Budget231");]
[2, 3, 2, vapi_DGOpenDesignDisplay("Budget232");]
[2, 3, 3, vapi_DGOpenDesignDisplay("Budget233");]
[2, 3, 4, vapi_DGOpenDesignDisplay("Budget234");]
[3, 1, 2, vapi_DGOpenDesignDisplay("Budget312");]
[3, 3, 1, vapi_DGOpenDesignDisplay("Budget331");]
[3, 3, 2, vapi_DGOpenDesignDisplay("Budget332");]
```

Action: Open Schematic/Displ

Object Group: My Content

Object: Budget112

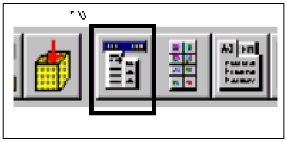
1: AC

2: None

3: All Power Budget

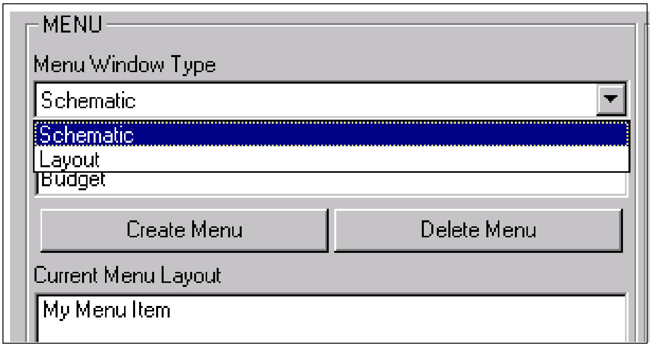
Using the Menu Editor. The steps that follow show the use of the Menu Editor.

24. From the DesignGuide Developer Studio Window toolbar, open the Menu Editor.



25. Decide if you need to have DesignGuides added to the Schematic and/or Layout window. For this example, we do not need them in the Layout window.

- For Menu Window Type, select **Layout.**
- Select **Delete Menu**. The Menu Window Type should show *Schematic*. Keep this.



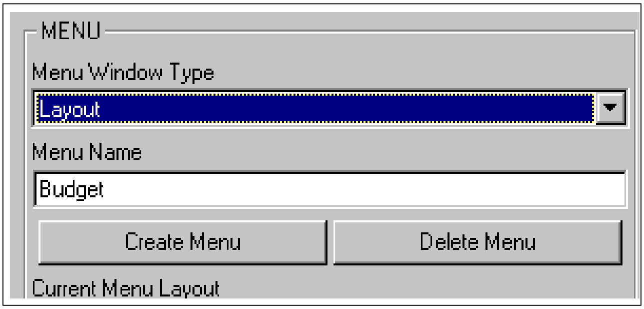
MENU

Menu Window Type

Schematic

Current Menu Layout

My Menu Item



MENU

Menu Window Type

Layout

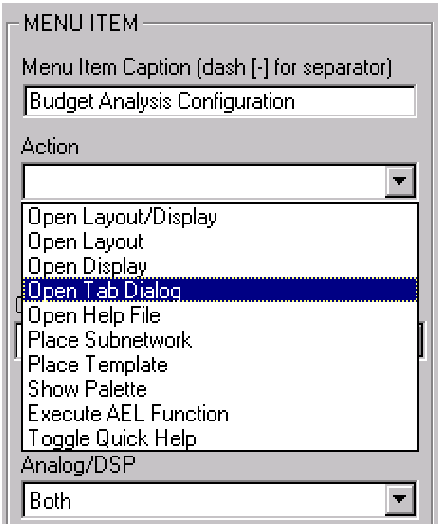
Menu Name

Budget

- Set the Menu Name to **Budget**. Press **Enter** when you finish typing in the new name (which will appear in the DesignGuide menu).

Defining a menu. The tab dialog needs to be configured into a menu, as explained in the next set of steps.

26. In the Menu Editor window, make sure that the Menu Window Type is set to **Schematic**.
27. Select the first entry in the Current Menu Layout to define it. Change the Menu Item Caption name to **Budget Analysis Configuration**.
28. Use the Action pull-down menu to define **Open Tab Dialog**.
29. Use the Object Group pull-down menu to select **My Contents**.
30. Use the Object pull-down menu to select the **Budget** tab dialog.
31. Use the Analog/DSP pull-down menu to select **Analog RF**.
32. Test the menu. After inspection, close the Menu Editor Window.
33. Save your work.



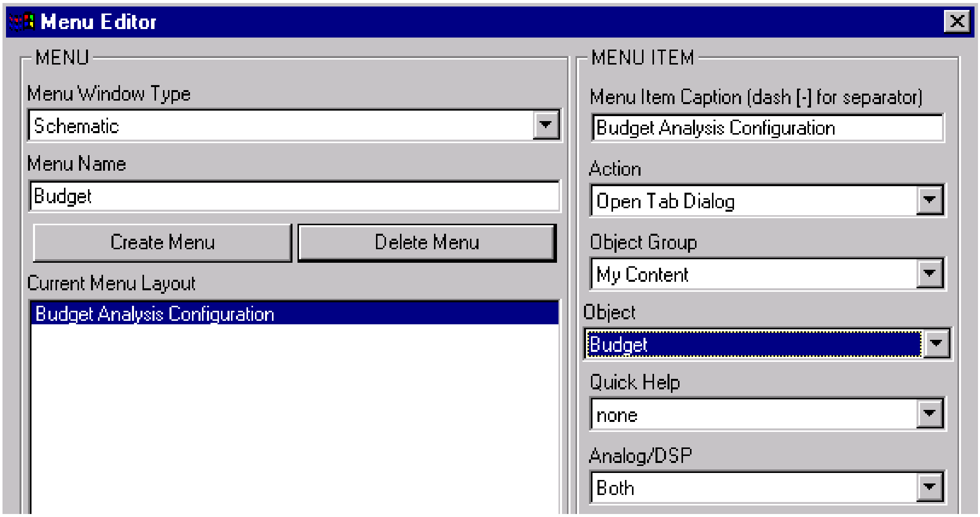
MENU ITEM

Menu Item Caption (dash [-] for separator): Budget Analysis Configuration

Action:

Analog/DSP: Both

Define the menu item caption and the action Open Tab Dialog.



Menu Editor

MENU

Menu Window Type: Schematic

Menu Name: Budget

Current Menu Layout: Budget Analysis Configuration

MENU ITEM

Menu Item Caption (dash [-] for separator): Budget Analysis Configuration

Action: Open Tab Dialog

Object Group: My Content

Object: Budget

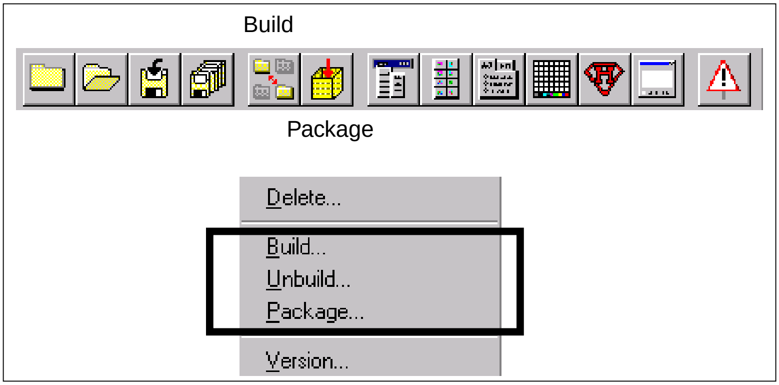
Quick Help: none

Analog/DSP: Both

Complete the menu definition.

Build Command Definitions. Prior to building the DesignGuide, it's useful to review the build command definitions:

- The *Build* function copies all content files and UI definitions to the directory *$HPEESOF_DIR/designguides/projects*. This is not the same as installing. It is usually used for debugging prior to packaging.
- The *Unbuild* function removes the files and UI definition from the directory *$HPEESOF_DIR/designguides/projects*.
- The *Package function* builds the DesignGuide, then archives it by producing a DEBIAN file. This can then be installed in ADS using the Add DesignGuide selection in the ADS Main window.



Building the DesignGuide. The DesignGuide can be built and/or packaged. For this example, we will package the DesignGuide, as it will produce an archive file.

34. From the DesignGuide Developer Studio window, select the **Package** toolbar button.

35. Define the Build Name, DesignGuide Name, version, dependency and description.

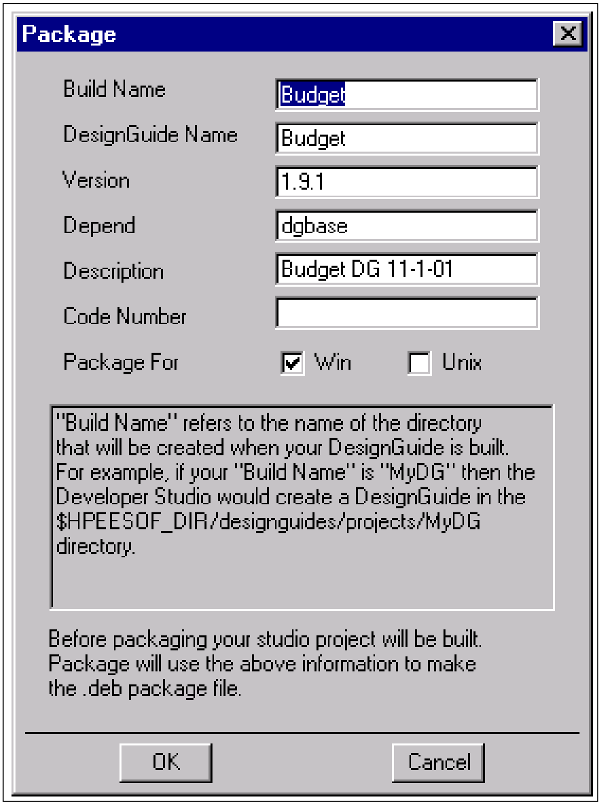
36. Select whether the build is for PC or for UNIX systems.

37. It's advisable to save the Studio project. Select **File** > **Save** from the DesignGuide Developer Studio window, or select the **Save** icon on the toolbar.

Note Code Number is used only if Agilent issues a license to secure it to a FlexLM codeword.

In the Package dialog box, the main fields are defined as follows:

- *Build* name is the name of the build directory in *$HPEESOF_DIR/designguides/projects*.
- *DesignGuideName* is the name of the DesignGuide file when it is packaged. (Characters such as underscore (_) can't be used.) Because different builds are needed for PCs and UNIX (if a palette is defined), you can add the build type as a suffix to the name (such as *pc* or *ux*).
- *Depend* defines a dependency on other required DEBIAN packages. The DEBIAN package file is located at *$HOME/studio_files/<designguidename>*.



Build Name: Budget

DesignGuide Name: Budget

Version: 1.9.1

Depend: dgbase

Description: Budget DG 11-1-01

Code Number:

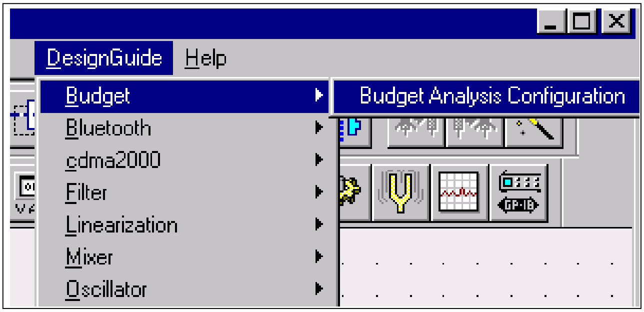
Package For ☑ Win ☐ Unix

"Build Name" refers to the name of the directory that will be created when your DesignGuide is built. For example, if your "Build Name" is "MyDG" then the Developer Studio would create a DesignGuide in the $HPEESOF_DIR/designguides/projects/MyDG directory.

Before packaging your studio project will be built. Package will use the above information to make the .deb package file.

Checking the Installation. The DesignGuide needs to be built and/or packaged, then checked.

38. We recommend that you check the DesignGuide by opening a new project file, then opening a schematic page to check the DesignGuide menu. If there are problems with the installation, try shutting down ADS and re-starting ADS.



39. To remove a DesignGuide after it has been built, select **File** > **Unbuild** from the DesignGuide Developer Studio Window.

40. To remove a DesignGuide after it has been packaged and added to ADS, select the **DesignGuide** > **Remove DesignGuide** command from the ADS Main window.

41. Save the DesignGuide project file.

Adding/Removing DesignGuides. DesignGuides are added or removed by selecting *DesignGuide* > *Add Design Guide* or *DesignGuide* > *List/Remove DesignGuide*, respectively, from the ADS Main window. The definitions are as follows:

- *Add DesignGuide* installs the DEBIAN archive file for the DesignGuide. This is a substitute for the *hpessofpkg -i* command.
- *RemoveDesignGuide* removes the DesignGuide only when it has been added using the AddDesignGuide command. This is a substitute for the *hpessofpkg -r* command. It cannot be used with a *build* DesignGuide.

Working with the Menus and Palette Editor

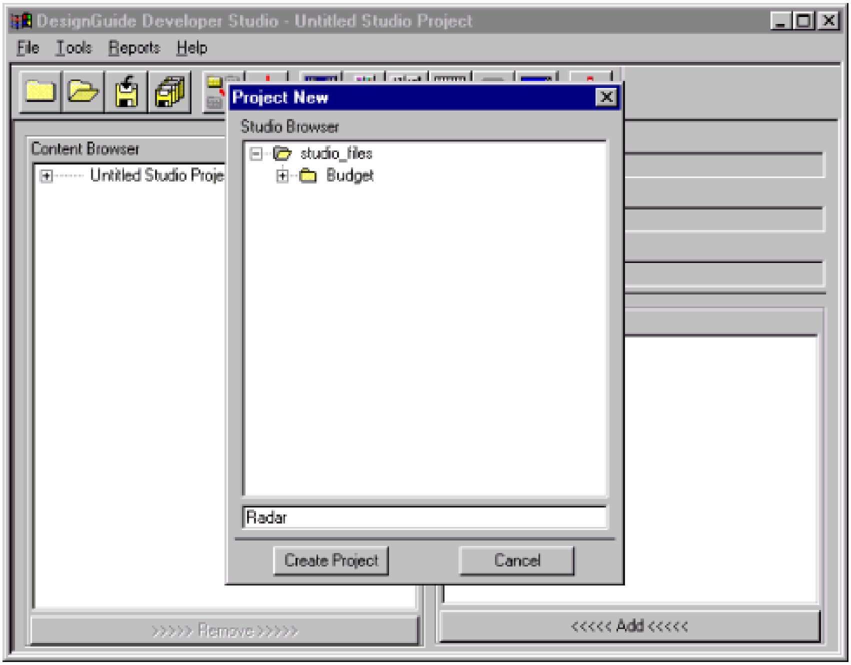
This section uses the example project *dg_radar_pr*j (Radar DesignGuide) to provide a brief summary of the following development steps:

- Creating a palette of subnetworks for the schematic
- Creating a menu as an alternate to a palette
- Using the Bitmap Editor
- Showing the use of the Quick Help

For this example, a single schematic with several subnetworks was used. The subnetworks are configured into a palette of related components. The subnetworks are also configured into a menu.

Building a New Project. You will need the project (for this example, *dg_Radar_prj)* to reside in your $HOME directory. This is in the *Examples/Training/Designguides directory*.

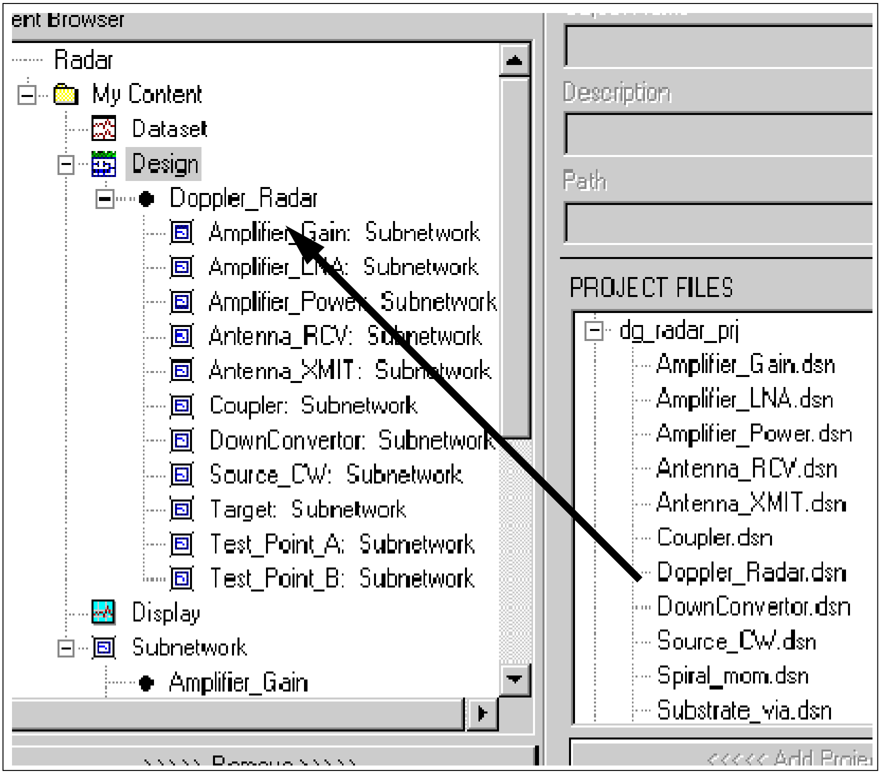
1. From the Main window, select **DesignGuide** > **DesignGuide Developer Studio** > **Start DesignGuide Studio**.
2. From the DesignGuide Developer Studio window, select **File** > **New**.
3. In the New Project dialog box, define the name of the project as **Radar**, then choose the **Create Project** button.



Radar

Registering the Content. This section shows the use of the Content Register.

4. From the DesignGuide Developer Studio Window, in the Content Browser, expand the folder hierarchy of **Radar** by selecting the **+** symbol by the name.
5. In the Content Browser window, expand the folder hierarchy of **My Content** by selecting the **+** symbol by the name.
6. Select the design icon in the Content Browser so that it is highlighted.
7. In the Project Files window, expand the folder hierarchy of **dg_radar_prj** by selecting the **+** symbol by the name of the project.
8. Select the top level design file named **Doppler_Radar.dsn**.
9. Select **Add Project File**.



10. The associated subnetworks are automatically added as dependent files.

Note Use the **Shift** key while selecting all the design files, then use the **Control** key to unselect **Doppler_Radar.dsn**.

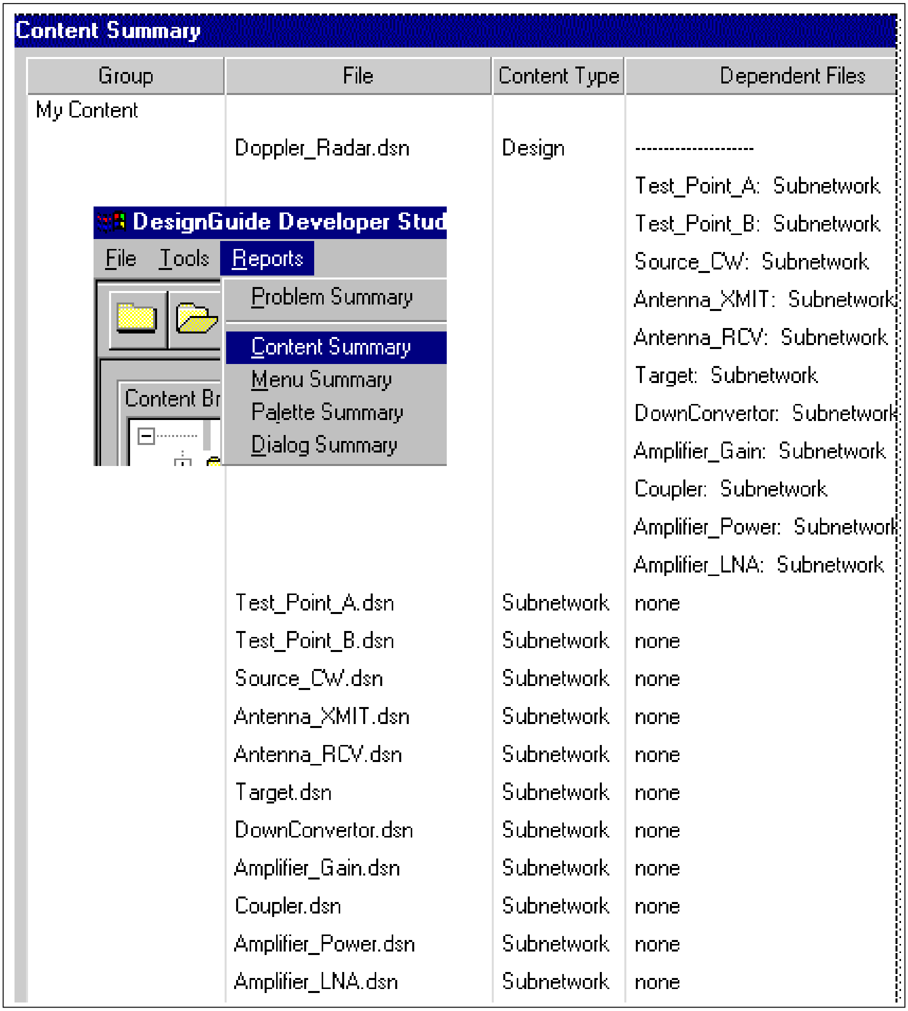
Viewing the Content Summary.

11. From the DesignGuide Developer Studio Window, select **Reports** > **Content Summary** to see the content summary report.

    - In Display, add **Doppler_Radar.dds**.
    - In Dataset, add **Doppler_Radar.ds**.

    If the Design and Display and Dataset all have the same name, it will work by default. If they have different names, you must map the files as dependent.

12. Select **File** > **Save**.

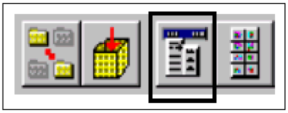


Content Summary

Group	File	Content Type	Dependent Files
My Content			
	Doppler_Radar.dsn	Design	--------------------
			Test_Point_A: Subnetwork
			Test_Point_B: Subnetwork
			Source_CW: Subnetwork
			Antenna_XMIT: Subnetwork
			Antenna_RCV: Subnetwork
			Target: Subnetwork
			DownConvertor: Subnetwork
			Amplifier_Gain: Subnetwork
			Coupler: Subnetwork
			Amplifier_Power: Subnetwork
			Amplifier_LNA: Subnetwork
	Test_Point_A.dsn	Subnetwork	none
	Test_Point_B.dsn	Subnetwork	none
	Source_CW.dsn	Subnetwork	none
	Antenna_XMIT.dsn	Subnetwork	none
	Antenna_RCV.dsn	Subnetwork	none
	Target.dsn	Subnetwork	none
	DownConvertor.dsn	Subnetwork	none
	Amplifier_Gain.dsn	Subnetwork	none
	Coupler.dsn	Subnetwork	none
	Amplifier_Power.dsn	Subnetwork	none
	Amplifier_LNA.dsn	Subnetwork	none

Using the Menu Editor. The steps that follow demonstrate use of the Menu Editor.

13. In the DesignGuide Developer Studio Window, open the Menu Editor.



14. In the Menu Editor window, the first step is to define whether you need to have DesignGuides added to the Schematic and/or Layout windows. For this example, we do not need DesignGuides in the Layout window
    - In the Menu Window Type, select **Layout**.
    - Select **Delete Menu.** The Menu Window Type should show **Schematic**. Keep this.
    - Set the Menu Name to **Radar**. Press **Enter** when you finish typing in the new name.

15. There are two alternatives for performing this step. The first one: From the Menu Editor window, define 20 menu items by choosing the **Insert Menu Item** button. Define the menu with the following names and structure. Use the arrow buttons to indent the menu names

Components

- Amplifiers
    - Gain Amp
    - LNA Amp
    - Power Amp
- Antennas
    - Receive Antenna
    - Transmit Antenna
- Coupler
- Down Converter
- CW Source
- Target
- Test Points
    - Test Point A
    - Test Point B

Component Palette

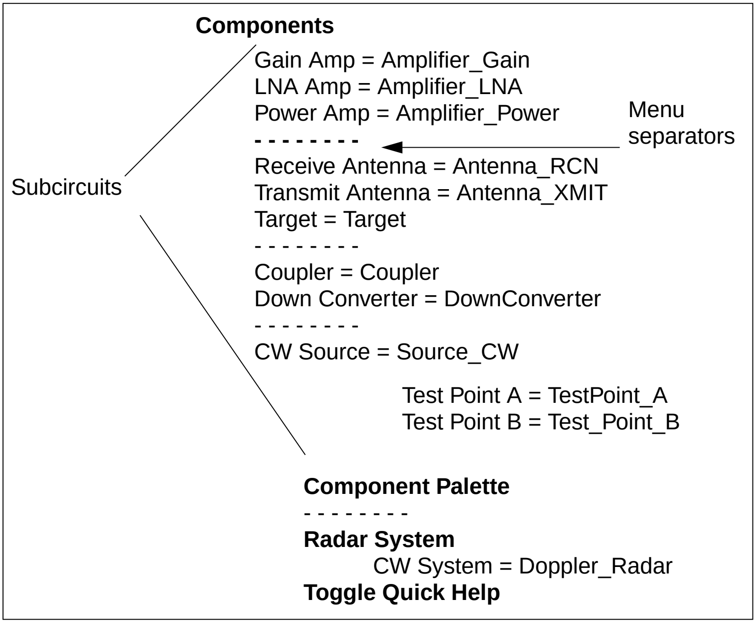
- - - - - - - -

Radar System

- CW System

Toggle Quick Help

The alternative action: In the Menu Editor window, define 19 menu items by choosing the **Insert Menu Item** button. Define the menu with the following names and structure. Use the arrow buttons to indent the menu names. This menu structure will be different from the remainder of the example, but has the benefit of only one submenu level.



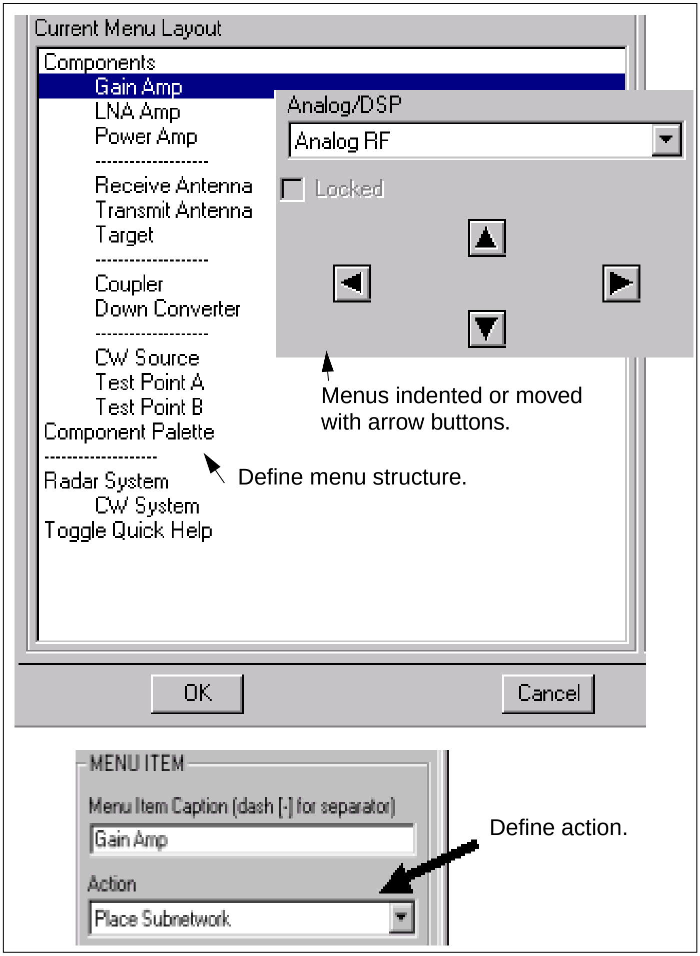
```
Components
    Gain Amp = Amplifier_Gain
    LNA Amp = Amplifier_LNA
    Power Amp = Amplifier_Power
    - - - - - - - -
    Receive Antenna = Antenna_RCN
    Transmit Antenna = Antenna_XMIT
    Target = Target
    - - - - - - - -
    Coupler = Coupler
    Down Converter = DownConverter
    - - - - - - - -
    CW Source = Source_CW
                Test Point A = TestPoint_A
                Test Point B = Test_Point_B

        Component Palette
        - - - - - - - -
        Radar System
              CW System = Doppler_Radar
        Toggle Quick Help
```

Table 9-1 and the illustration that follows summarize this process.

Table 9-1. Bitmaps and their Objects

Bitmap	Object
amp.bmp	Amplifer_Gain.dsn
lna.bmp	Amplifer_LNA.dsn
pa.bmp	Amplifer_Power.dsn
target.bmp	Target.dsn
xmt_ant.bmp	Antenna_XMIT.dsn
rcv_ant.bmp	Antenna_RCV.dsn
cw_source.bmp	Source_CW.dsn
coupler.bmp	Coupler.dsn
downconverter.bmp	DownConverter.dsn
TP_A.bmp	Test_Point_A.dsn
TP_B.bmp	Test_Point_B.dsn
TP_C.bmp	No subcircuit



Analog/DSP: Analog RF

Menu Item Caption (dash [-] for separator): Gain Amp

Action: Place Subnetwork

Menus indented or moved with arrow buttons.

Define menu structure.

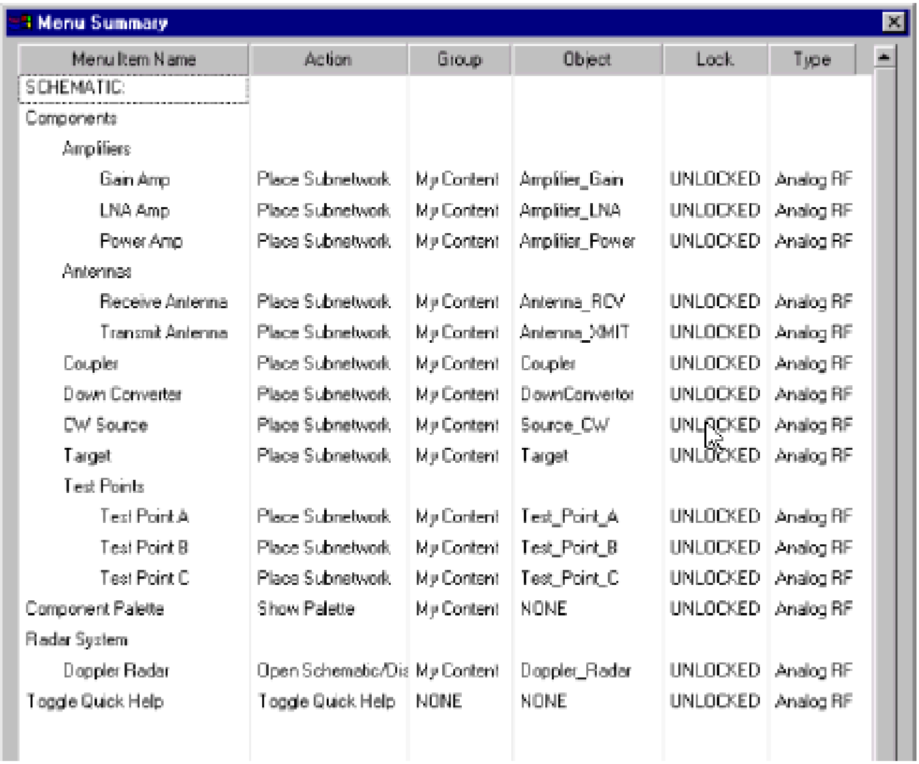
Define action.

Using the Define Menu. The steps that follow demonstrate the use of the Define menu.

16. From the Menu Editor window, for each menu selection, select the Action **Place Subnetwork** for each of the 12 subnetworks (using **Group** and **Object** selections for each object). Exceptions include *Component Palette*, *Doppler Radar* (open the schematic *Doppler_Radar*), and *Toggle Quick Help*.

    Note Don't map the Component Palette menu until after it is defined in step 22.

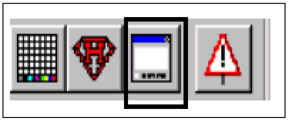
17. For the *Doppler Radar* menu option, select the Action **Open Schematic/Display** for the design object.
18. Map the menu name **Toggle Quick Help** to the *Toggle Quick Help* action.
19. From the DesignGuide Developer Studio window, view **Reports** > **Menu Summary** to check the menu definition summary.



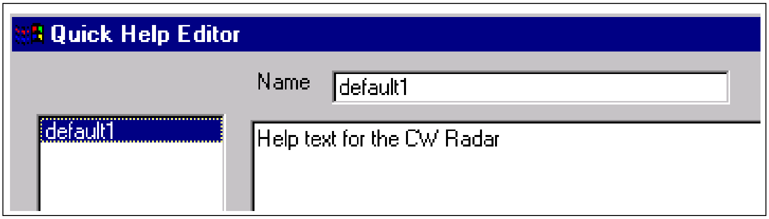
Menu Item Name	Action	Group	Object	Lock	Type
SCHEMATIC:					
Components					
Amplifiers					
Gain Amp	Place Subnetwork	My Content	Amplifier_Gain	UNLOCKED	Analog RF
LNA Amp	Place Subnetwork	My Content	Amplifier_LNA	UNLOCKED	Analog RF
Power Amp	Place Subnetwork	My Content	Amplifier_Power	UNLOCKED	Analog RF
Antennas					
Receive Antenna	Place Subnetwork	My Content	Antenna_RCV	UNLOCKED	Analog RF
Transmit Antenna	Place Subnetwork	My Content	Antenna_XMIT	UNLOCKED	Analog RF
Coupler	Place Subnetwork	My Content	Coupler	UNLOCKED	Analog RF
Down Converter	Place Subnetwork	My Content	DownConvertor	UNLOCKED	Analog RF
CW Source	Place Subnetwork	My Content	Source_CW	UNLOCKED	Analog RF
Target	Place Subnetwork	My Content	Target	UNLOCKED	Analog RF
Test Points					
Test Point A	Place Subnetwork	My Content	Test_Point_A	UNLOCKED	Analog RF
Test Point B	Place Subnetwork	My Content	Test_Point_B	UNLOCKED	Analog RF
Test Point C	Place Subnetwork	My Content	Test_Point_C	UNLOCKED	Analog RF
Component Palette	Show Palette	My Content	NONE	UNLOCKED	Analog RF
Radar System					
Doppler Radar	Open Schematic/Dis	My Content	Doppler_Radar	UNLOCKED	Analog RF
Toggle Quick Help	Toggle Quick Help	NONE	NONE	UNLOCKED	Analog RF

Using the Quick Help Editor. The steps that follow demonstrate the use of the Quick Help Editor.

20. From the DesignGuide Developer Studio window, open the Quick Help Editor window from the toolbar.



21. Define a Quick Help entry for the Radar System examples. This is explanatory text to help the user, which can be assigned to a menu. This is done by selecting **add**, then changing the name and title, and adding descriptive text. For now, type some arbitrary text for the Quick Help.



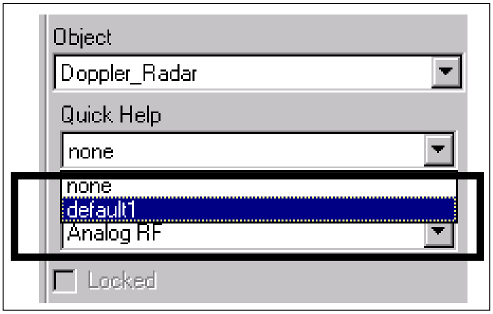
Quick Help Editor

Name default1

default1

Help text for the CW Radar

22. Map the Quick Help to the Menu. From the Menu Editor window, assign the Quick Help for the CW Radar System design that was defined in Step 21 to the Doppler Radar menu selection.



Object

Doppler_Radar

Quick Help

none

none

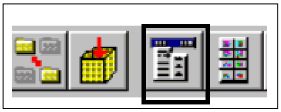
default1

Analog RF

□ Locked

Using the Palette Editor. The subnetworks for the radar DesignGuide will be added to a palette.

23. From the DesignGuide Developer Studio window, open the **Palette Editor**.



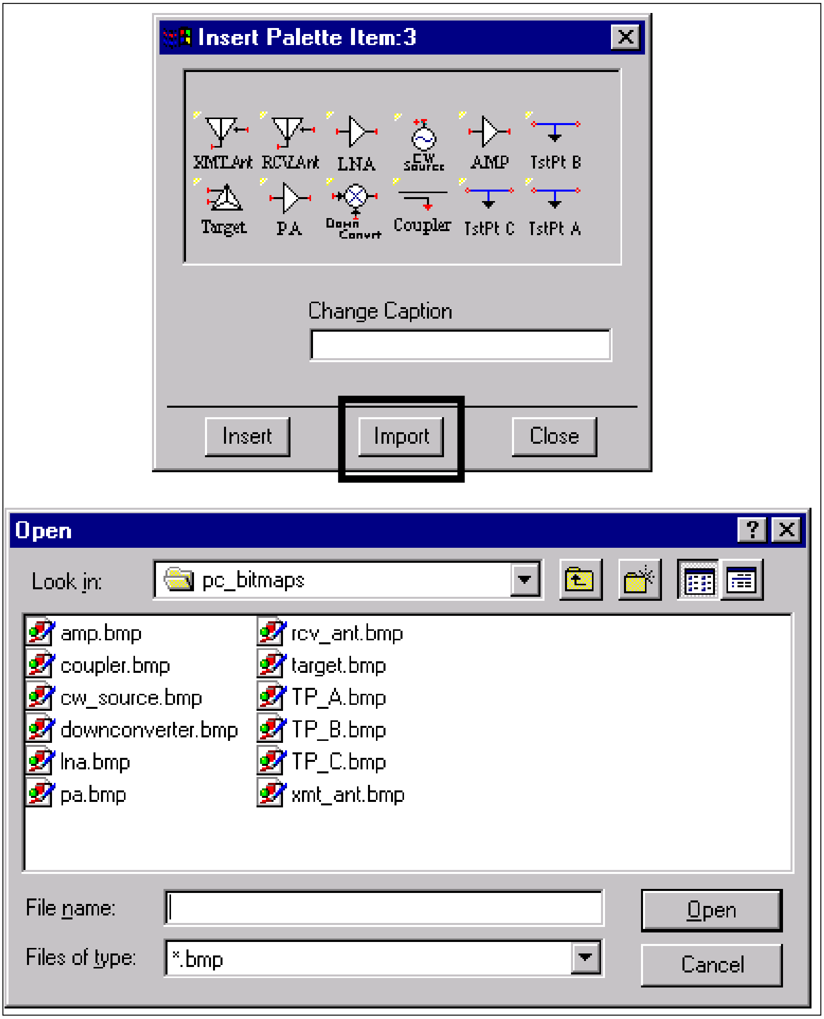
24. From the Palette Editor, select the **Insert Palette Item** button.

25. From the Insert Palette Item dialog, use the **Import** button and navigate to the pre-defined icons located at *$HOME/dg_radar_prj/ui/pc_bitmaps.*The bitmaps were predefined with the Bitmap Editor, and which is not shown as part of this lab.

Note if you are running the program on UNIX, there is not an **Import** button. You will have to copy the bitmaps from *$HOME/dg_radar_prj/ui/unix_bitmaps* to *$HOME/studio_files/Radar/bitmaps/palette*. For this example, these bitmaps were converted from PC to UNIX using the Bitmap Editor.

---

26. Select all of the icons (one at a time). As you select them, they will display in the Insert Palette Item Dialog window.



Look in: pc_bitmaps

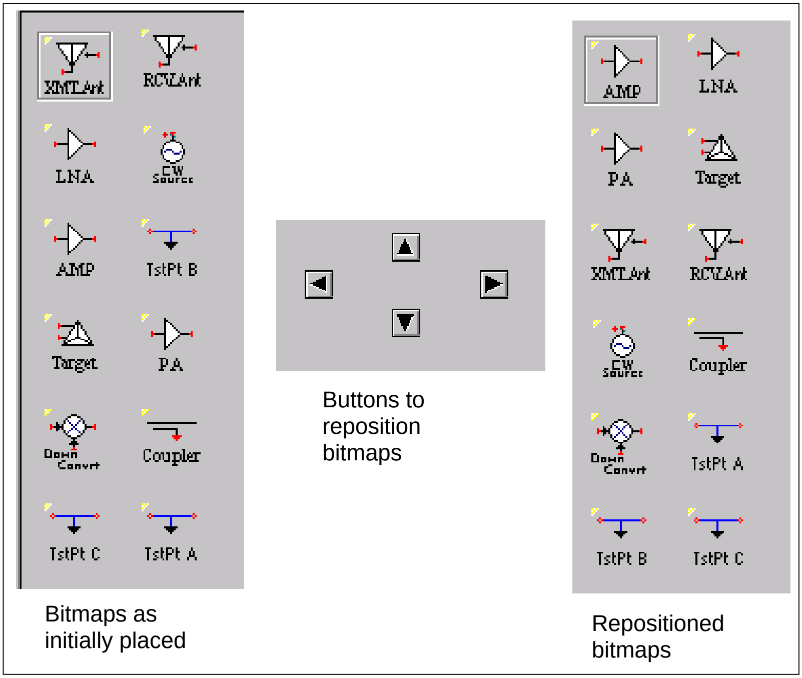
amp.bmp
coupler.bmp
cw_source.bmp
downconverter.bmp
lna.bmp
pa.bmp
rcv_ant.bmp
target.bmp
TP_A.bmp
TP_B.bmp
TP_C.bmp
xmt_ant.bmp

File name:

Files of type: *.bmp

27. From the Bitmap dialog box,use the **Insert** button to place the palette bitmaps.

28. After inserting the bitmaps, if the bitmaps are not in a preferred order, use the arrow buttons to re-position the icons.

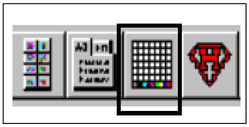


29. Associate an action to each icon. In each case, the action will be **Place Subnetwork**. (Refer to the sections that follow for menu subnetwork mapping definition.)

30. Change the palette name to *radar_palette*. At any time, choose the **Test** button to review the functionality of the palette.

Using the Bitmap Editor. The steps that follow demonstrate the use of the Bitmap Editor.

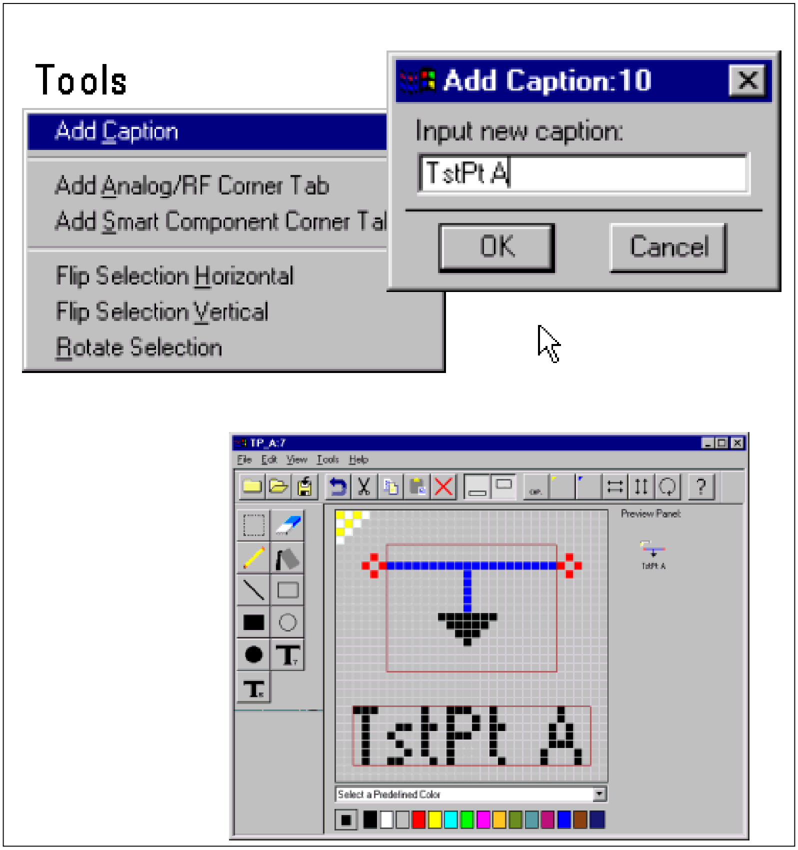
31. From the DesignGuide Developer Studio window, or from the Palette Editor, open the Bitmap Editor.



32. As an exploration, use the bitmap edit to change one of the icons (such as Test_Point_A, or Target) and change the color, or make entirely new bitmaps. In the bitmap editor window, explore the following commands:

    - *View >Caption Guide Display*
    - *View > Image Guide Display*
    - *Tools > Add Caption*
    - *Tools > Add Analog/RF Corner Tab*
    - *File > Save As*

Note If you are running the program under UNIX, re-start ADS to view changes to bitmaps in the Palette Editor.

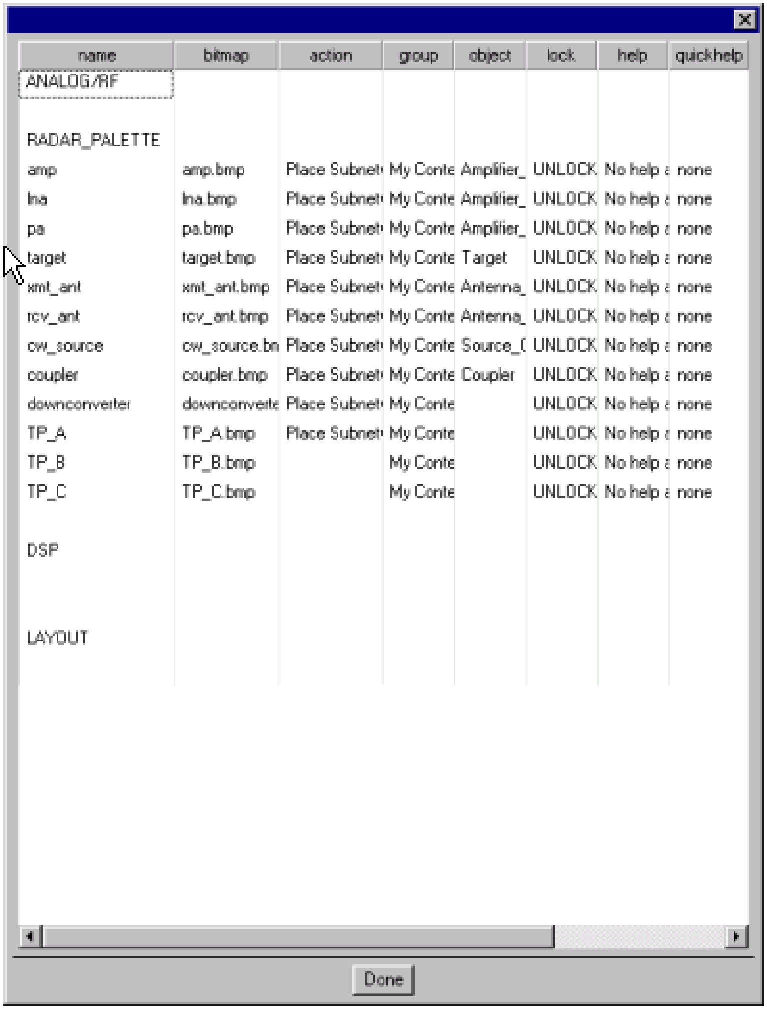


Input new caption: TstPt A

33. From the Menu Editor window, associate the Show Palette action to the Component Palette menu. Use the pull-down menu to select the newly created Radar palette object.

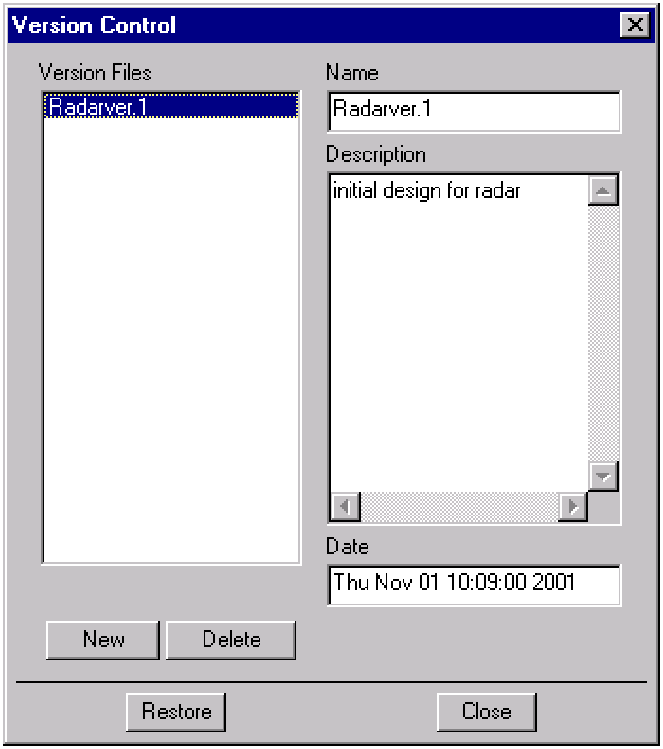
Viewing the Palette Editor Report Summary. The following steps provide an example of the use of the Palette Editor Report Summary.

34. From the DesignGuide Developer Studio window, select **Reports** > **Palette Summary** to view the Palette definition summary (as shown here). It’s advisable to save the Studio project prior to packaging or building. Select **File** > **Save** or select the **Save** icon on the tool bar.



name	bitmap	action	group	object	lock	help	quickhelp
ANALOG/RF							
RADAR_PALETTE							
amp	amp.bmp	Place Subnet	My Conte	Amplifier_	UNLOCK	No help a	none
lna	lna.bmp	Place Subnet	My Conte	Amplifier_	UNLOCK	No help a	none
pa	pa.bmp	Place Subnet	My Conte	Amplifier_	UNLOCK	No help a	none
target	target.bmp	Place Subnet	My Conte	Target	UNLOCK	No help a	none
xmt_ant	xmt_ant.bmp	Place Subnet	My Conte	Antenna_	UNLOCK	No help a	none
rcv_ant	rcv_ant.bmp	Place Subnet	My Conte	Antenna_	UNLOCK	No help a	none
cw_source	cw_source.bn	Place Subnet	My Conte	Source_(	UNLOCK	No help a	none
coupler	coupler.bmp	Place Subnet	My Conte	Coupler	UNLOCK	No help a	none
downconverter	downconverte	Place Subnet	My Conte		UNLOCK	No help a	none
TP_A	TP_A.bmp	Place Subnet	My Conte		UNLOCK	No help a	none
TP_B	TP_B.bmp		My Conte		UNLOCK	No help a	none
TP_C	TP_C.bmp		My Conte		UNLOCK	No help a	none
DSP							
LAYOUT							

Optional step: From the DesignGuide Developer Studio window, define the revision information by selecting **File** > **Version**. The Version Control dialog box is shown here.



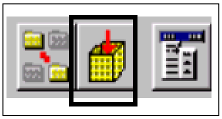
Version Files: Radarver.1

Name: Radarver.1

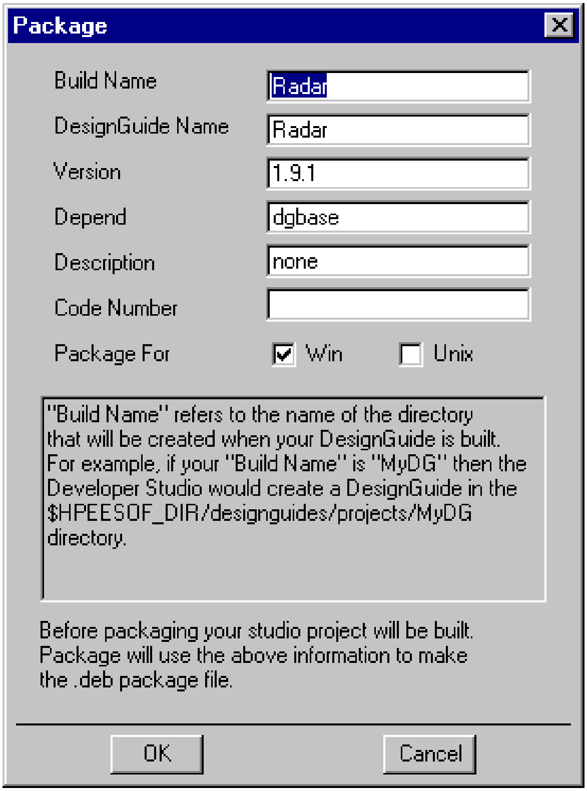
Description: initial design for radar

Date: Thu Nov 01 10:09:00 2001

35. Package the DesignGuide. Select the Package Studio Product toolbar icon.



The Package dialog box is shown here.



Build Name: Radar

DesignGuide Name: Radar

Version: 1.9.1

Depend: dgbase

Description: none

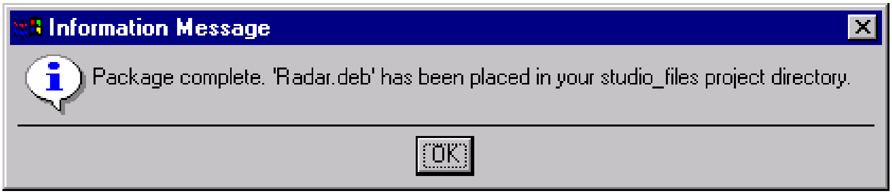
Code Number:

Package For: ☑ Win ☐ Unix

"Build Name" refers to the name of the directory that will be created when your DesignGuide is built. For example, if your "Build Name" is "MyDG" then the Developer Studio would create a DesignGuide in the $HPEESOF_DIR/designguides/projects/MyDG directory.

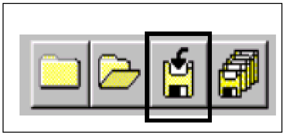
Before packaging your studio project will be built. Package will use the above information to make the .deb package file.

A confirmation message appears when packaging is complete.

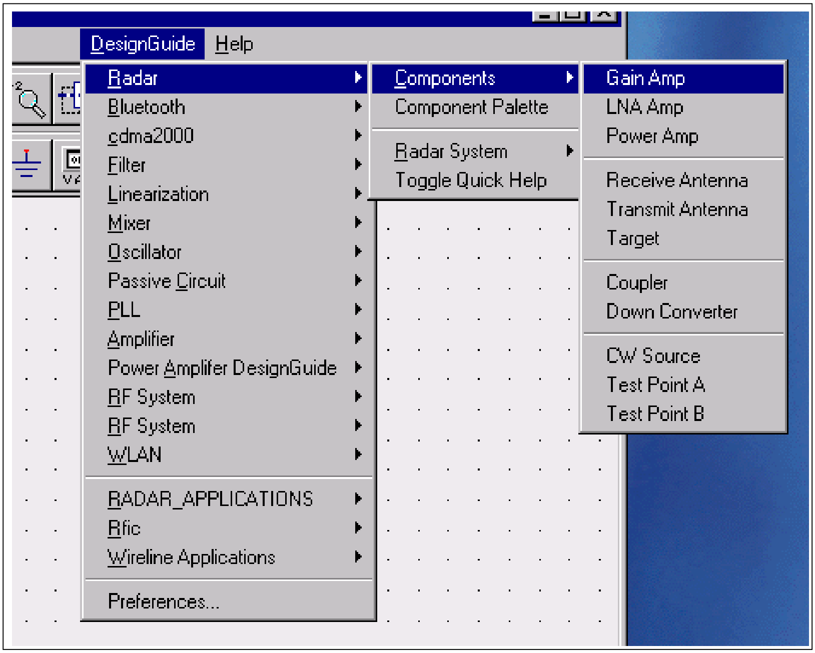


Package complete. 'Radar.deb' has been placed in your studio_files project directory.

36. Save the DesignGuide file. (It was also saved in Step 35, but this will save any settings made for Packaging.)



37. Check the installation by restarting ADS, opening a new project, and checking the DesignGuide menu.



Creating Palettes for Schematic and Layout Windows

This section uses the example project *dg_rfic_models_prj* (RFIC Model DesignGuide) to provide a brief summary of the process of creating palettes for Schematic and Layout windows. The example uses a collection of related subnetworks that pertain to RFIC. Several of the components have both an electrical model and a layout representation., either static footprint or parameterized (GCC). The example is not intended to show how the content of the subnetworks was prepared.

This example does the following:

- Incorporates a previously created collection or library of ADS models
- Shows how to link to models that use GCC for Layout
- Creates a palette for Schematic and Layout windows
- Shows the use of AEL as an action

Note Though this example is designed to demonstrate how to implement a library of subcircuits, it's not a library in the usual sense of the word. When a DesignGuide subcircuit is inserted into the schematic, a copy is made of the original subcircuit and placed in the current ADS project, which becomes disassociated from the original subcircuit. Unlike a real library model, which would be updated if the original version is updated, it does not update the copies contained in the ADS projects.

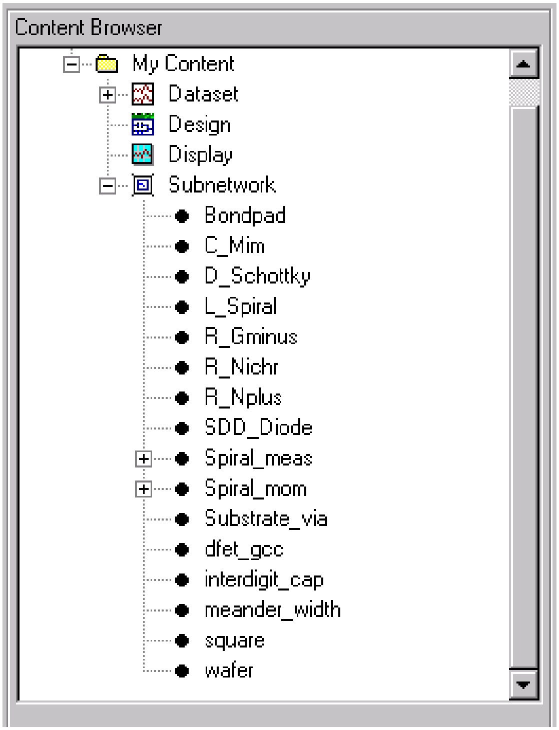
Table and the illustration that follow show an overview of the models.

Table 9-2. Designs and their Electrical Models

Design Name	Electrical Model	Physical Layout
Bondpad	equivalent ckt	n/a
C_Mlm	equivalent ckt	n/a
D_Schottky	equivalent ckt	n/a
L_Spiral	equivalent ckt	GCC model
R_Gminus	equivalent ckt	n/a
R_Nlchr	equivalent ckt	n/a

Table 9-2. Designs and their Electrical Models (continued)

Design Name	Electrical Model	Physical Layout
R_Nplus	equivalent ckt	n/a
SDD_Diode	SDD model	n/a
Spiral	Momentum data	static layout
Spiral_meas	Measured data	n/a
Substrate_via	equivalent ckt	static layout
dfet_gcc	equivalent ckt	GCC model
square	n/a	static layout
wafer	n/a	GCC model


Content Browser

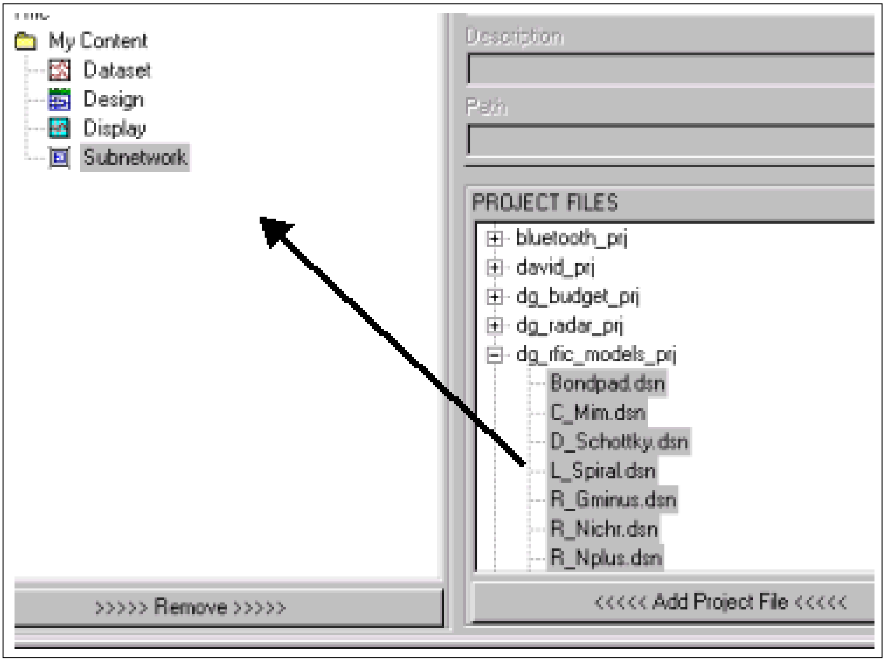
- My Content
  - Dataset
  - Design
  - Display
  - Subnetwork
    - Bondpad
    - C_Mim
    - D_Schottky
    - L_Spiral
    - R_Gminus
    - R_Nichr
    - R_Nplus
    - SDD_Diode
    - Spiral_meas
    - Spiral_mom
    - Substrate_via
    - dfet_gcc
    - interdigit_cap
    - meander_width
    - square
    - wafer

1. Make sure the *dg_rfic_models_prj* file is placed in the $HOME directory.

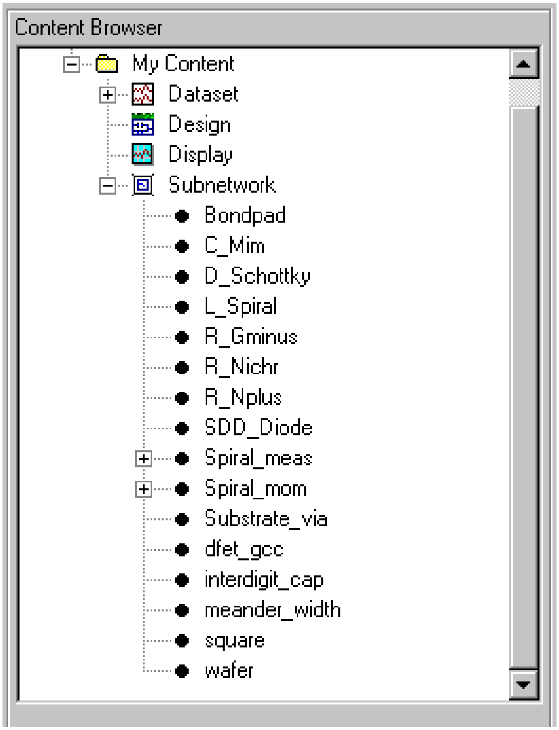
2. From the ADS Main window, select **DesignGuide** > **DesignGuide Developer Studio** > **Start DesignGuide Studio**.
3. From the DesignGuide Developer Studio window, select **File** > **New.**
4. From the New Project dialog, define the name of the project as **Rfic,** then select the **Create Project** button.

Mapping the Subnetworks

5. Select the subnetworks icon in the Content Browser and map all of the design files from the *dg_rfic_models_pr*j directory.



Following is a list of the mapped subnetworks.

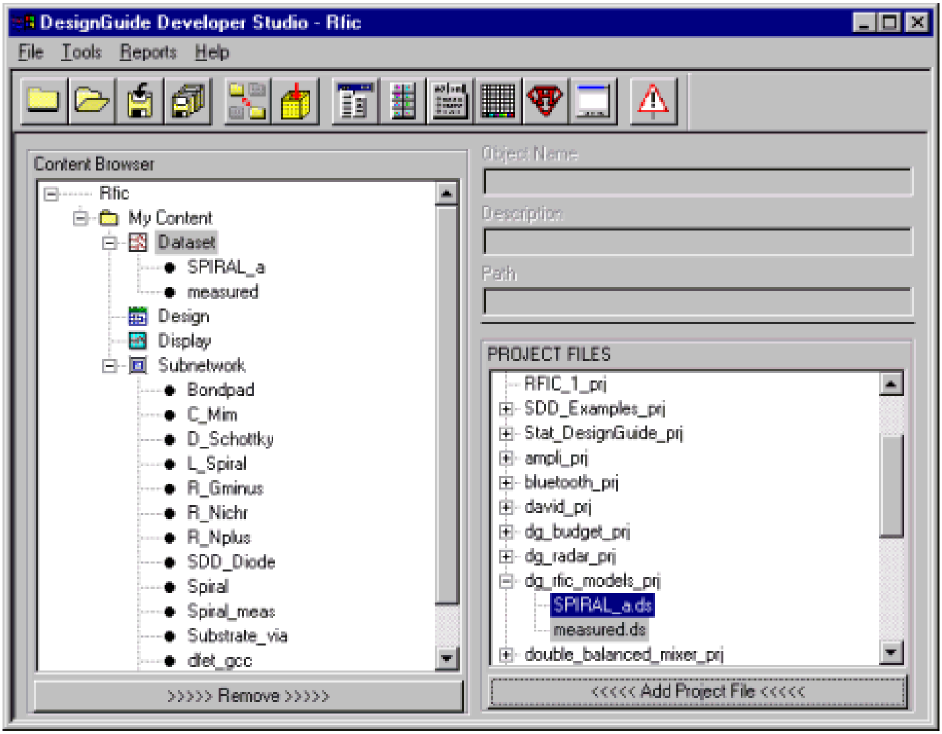


Content Browser

- My Content
  - Dataset
  - Design
  - Display
  - Subnetwork
    - Bondpad
    - C_Mim
    - D_Schottky
    - L_Spiral
    - R_Gminus
    - R_Nichr
    - R_Nplus
    - SDD_Diode
    - Spiral_meas
    - Spiral_mom
    - Substrate_via
    - dfet_gcc
    - interdigit_cap
    - meander_width
    - square
    - wafer

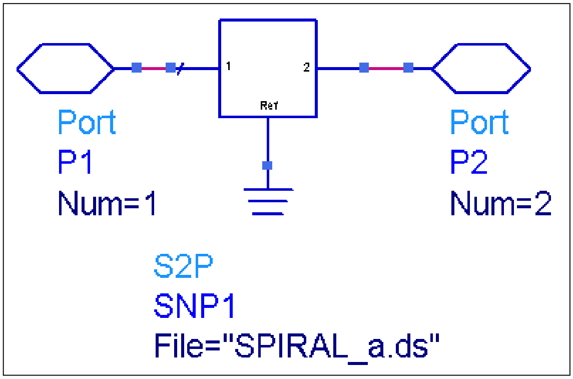
Mapping the Dataset Files

6. From the Content Editor, map the dataset files. The two datasets are from a Momentum simulation of a spiral inductor (*Spiral_a.ds*), and from an NWA measurement of a spiral inductor (*measured.ds*).

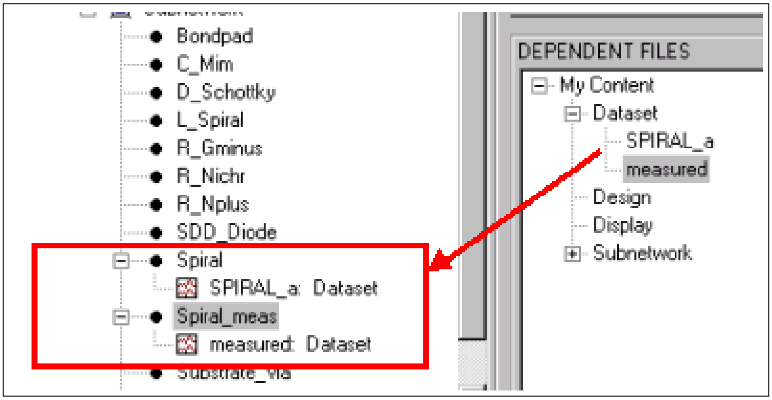


Using Data-based Models

7. From the Content Editor window, the *spiral_mom.dsn* subnetwork is a data-based model that uses the S-parameter results from a Momentum simulation.



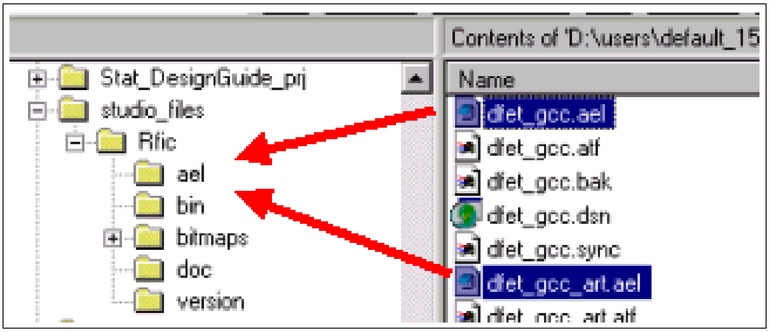
Select **spiral_mom** in the Content Browser, and select **SPIRAL_a** from the dependent files, and choose **Add Dependent Files.**



8. The *Spiral meas* sub-network is a data-based model that uses the S-parameter results from a NWA measurement. Select **Spiral_meas** in the Content Browser, and select **measured** from the dependent files, and choose **Add Dependent Files**.

Using GCC Models

9. From the Content Editor Window: For any subnetwork that uses GCC for the layout, copy the two AEL files for the subnetwork from *$HOME/dg_rfic_models_prj/networks* directory to the *$HOME/studio_files/Rfic/ael* directory.



The following AEL files will be needed to support the subnetwork definition in a later step. Copy these files:

- *dfet_gcc.ael*
- *dfet_gcc_art.ael*
- *interdigit_cap.ael*
- *interdigit_cap_art.ael*
- *meander_width.ael*
- *meander_width_art.ael*
- *square.ael*
- *square_art.ael*
- *wafer.ael*
- *wafer_art.ael*

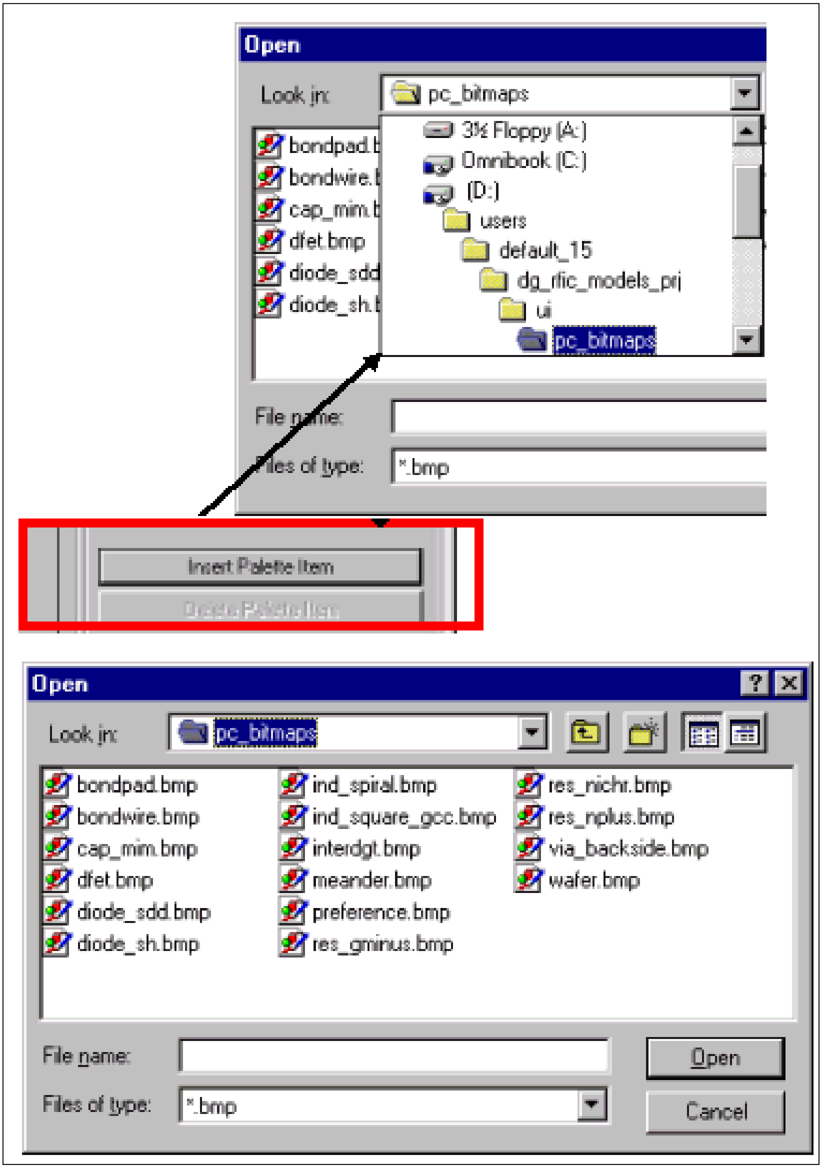
Using the Palette Editor

10. In the Palette Editor, select **Insert Palette Items**, then **Import**,

On UNIX systems, where there is no Import button, copy the files from the ADS project directory to *$HOME/studio_files/Rfic/bitmaps/palette*. On PCs, use the file browser to navigate to *$HOME/dg_rfic_models_prj/ui/pc_bitmaps* and select and import (one at a time) the following bitmaps located there:

- *bondpad.bmp*
- *bondwire.bmp*
- *cap_mim.bmp*
- *dfet.bmp*
- *diode_sdd.bmp*
- *diode_sh.bmp*
- *ind_spiral.bmp*
- *ind_square_gcc.bmp*
- *iinterdgt.bmp*
- *meander.bmp*
- *preference.bmp*
- *res_gminus.bmp*
- *res_nichr.bmp*
- *res_nplus.bmp*
- *via_backside.bmp*
- *wafer.bmp*

Note The bitmaps can be created using the Bitmap Editor, which is not shown as part of this example. You are encouraged to explore the Bitmap Editor.



Look in: pc_bitmaps

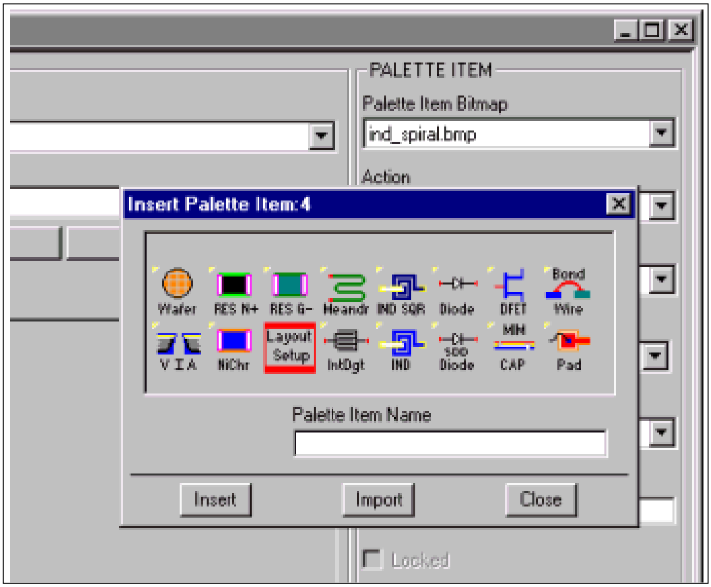
bondpad.bmp
bondwire.bmp
cap_mim.bmp
dfet.bmp
diode_sdd.bmp
diode_sh.bmp
ind_spiral.bmp
ind_square_gcc.bmp
interdgt.bmp
meander.bmp
preference.bmp
res_gminus.bmp
res_nichr.bmp
res_nplus.bmp
via_backside.bmp
wafer.bmp

File name:

Files of type: *.bmp

11. Place the following bitmaps into the Palette, and arrange them by using the arrow buttons.

    - *res_nichr.bmp (NiChr)*
    - *res_nplus.bmp (RES N+)*
    - *res_gminus.bmp (RES G-)*
    - *dfet.bmp (DFET)*
    - *diode_sh.bmp (Diode)*
    - *diode_sdd.bmp (SDD Diode)*
    - *via_backside.bmp (VIA)*
    - *bondpad.bmp (Pad)*
    - *cap_mim.bmp (MIM CAP)*
    - *ind_spiral.bmp (IND)*



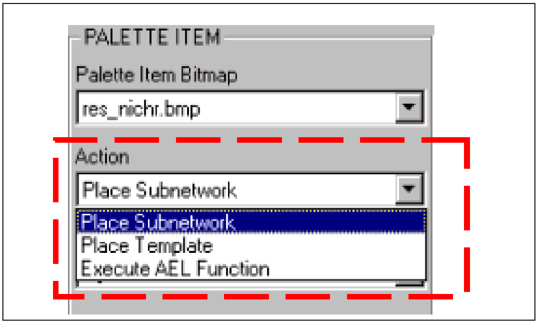
PALETTE ITEM

Palette Item Bitmap: ind_spiral.bmp

Insert Palette Item:4

12. In the Palette Editor, define the actions for each icon, which is **Place Subnetwor**k. The following is the mapping of bitmaps to objects:

    - *res_nichr.bmp (NiChr)-- R_Nichr*
    - *res_gminus.bmp (RES G-)-- R_Gminus*
    - *res_nplus.bmp (RES N+)-- R_Nplus*
    - *dfet.bmp (DFET)-- dfet_gcc*
    - *diode_sh.bmp (Diode)-- D_Schottky*
    - *diode_sdd.bmp (SDD Diode)-- SDD_Diode*
    - *via_backside.bmp (VIA)-- Substrate_via*
    - *bondpad.bmp (PAD)-- Bondpad*
    - *cap_mim.bmp (MIM CAP)-- C_Mim*
    - *ind_spiral.bmp (IND) -- L_Spiral*



PALETTE ITEM

Palette Item Bitmap

res_nichr.bmp

Action

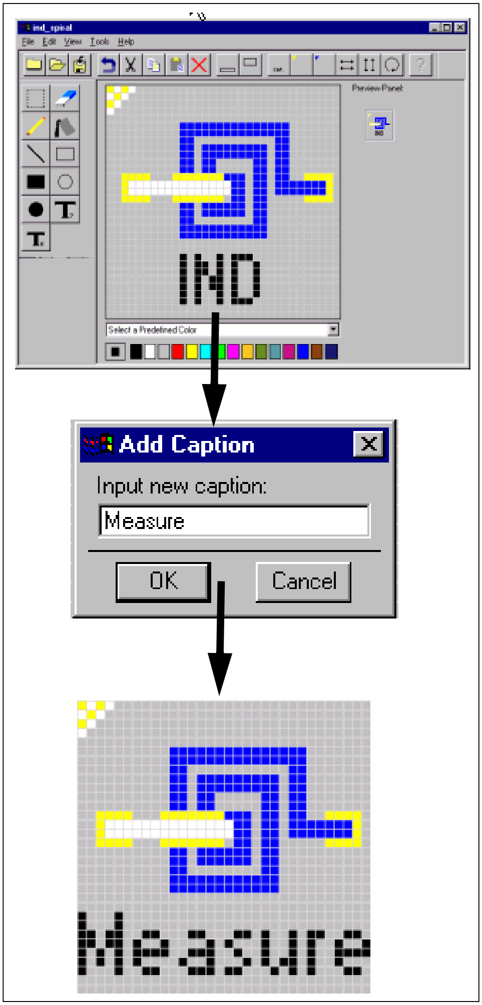
Place Subnetwork

Defining New Bitmaps

13. From the Palette Editor: We want to add two more inductors to the palette. This can be done in two ways: through the Bitmap Editor, or from the Insert Palette Item dialog (not available on UNIX systems). We will do this both ways to illustrate both processes. Select the inductor icon from the Palette Editor, and select the Edit Bitmap button. The icon is now shown in the bitmap editor.

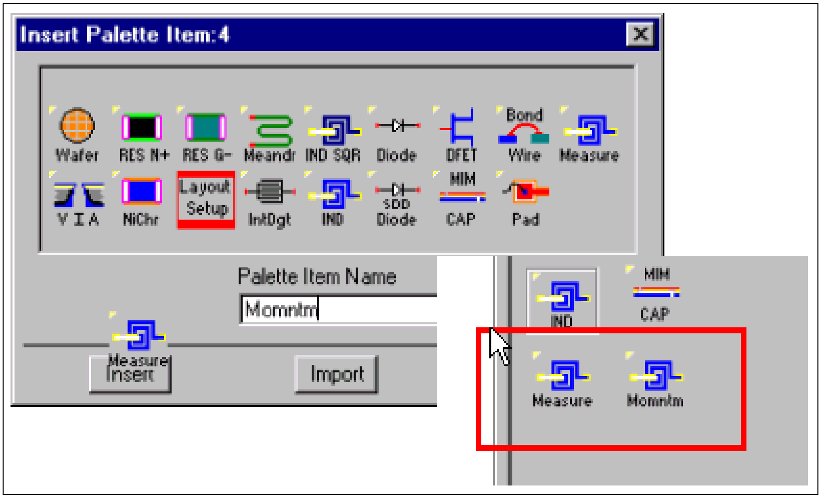
14. From the Bitmap Editor, change the caption by pressing the **Caption** button, or selecting the **Tools** > **Add Caption** command. Change the caption to **Measure.**

15. Save the bitmap under a new name **ind_meas.bmp** by selecting **File** > **Save as PC** (if you are using a PC; otherwise select **File** > **Save as UNIX**).



Input new caption: Measure

16. From the Palette Editor: If you are running the program under UNIX, repeat steps 13 and 14 with the caption name **Momntm**. If you are using the PC, select one of the existing inductor models in the palette. Select the **Insert Palette Item** button. Enter the new name in the Palette Item Name (**Momntm**), and choose the **Insert** button. This will change the name on the bitmap and create a new bitmap file.



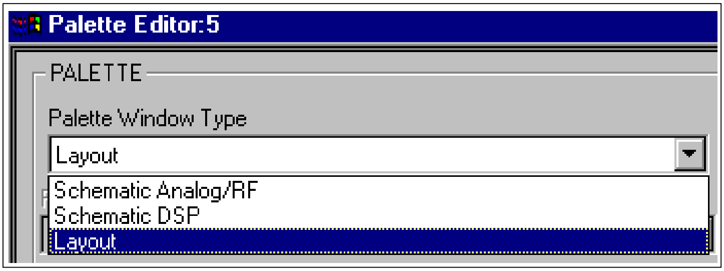
Palette Item Name: Momntm

Using the Layout Palette

Note On UNIX systems, if a change is made to a bitmap and saved with the same name, the changes will not show up in the palette until you shut down and restart ADS.

---

17. In the Palette Editor, change to a new palette for the Layout Window by selecting **Layout** from the pull-down menu under *Palette Window Type*.



Palette Editor:5

PALETTE

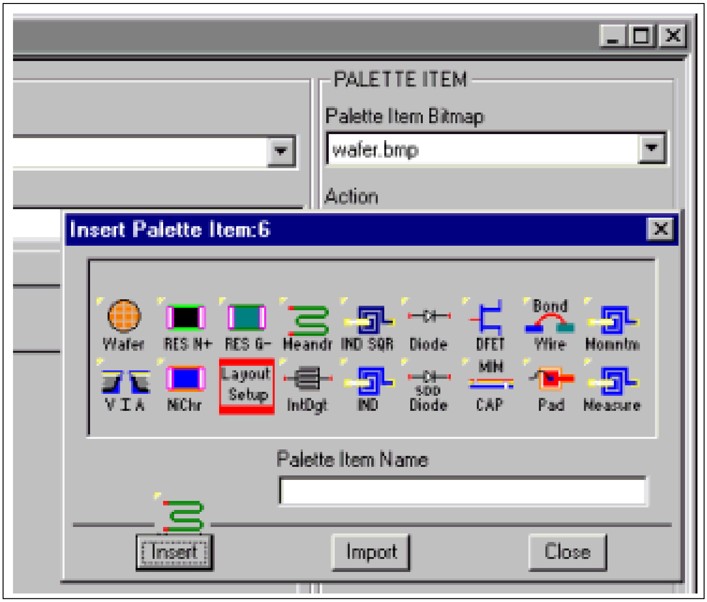
Palette Window Type

Layout

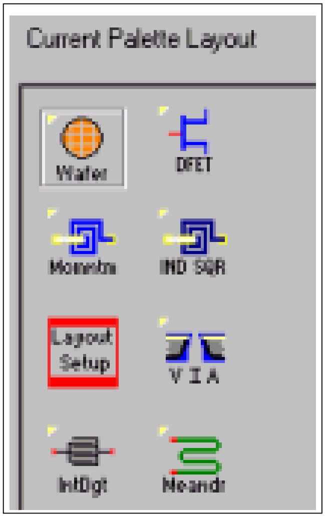
18. Select **Create Palette** and change the name of the palette to **RFIC_Layout.**

19. Add the following icons:

    - *Wafer.bmp* for *wafer.dsn*. (Wafer has no electrical model, but has a GCC-based layout.)
    - *dfet.bmp* for *dfet_gcc.dsn*. (DFET has an electrical equivalent nonlinear circuit model and a GCC-based layout.)
    - *ind_mom.bmp* for *spiral_mom.dsn*. (This design has a data-based S-parameter model from Momentum and a static layout footprint.)
    - *Ind_square_gcc.bmp* for *square.dsn*. (IND SQR is a data-based model with a GCC-based layout.)
    - *via_backside.bmp* for *substrate_via.dsn*. (VIA is the substrate via that has an electrical equivalent linear circuit model and a static layout.)
    - *Interdgt.bmp* for *Interdigit_cap.dsn* (No electrical model, GCC-based layout)
    - *meander.bmp* for *meander_width.dsn*. (No electrical model, GCC-based layout)
    - *Layout Setup* will be a set of AEL commands, which in this case set up user preferences.



Shown here is the final RFIC Layout palette.

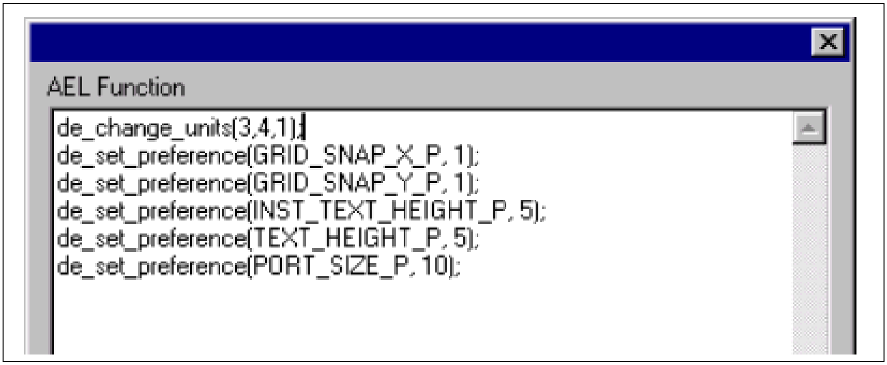


Defining AEL Actions

20. From the Palette Editor: For the *Layout Setup* bitmap, define the action as **Execute AEL Function**. Type in the following AEL commands.

    - *de_change_units(3,4,1);*
    - *de_set_preference(GRID_SNAP_X_P,1);*
    - *de_set_preference(GRID_SNAP_Y_P,1);*
    - *de_set_preference(INST_TEXT_HEIGHT_P,5);*
    - *de_set_preference(TEXT_HEIGHT_P,5);*
    - *de_set_preference(PORT_SIZE_P,10);*

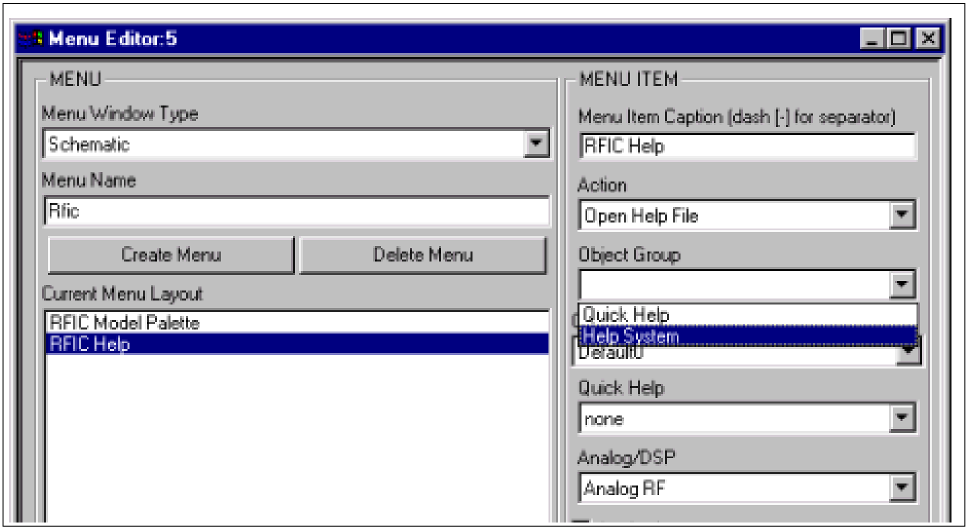
    These option settings are usually set by selecting **Option**> **Preferences** in a Layout window, and these AEL commands can be viewed from the ADS Main window by selecting **Options** > **Command Line.**



```
de_change_units(3,4,1);
de_set_preference(GRID_SNAP_X_P, 1);
de_set_preference(GRID_SNAP_Y_P, 1);
de_set_preference(INST_TEXT_HEIGHT_P, 5);
de_set_preference(TEXT_HEIGHT_P, 5);
de_set_preference(PORT_SIZE_P, 10);
```

Define the Menu for Schematic

21. From the Menu Editor, define a menu for the Schematic window. The first menu item is named *RFIC Model Palette*. The action is *Show Palette*. The Object Group is *My Content* and the Object is *RFIC Models*.



Menu Window Type: Schematic

Menu Name: Rfic

Current Menu Layout:
RFIC Model Palette
RFIC Help

Menu Item Caption (dash [-] for separator): RFIC Help

Action: Open Help File

Object Group:

DefaultU

Quick Help: none

Analog/DSP: Analog RF

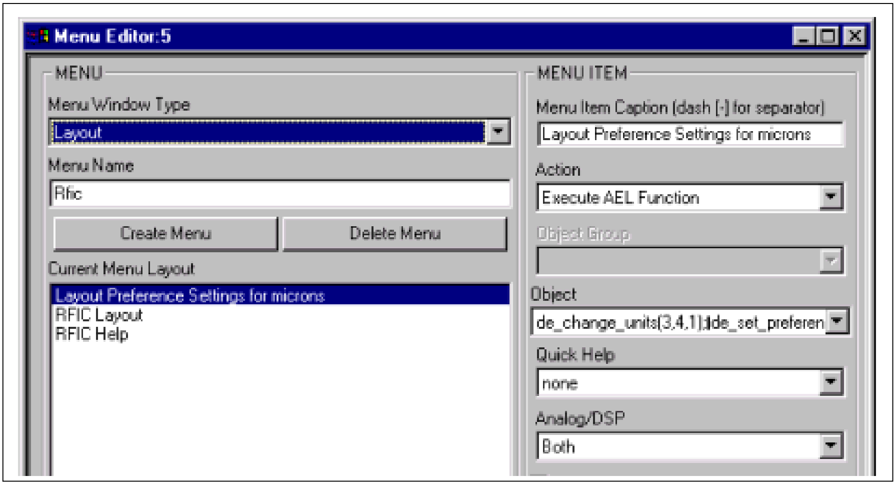
Defining the Menu for Layout

22. From the Menu Editor, change the Menu Window Type to **Layout**. Define the following menus:

    - The first menu item is named *Set Layout Preference*. The action is to execute the same AEL commands shown in step 20. These commands must be copied and pasted into the AEL Editor window. This serves as a menu alternative to the palette.

- The second menu item is named *RFIC Layout*. The action is to display the Palette.The Object Group is *My Content*.The object is *Rfic_layout*.

Note RFIC Help is not included in this example, but could be set up as a Quick Help from the Quick Help Editor o ra System Help HTML file.



Menu Editor:5

MENU

Menu Window Type
Layout

Menu Name
Rfic

Current Menu Layout
Layout Preference Settings for microns
RFIC Layout
RFIC Help

MENU ITEM

Menu Item Caption (dash [-] for separator)
Layout Preference Settings for microns

Action
Execute AEL Function

Object Group

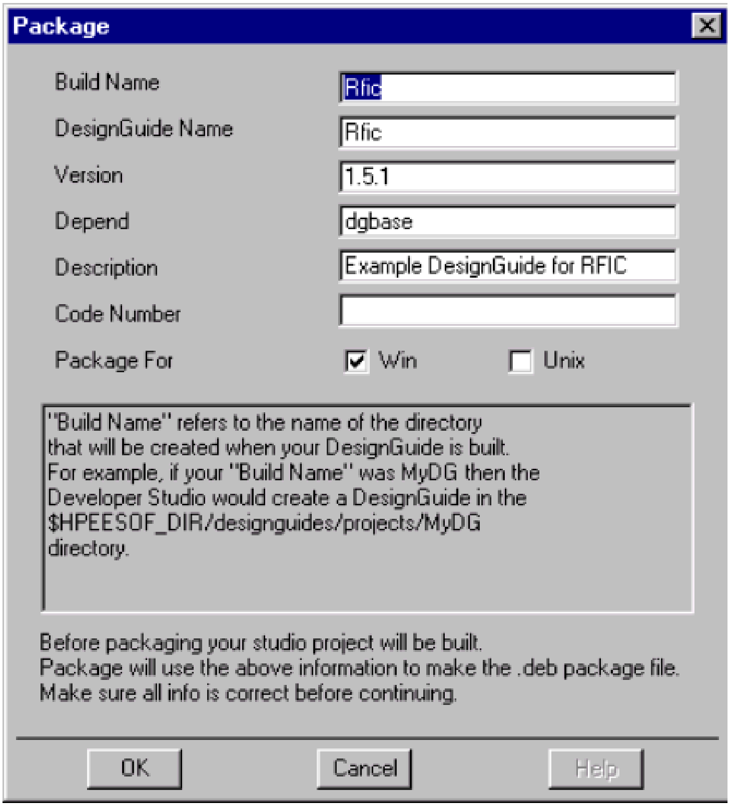
Object
de_change_units(3,4,1);de_set_preferen

Quick Help
none

Analog/DSP
Both

Packaging the RFIC DesignGuide

23. Close all of the Editor windows prior to saving the project.
24. From the Main Window, save the project.
25. Build and/or package this DesignGuide.



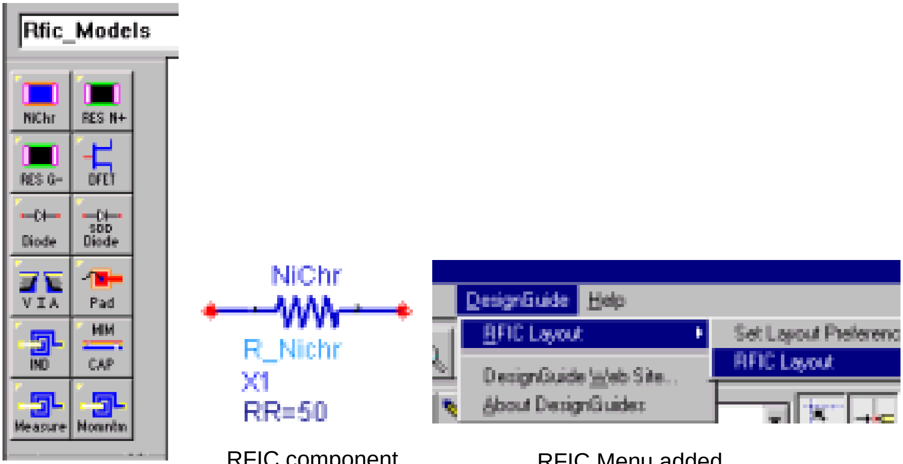
Build Name: Rfic
DesignGuide Name: Rfic
Version: 1.5.1
Depend: dgbase
Description: Example DesignGuide for RFIC
Code Number:
Package For: ☑ Win ☐ Unix

"Build Name" refers to the name of the directory that will be created when your DesignGuide is built. For example, if your "Build Name" was MyDG then the Developer Studio would create a DesignGuide in the $HPEESOF_DIR/designguides/projects/MyDG directory.

Before packaging your studio project will be built.
Package will use the above information to make the .deb package file.
Make sure all info is correct before continuing.

26. After building or packaging, save the project again.
27. Exit and re-start ADS. Start a new ADS project and test the RFIC DesignGuide from the Schematic and Layout windows.

    Shown here are the additions made to the Schematic window.

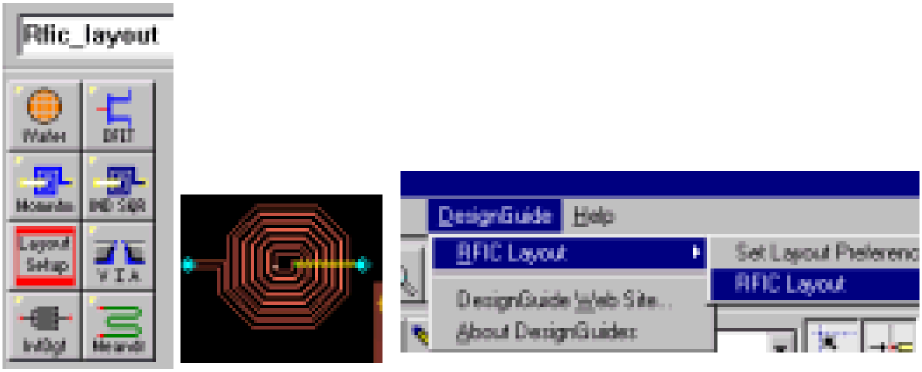


RFIC Component Palette

RFIC component placed in schematic.

RFIC Menu added

Shown here are the additions made to the Layout window.



RFIC Layout Palette

RFIC component placed in Layout

RFIC Menu added

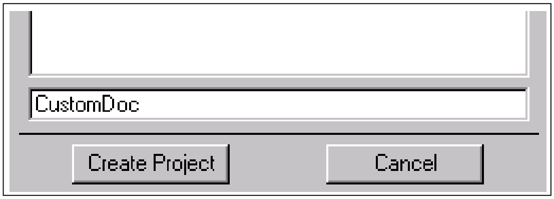
Linking an HTML Document to a Menu

This section demonstrates how to:

- Link to HTML documentation
- Add a menu to the ADS Main window

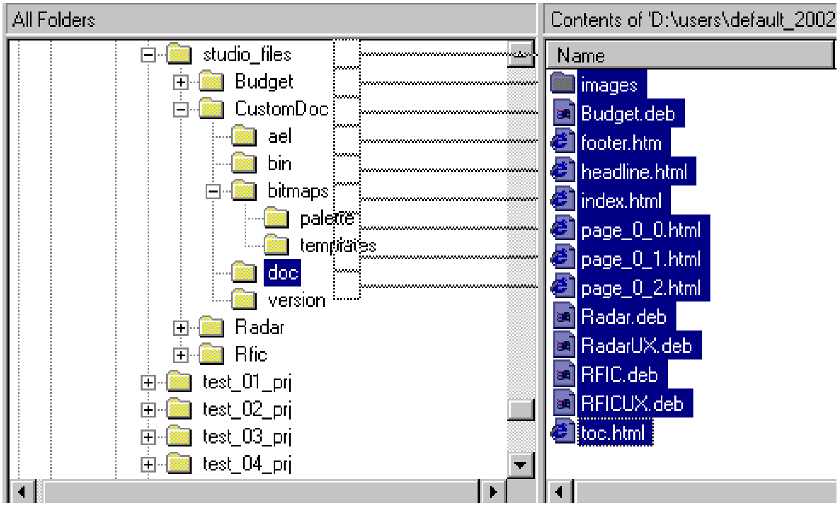
Note If you have already installed the tutorial for DesignGuides in ADS, review this example to see which method was used to install this document, but change the name of the menu since it will overwrite the existing installation. Also, to perform this exercise, you will need to have the DesignGuide Developer Studio tutorial file installed in ADS.

1. From the Main Window, select **DesignGuide** > **DesignGuide Developer Studio** > **Start DesignGuide Studio**.
2. Select **File** > **New.**
3. From the New Project dialog, define the name of the project as **CustomDoc,** then select the **Create Project** button.



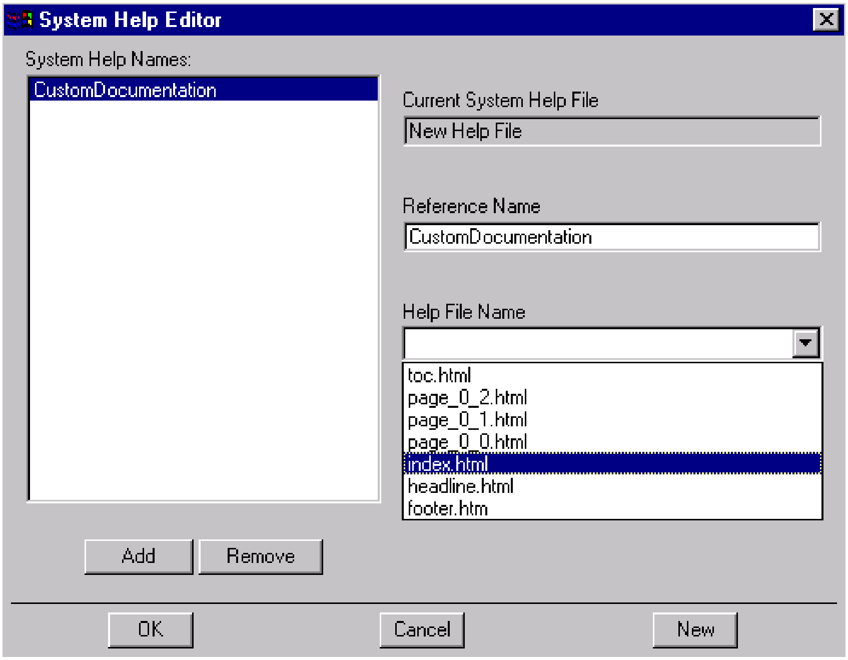
CustomDoc

Note When you create the content for an HTML document, make sure that it uses relative references, and all of the content is self-contained within a single directory structure. For this example, copy the entire HTML directory structure from *$HPEESOF_DIR/designguides/projects/tutorial/doc* to *$HOME/studio_files/Tutorial/doc*.



Defining the HTML Document

4. From the DesignGuide Developer Studio window, open the **System Help Editor**.
5. From the System Help Editor, select **Add**, then highlight the entry that now shows in the *System Help Names* window.
6. From the pull-down menu under *Help File Name*, select **index.html**. This is the top-level page for the HTML documentation. This can be created by any method for developing Web content.



System Help Names: CustomDocumentation

Current System Help File: New Help File

Reference Name: CustomDocumentation

Help File Name:
toc.html
page_0_2.html
page_0_1.html
page_0_0.html
index.html
headline.html
footer.htm

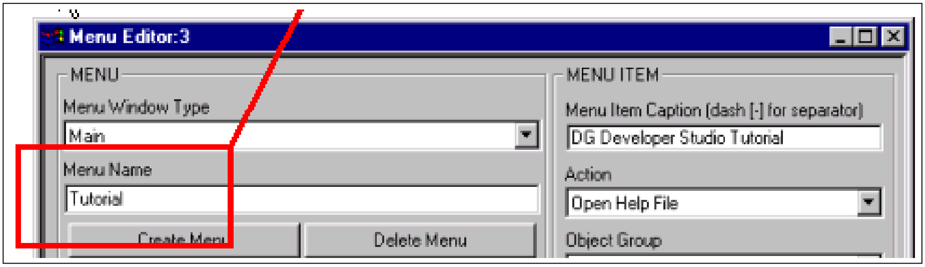
Defining a Menu for the ADS Main Window

7. From the Menu Editor, select the **Create Menu** button to create an entry for **Main** in *Menu Window Type*.

8. Delete the Schematic and Layout entries in *Menu Window Type* by making them active in turn and selecting the **Delete Menu** button.

9. Define the menu:

   - Menu Item Caption as **DG Developer Studio Tutorial**
   - Action as **Open Help File**
   - Object Group as **Help System**
   - Object as **Custom Documentation** (make sure to change **Default0** to a unique name)
   - Analog/DSP as **Both**



Menu Window Type: Main

Menu Name: Tutorial

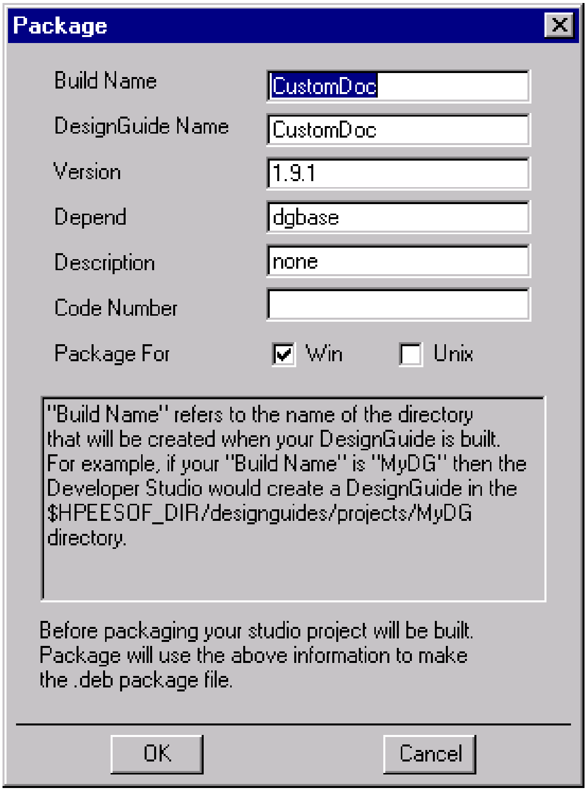
Menu Item Caption (dash [-] for separator): DG Developer Studio Tutorial

Action: Open Help File

Object Group

Packaging the Tutorial DesignGuide

10. From the Content Editor, save the project.
11. Select **Package Studio Project.**



Build Name: CustomDoc

DesignGuide Name: CustomDoc

Version: 1.9.1

Depend: dgbase

Description: none

Code Number:

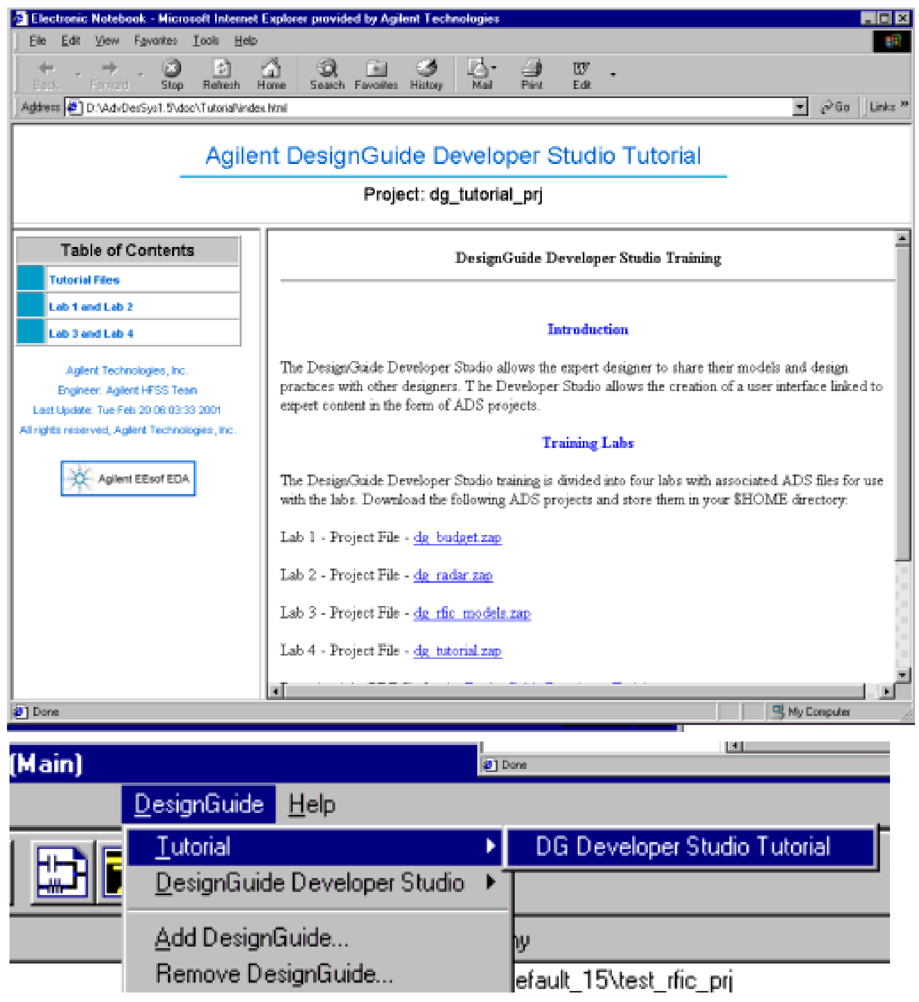
Package For ☑ Win □ Unix

"Build Name" refers to the name of the directory that will be created when your DesignGuide is built. For example, if your "Build Name" is "MyDG" then the Developer Studio would create a DesignGuide in the $HPEESOF_DIR/designguides/projects/MyDG directory.

Before packaging your studio project will be built. Package will use the above information to make the .deb package file.

12. Enter the package information, then choose **OK**.
13. Save the project again.
14. Exit ADS and re-start it

15. Check for the tutorial under the DesignGuide menu on the ADS Main window. It should start a web browser with the tutorial files and description.



Agilent DesignGuide Developer Studio Tutorial

Project: dg_tutorial_prj

Table of Contents

- Tutorial Files
- Lab 1 and Lab 2
- Lab 3 and Lab 4

Agilent Technologies, Inc.
Engineer: Agilent HFSS Team
Last Update: Tue Feb 20 06:03:33 2001
All rights reserved, Agilent Technologies, Inc.

DesignGuide Developer Studio Training

Introduction

The DesignGuide Developer Studio allows the expert designer to share their models and design practices with other designers. The Developer Studio allows the creation of a user interface linked to expert content in the form of ADS projects.

Training Labs

The DesignGuide Developer Studio training is divided into four labs with associated ADS files for use with the labs. Download the following ADS projects and store them in your $HOME directory.

Lab 1 - Project File - dg_budget.zap

Lab 2 - Project File - dg_radar.zap

Lab 3 - Project File - dg_rfic_models.zap

Lab 4 - Project File - dg_tutorial.zap

Electronic Notebook - Microsoft Internet Explorer provided by Agilent Technologies
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address D:\AdvDesSys1.5\doc\Tutorial\index.html
Go Links
Agilent EEsof EDA
Done
My Computer
(Main)
Done
DesignGuide Help
Tutorial
DG Developer Studio Tutorial
DesignGuide Developer Studio
Add DesignGuide...
Remove DesignGuide...
efault_15\test_rfic_prj

Chapter 10: DesignGuide Style Guide

The *DesignGuides Style Guide* describes how usability concepts will help optimize the interaction between users and Advanced Design System DesignGuides. Wherever possible, examples from DesignGuides will be used.

This document is written for developers of DesignGuides. It contains information that will help you create DesignGuides that fit within the ADS product environment. It provides specific information that will help you plan and make user interface decisions about your design. The information in this document is not platform-specific and can be applied to DesignGuides running in either Microsoft Windows or UNIX Motif operating systems.

Information for this document was obtained from DesignGuide developers, Agilent Comms EDA and Customer Usability Engineering personnel. A bibliography of additional usability sources is listed at the end of this chapter.

User-Centered Design Principles

The usability of DesignGuides is an essential ingredient to customers success. Time spent designing usability into DesignGuides will contribute to customers success in doing their jobs and the success of DesignGuides as a product.

Computer users have continually rising expectations about the usability of their software applications. They work in environments that demand better results in shorter periods of time. They complain about feature-laden software; they can’t find features they care about, and they can’t figure out how to use the features they find.

DesignGuides have been developed to allow users to focus on their work (Job 1 – making good design decisions) and not the tool (Job 2 –manipulating the ADS user interface) by eliminating many steps required to manually create schematics, set up and run simulations, and set up and view results.

Delivering simplicity does not mean just removing functionality. Simple, elegant user interfaces require work. Even a simple interface can require a significant investment in code. The benefits in taking the time to do it right include reduced training and support costs and happy, successful, productive and loyal customers.

Because great usability is one of the goals of DesignGuides, a review of the basic principles of user-centered design is appropriate. This section describes basic usability principles that should be considered when creating DesignGuides.

You will very likely find that you cannot design a product to meet every user-centered design principle. You will have to make decisions based on which principle or set of principles is most important in meeting your current design goal.

User in Control

Users should always feel in control of the DesignGuide. People learn best when they’re actively engaged. User interface objects are always in view while the user is physically manipulating the objects. When the user performs operations on an object, the results of those operations are immediately visible.

Create a balance between providing people with the capabilities they need to get their work done and preventing them from destroying data. If there will be grave consequences to actions they are about to take, they will want to know how to avoid this situation.

As people become more proficient with applications, they will want shortcuts to commands that required greater numbers of steps. Provide users with standard keyboard shortcuts as well as the ability to customize the user interface to meet their specific needs.

Consistency

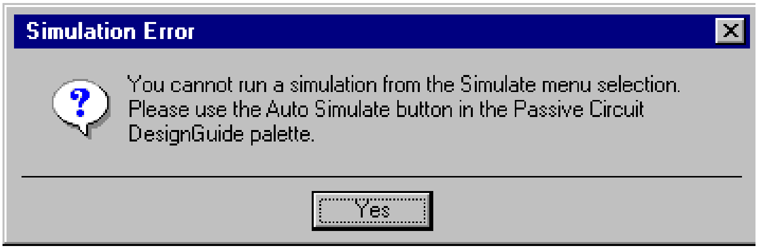
Consistency in the user interface allows users to transfer their knowledge and skills from one application to another. Use standard elements of Windows and Motif user interfaces to ensure consistency within your DesignGuide and to benefit from consistency across applications.

Effective software tools are consistent in several different ways. Consistency in the interface helps people learn and then easily recognize the graphic language of the interface. For example, once users know what a checkbox looks like, they don't have to learn another symbol when making choices. Consistency means users have to learn basic user interface behavior only once. They can spend their time on the task at hand, modifying a schematic, running simulations, and analyzing results.

Feedback and Dialog

Keep people informed about what's happening with DesignGuides. Provide feedback as they perform tasks and make the feedback as immediate as possible. When a user starts a process, provide a visual or audible indication that the command was received and the process has started. Provide as much information as possible about how long operations take. Tell the user how to get out of the current situation whenever possible. Provide direct, simple feedback they can understand. Messages should spell out what caused the error so that users can prevent the situation from happening again

.


Simulation Error

You cannot run a simulation from the Simulate menu selection. Please use the Auto Simulate button in the Passive Circuit DesignGuide palette.

Forgiveness

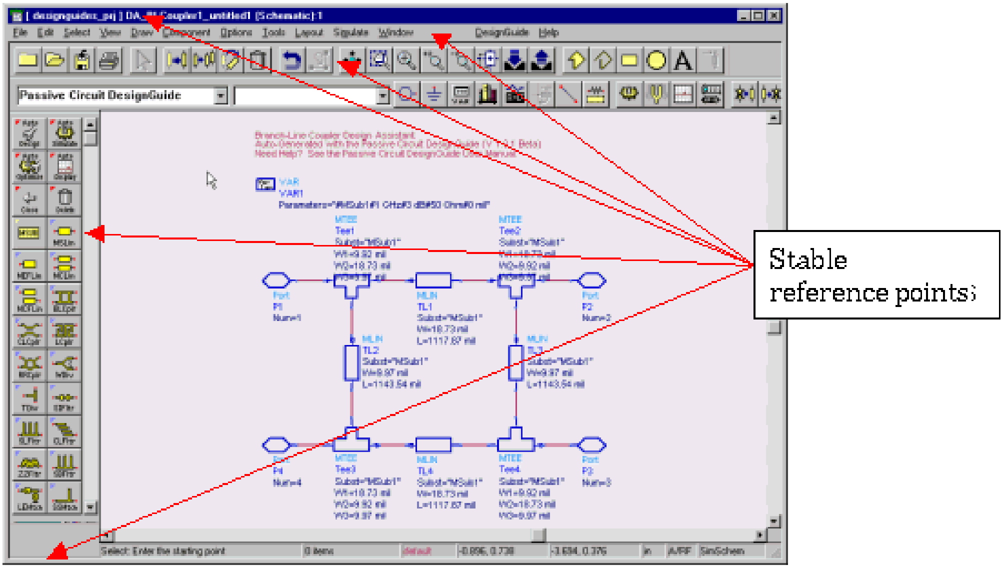
Encourage people to explore DesignGuides by building in forgiveness. Forgiveness means that actions on the computer are generally reversible. People need to feel that they can try things without damaging the system. Always support *Undo*.

Always warn people before they start a task that will cause irretrievable data loss. When options are clearly presented and feedback is appropriate and timely, learning how to use software should be relatively error-free. Frequent error messages are a good indication that something is wrong with the program design.

Stable
reference points

Stability

If people are to cope with the complexity of computers, they need stable reference points in the user interface. To give users a visual sense of stability, DesignGuides (and ADS) define a number of consistent graphic elements (menu bar, tool bars, palettes, schematic symbols, etc.) to maintain a sense of stability.



[designguides_prj] DA_BLCoupler1_untitled1 [Schematic]:1

Passive Circuit DesignGuide

Branch-Line Coupler Design Assistant
Auto-Generated with the Passive Circuit DesignGuide (V 1.0.1 Beta)
Need Help? See the Passive Circuit DesignGuide User Manual

VAR
VAR1
Parameters="W#Sub1#1 GHz#3 dB#50 Ohm#0 mil"

Select: Enter the starting point | 0 items | default | -0.896, 0.738 | -3.634, 0.376 | in | A/RF | SimSchem

To give users a conceptual sense of stability, the interface provides a clear, finite set of objects on a clear, finite set of actions to perform on those objects. Even when particular actions are unavailable, they are not eliminated from the user interface but are merely dimmed.

Modeless Features

Try to create modeless features that allow people to do whatever they want (and when they want) in DesignGuides. Avoid using modes because a mode typically restricts the operations that the user can perform while it is in effect. It locks the user into one operation and doesn't allow any other work to proceed until that operation is completed. In contrast, a modeless state allows the user to perform one operation at a time and thus gain more control over what he or she can do on the computer and in a DesignGuide. For example, consider the problems users would encounter if they could not use ADS while running a simulation. On the other hand, if the user initiates an operation that could cause ADS to crash, he or she should not be allowed to take any action until acknowledging the problem.

Aesthetic Integrity

Aesthetic integrity means that information is well organized and consistent with principles of visual design. This means that things look good on the screen and the display technology is of high quality.

Follow these guidelines:

- Keep user interface graphics simple. More is less. Too much information clutters an interface and confuses users. Don't clutter the screen with too many windows, overload the user with complex icons, or put dozens of buttons in dialog boxes.
- Don't change the meaning of standard items. For example, checkboxes should not be used for multiple choices one time and exclusive choices in another part of the interface.
- Don't use arbitrary graphic images to represent concepts. When you add nonstandard symbols to menus, dialog boxes, toolbars or palettes, the meaning may be clear to you, but to other people the symbols may appear as something different or distracting. Follow specific guidelines established for ADS graphic images, such as toolbar or palette icons.

Before You Start

Designing effective user interfaces is more than just following a set of rules. An understanding of your target customer is important to providing an effective solution that meets their needs. The creation of schematics and presentation of data is a highly individual and creative process. DesignGuides turn standard ADS schematic and data display windows into interactive user interfaces. A typical schematic or data display might be viewed by relatively few people throughout a typical project life cycle. DesignGuides will be viewed and used by many people.

This broader audience requires that DesignGuide developers have good understanding of what users want to accomplish. User profiles and understanding users work and task flows will provide focused information for creation of DesignGuides. Before reviewing user profiles and users task flow, there are a few general concepts that should be considered.

The 80/20 Rule

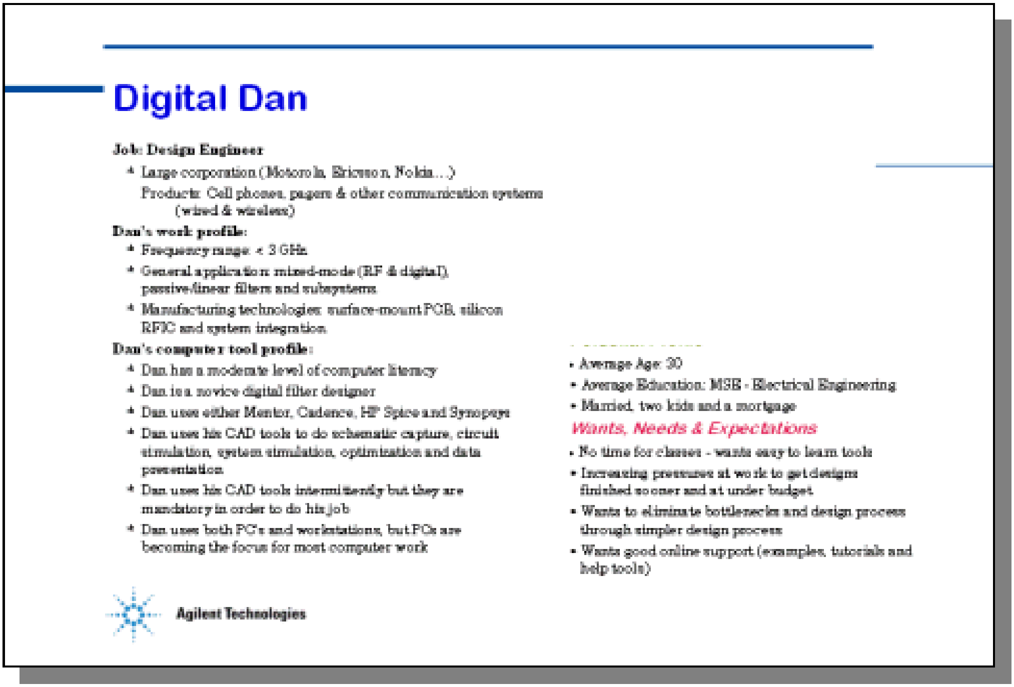
Eighty percent of users of a given application will use only twenty percent of the application features. While this statement may seem overly broad, especially when applied to CAD/CAE software tools, it is wise to review a design with this concept in mind. Does your DesignGuide contain the right amount of capability to meet a majority of your users needs? Is there an abundance of features that only very experienced users will want to use and may get in the way of the majority of the DesignGuides users?

First Hour Experience

User’s overall success with an application can be heavily influenced by their initial experience. This is not just using the software, but the whole product experience beginning from when they first open the software package. A good first impression is critical to gaining a user’s trust and confidence. Users will want to try an application immediately after it has been installed. Will your DesignGuide provide users with all the tools to become successful ADS users in less than an hour after installation is completed?

User Profiles

User profiles consolidate and focus a wealth of information about the people who use DesignGuides.



Digital Dan

Job: Design Engineer

- Large corporation (Motorola, Ericsson, Nokia…)

  Products: Cell phones, pagers & other communication systems (wired & wireless)

Dan's work profile:

- Frequency range: < 3 GHz
- General application: mixed-mode (RF & digital), passive/linear filters and subsystems
- Manufacturing technologies: surface-mount PCB, silicon RFIC and system integration

Dan's computer tool profile:

- Dan has a moderate level of computer literacy
- Dan is a novice digital filter designer
- Dan uses either Mentor, Cadence, HP Spice and Synopsys
- Dan uses his CAD tools to do schematic capture, circuit simulation, system simulation, optimization and data presentation
- Dan uses his CAD tools intermittently but they are mandatory in order to do his job
- Dan uses both PC's and workstations, but PCs are becoming the focus for most computer work

[illegible]

- Average Age: 30
- Average Education: MSE - Electrical Engineering
- Married, two kids and a mortgage

Wants, Needs & Expectations

- No time for classes - wants easy to learn tools
- Increasing pressures at work to get designs finished sooner and at under budget
- Wants to eliminate bottlenecks and design process through simpler design process
- Wants good online support (examples, tutorials and help tools)

Agilent Technologies

They help highlight specific areas of a users work that will be improved.

- Job description

  User's engineering expertise, technology and process knowledge and overall level of experience in these areas

- User objectives

  What gets in the way of the users ability to do his/her job and how will DesignGuides help eliminate those roadblocks?

- Demographics, Education, Skills and Characteristics

User’s job location (country and type of company), education – formal and on the job, specific job skills, age, career path

- Tools

  Hardware and software tools used by the user

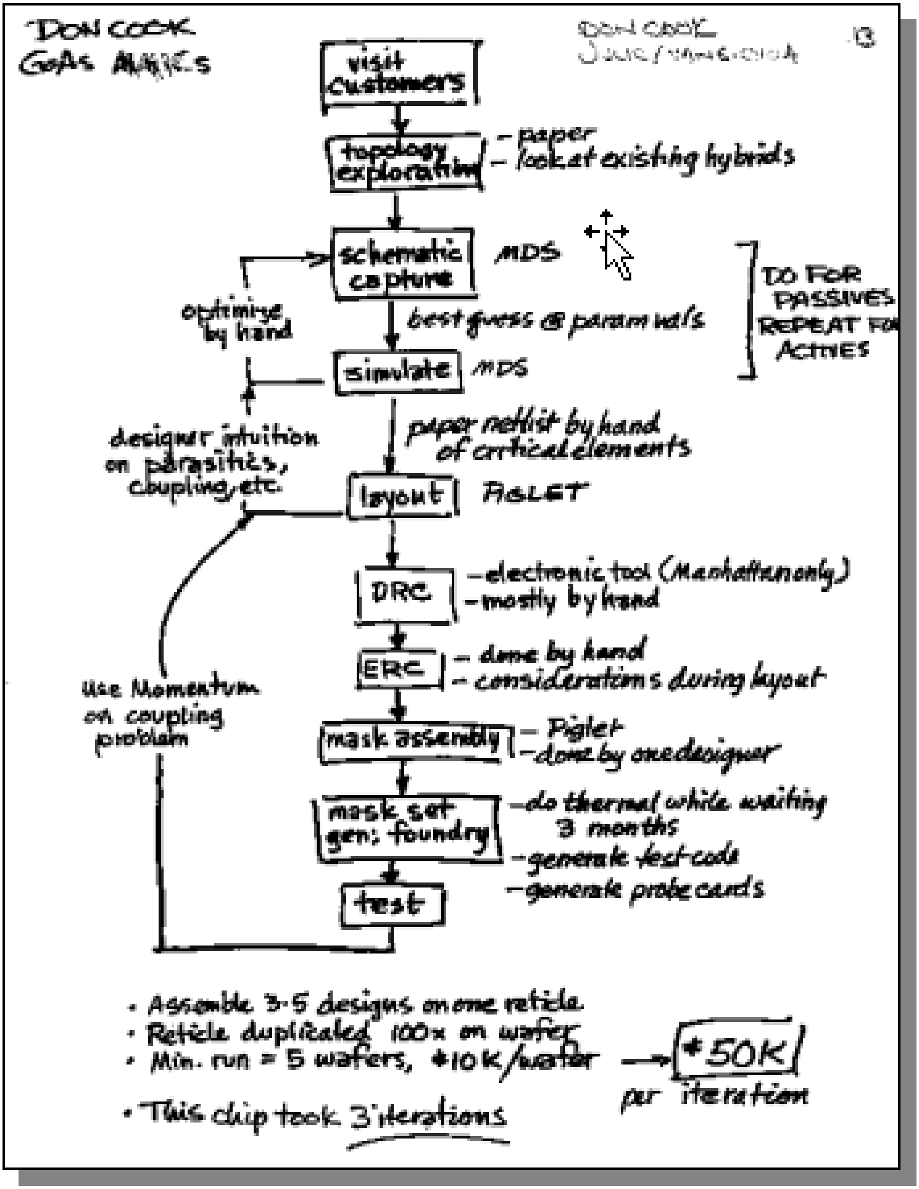
- Work environment

  R&D, production, field etc.

- Concerns and opportunities
- What will help users do their jobs better (schedules, money, meeting specs., project or process breakthroughs)?

User Task Flow

As shown in the illustration that follows, understanding the steps people take while doing their job will give you insights into where they need help – where the breakdown points are that increase costs, ruin schedules and cause missed specification targets. Task analysis can be as general or as specific as required. In the case of DesignGuides, you may need to focus on how the current ADS user interface impacts a user. However, an overview of a user’s entire task while building a part might also provide insight into areas where ADS can be applied, via DesignGuides in new and innovative ways.



- Assemble 3-5 designs on one reticle
- Reticle duplicated 100x on wafer
- Min. run = 5 wafers, $10K/wafer → $50K per iteration
- This chip took 3 iterations

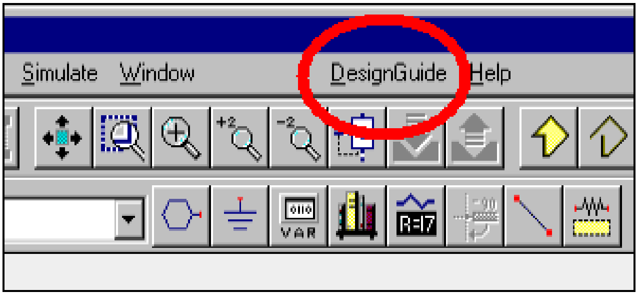
DesignGuide Usability Guidelines

The following guidelines are intended to provide DesignGuide developers with a set of rules that will provide a solid base from which to build DesignGuide user interfaces. Their purpose is to help remove ambiguities that can confuse and frustrate users by providing a consistent look and feel across the DesignGuide user interface.

As DesignGuides mature, these guidelines will change. Periodic reviews of the DesignGuides, possibly based on specific releases, should be anticipated. This style guide will follow the typical task flow a user follows when using DesignGuides. For each task, specific guidelines will be reviewed. Additional topics have been added to cover areas not specific to a task.

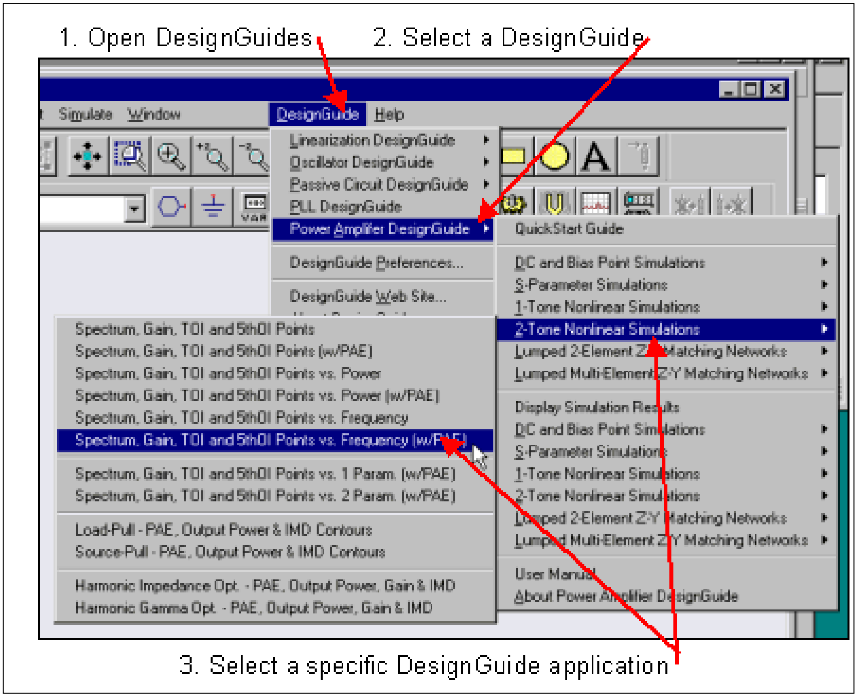
Opening and Selecting DesignGuides

The DesignGuide menu structure allows users to perform all DesignGuide operations (viewing a Quick Start Guide, opening specific DesignGuide applications, opening DesignGuide palettes, viewing simulation results and opening the User Manual and QuickStart documents).



DesignGuide Menu Structure

When building a DesignGuide, the menu should be structured to guide users sequentially through the DesignGuide process.



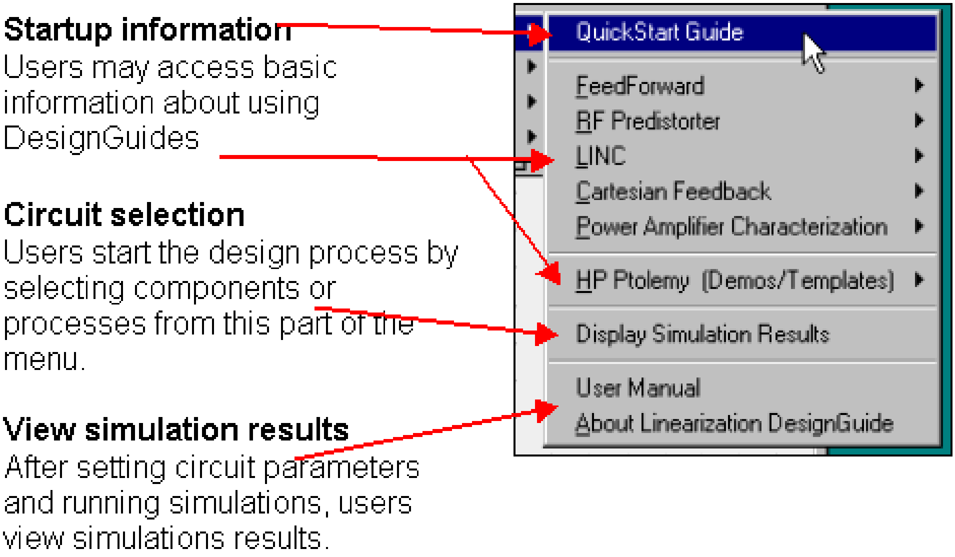
1. Open DesignGuides

2. Select a DesignGuide

3. Select a specific DesignGuide application

Simulate
Window
DesignGuide
Help
Linearization DesignGuide
Oscillator DesignGuide
Passive Circuit DesignGuide
PLL DesignGuide
Power Amplifier DesignGuide
DesignGuide Preferences...
DesignGuide Web Site...
QuickStart Guide
DC and Bias Point Simulations
S-Parameter Simulations
1-Tone Nonlinear Simulations
2-Tone Nonlinear Simulations
Lumped 2-Element Z-Y Matching Networks
Lumped Multi-Element Z-Y Matching Networks
Display Simulation Results
DC and Bias Point Simulations
S-Parameter Simulations
1-Tone Nonlinear Simulations
2-Tone Nonlinear Simulations
Lumped 2-Element Z-Y Matching Networks
Lumped Multi-Element Z-Y Matching Networks
User Manual
About Power Amplifier DesignGuide
Spectrum, Gain, TOI and 5thOI Points
Spectrum, Gain, TOI and 5thOI Points (w/PAE)
Spectrum, Gain, TOI and 5thOI Points vs. Power
Spectrum, Gain, TOI and 5thOI Points vs. Power (w/PAE)
Spectrum, Gain, TOI and 5thOI Points vs. Frequency
Spectrum, Gain, TOI and 5thOI Points vs. Frequency (w/PAE)
Spectrum, Gain, TOI and 5thOI Points vs. 1 Param. (w/PAE)
Spectrum, Gain, TOI and 5thOI Points vs. 2 Param. (w/PAE)
Load-Pull - PAE, Output Power & IMD Contours
Source-Pull - PAE, Output Power & IMD Contours
Harmonic Impedance Opt. - PAE, Output Power, Gain & IMD
Harmonic Gamma Opt. - PAE, Output Power, Gain & IMD

The following illustration shows a standard DesignGuide menu layout.



Startup information
Users may access basic information about using DesignGuides

Circuit selection
Users start the design process by selecting components or processes from this part of the menu.

View simulation results
After setting circuit parameters and running simulations, users view simulations results.

QuickStart Guide
FeedForward
RF Predistorter
LINC
Cartesian Feedback
Power Amplifier Characterization
HP Ptolemy (Demos/Templates)
Display Simulation Results
User Manual
About Linearization DesignGuide

Menu Usability Notes

Following are guidelines on creating usable menus.

Menu Labels

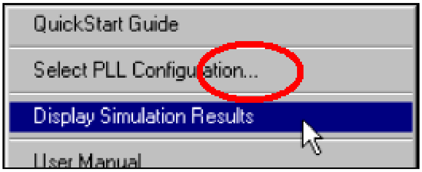
DesignGuide menu labels should avoid using non-standard abbreviations or descriptions that have no meaning to users. Users should not have to guess at what kind of window will open when a DesignGuide menu is selected. Shorter is better.

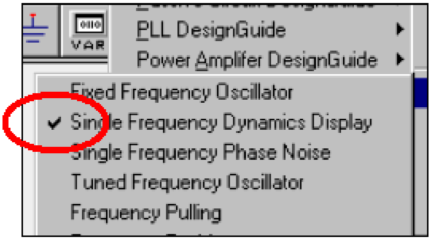
Menu Levels

General user software guidelines suggest that pulldown menus should be structured so that users do not have to go more than two levels deep, with three as the maximum.

Standard Menu Control Features

Use standard operating system fonts, navigation controls (sub-menu arrows and label ellipses to indicate dialog boxes) and keyboard shortcuts indicators for DesignGuide menus.

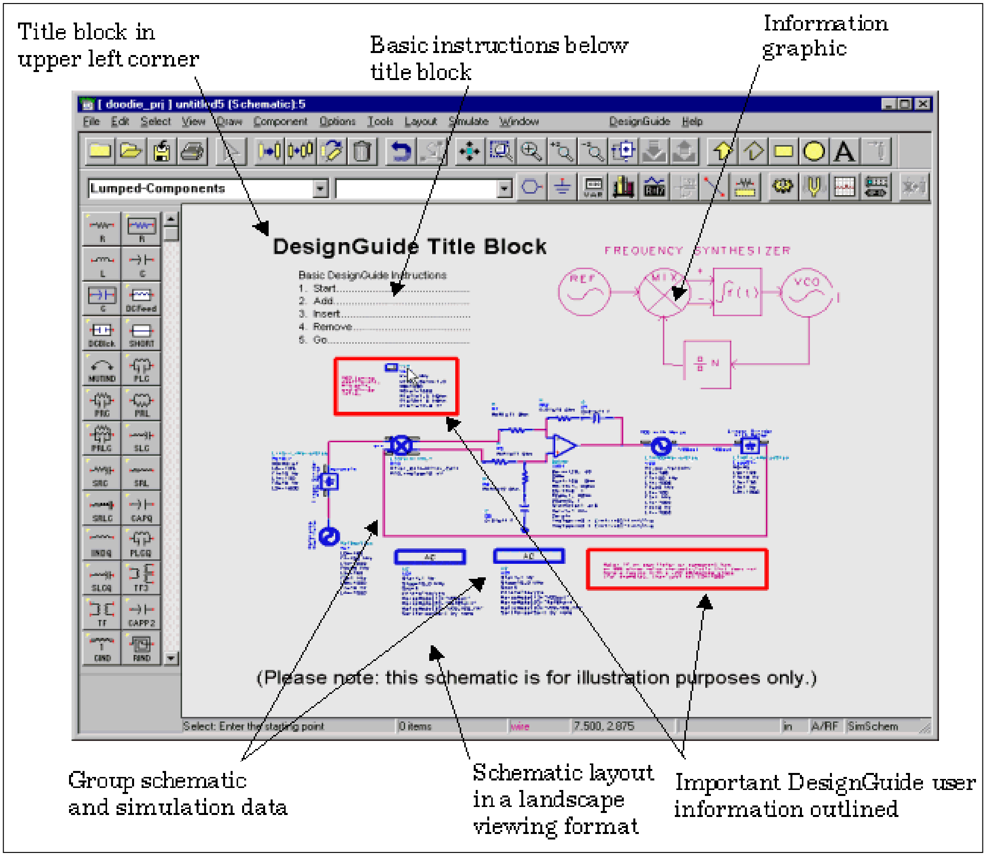



VAR
PLL DesignGuide
Power Amplifer DesignGuide
Fixed Frequency Oscillator
✓ Single Frequency Dynamics Display
Single Frequency Phase Noise
Tuned Frequency Oscillator
Frequency Pulling

DesignGuide Schematics

The physical layout of schematic and data display windows is a highly iterative and creative process. The interactive nature of DesignGuides requires developers to take a closer look at the overall usability of schematics. By applying a few basic usability principles to DesignGuide schematics, users will experience better ease of learning and increased ease of use. This section covers guidelines for schematics. Data display guidelines will be reviewed in the next section.

The following illustration shows a hypothetical DesignGuide Schematic layout.



Title block in upper left corner

Basic instructions below title block

Information graphic

Group schematic and simulation data

Schematic layout in a landscape viewing format

Important DesignGuide user information outlined

DesignGuide Title Block

Basic DesignGuide Instructions

1. Start..................................
2. Add....................................
3. Insert..................................
4. Remove...............................
5. Go......................................

(Please note: this schematic is for illustration purposes only.)

Select: Enter the starting point | 0 items | wire | 7.500, 2.875 | in | A/RF | SimSchem

Schematic Window Usability Notes

Following are tips on usability of the Schematic window.

Overall Format for DesignGuide Layouts

Use a landscape format when creating schematic and data display layouts. This format corresponds to the computer screen aspect ratio. Try to group components within the window for quick visual recognition. Well grouped schematics also help users when performing cut and paste operations and when printing schematics.

Locating Primary User Information

If possible, schematic and data display windows should locate the most important user data in the upper left and top of the window. Users generally start to scan their view from this location. This provides users with preliminary information and instructions about how to use the DesignGuide. If necessary, provide links from instructions (numbers, color coding etc.) to relevant points in the windows. This helps users focus on information and processes important to their success.

On-schematic Help for New Users

If your users are predominantly new to ADS, include hints or descriptions of ADS commands that may be unfamiliar to them. For example, users may not know how to access sub-circuits. A note about the *Push-in* and *Pop-out* features of ADS might be placed on the schematic near a sub-circuit symbol.

Selecting Text for DesignGuides

Title, descriptive and instruction text should be large enough for users to read when a schematic or data display window is open in the full screen mode in a 15” to 16” CRT. The Arial font family is recommended for text in ADS DesignGuides.

- For titles and headings (24 point and above)

  Arial for CAE Bold (color black)

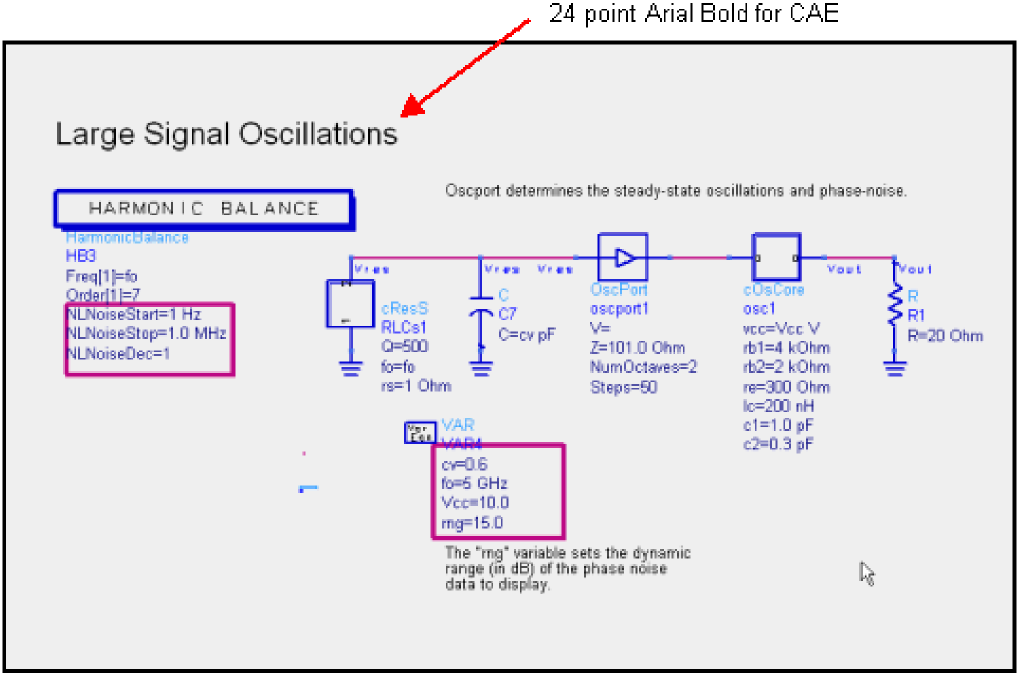
  (Do not use outlining around titles or headings.)

- For components, instructions and description text (12 point and above)

  Arial for CAE Bold (color black)

Specify the text size in your DesignGuide to provide optimal viewing and reading capabilities within your DesignGuide windows. The following two figures show applications of title text based on schematic size.

24 point Arial Bold for CAE



Large Signal Oscillations

HARMONIC BALANCE

HarmonicBalance
HB3
Freq[1]=fo
Order[1]=7
NLNoiseStart=1 Hz
NLNoiseStop=1.0 MHz
NLNoiseDec=1

Oscport determines the steady-state oscillations and phase-noise.

cResS
RLCs1
Q=500
fo=fo
rs=1 Ohm

C
C7
C=cv pF

OscPort
oscport1
V=
Z=101.0 Ohm
NumOctaves=2
Steps=50

cOsCore
osc1
vcc=Vcc V
rb1=4 kOhm
rb2=2 kOhm
re=300 Ohm
lc=200 nH
c1=1.0 pF
c2=0.3 pF

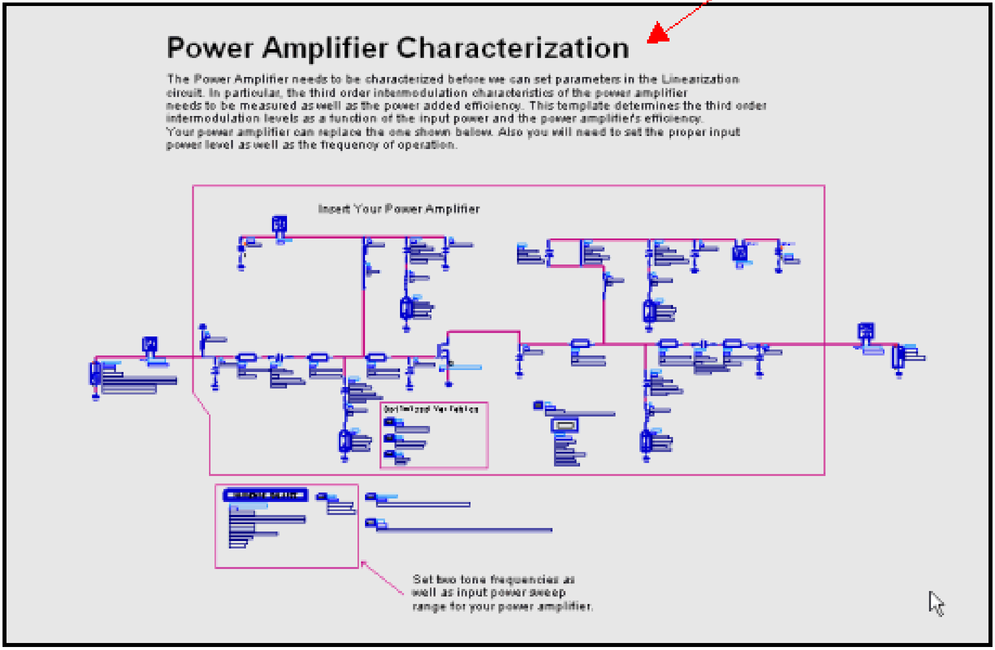
R
R1
R=20 Ohm

VAR
VAR4
cv=0.6
fo=5 GHz
Vcc=10.0
rng=15.0

The "rng" variable sets the dynamic range (in dB) of the phase noise data to display.

"Small schematic"

80 point Arial Bold for CAE



Power Amplifier Characterization

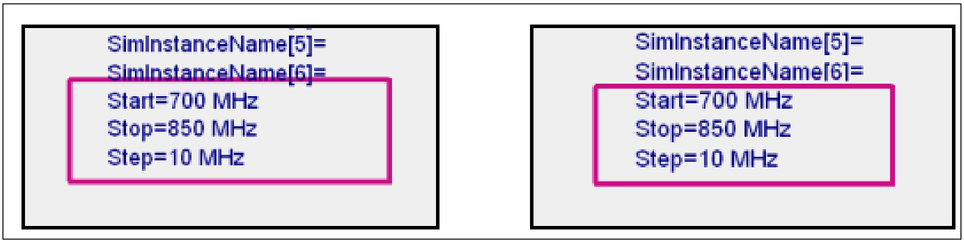
The Power Amplifier needs to be characterized before we can set parameters in the Linearization circuit. In particular, the third order intermodulation characteristics of the power amplifier needs to be measured as well as the power added efficiency. This template determines the third order intermodulation levels as a function of the input power and the power amplifier's efficiency.
Your power amplifier can replace the one shown below. Also you will need to set the proper input power level as well as the frequency of operation.

Set two tone frequencies as well as input power sweep range for your power amplifier.

"Large schematic"

Outlining Important Information in Schematics

When outlining critical instructions for users, take care when placing the box so that you do not hide other schematic information. This may require turning off the snap-to-grid feature. The figure shows current outlining (left) and revised outlining (right).

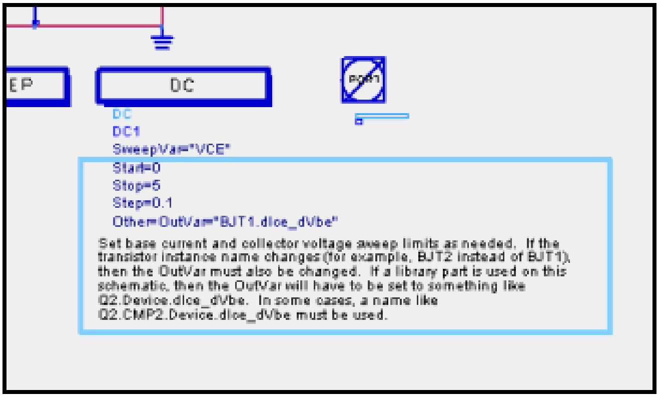


SimInstanceName[5]=
SimInstanceName[6]=
Start=700 MHz
Stop=850 MHz
Step=10 MHz

SimInstanceName[5]=
SimInstanceName[6]=
Start=700 MHz
Stop=850 MHz
Step=10 MHz

Red outlining has been the predominant method of highlighting important information in DesignGuides. To reduce the impact of this color in schematics and data displays with a lot of outlining, consider using a combination of lighter line weight. Review the following guidelines.

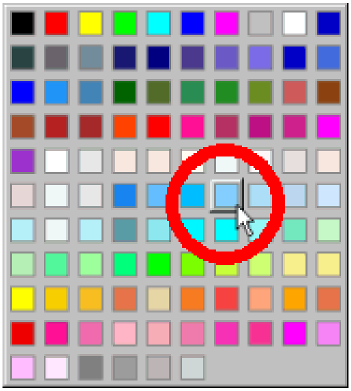
- Use a line weight of 3 and light blue color for important information.
- Use a line weight of 3 and the color red for critical information.



DC
DC1
SweepVar="VCE"
Start=0
Stop=5
Step=0.1
Other=OutVar="BJT1.dIce_dVbe"

Set base current and collector voltage sweep limits as needed. If the transistor instance name changes (for example, BJT2 instead of BJT1), then the OutVar must also be changed. If a library part is used on this schematic, then the OutVar will have to be set to something like Q2.Device.dIce_dVbe. In some cases, a name like Q2.CMP2.Device.dIce_dVbe must be used.

Suggested outline color



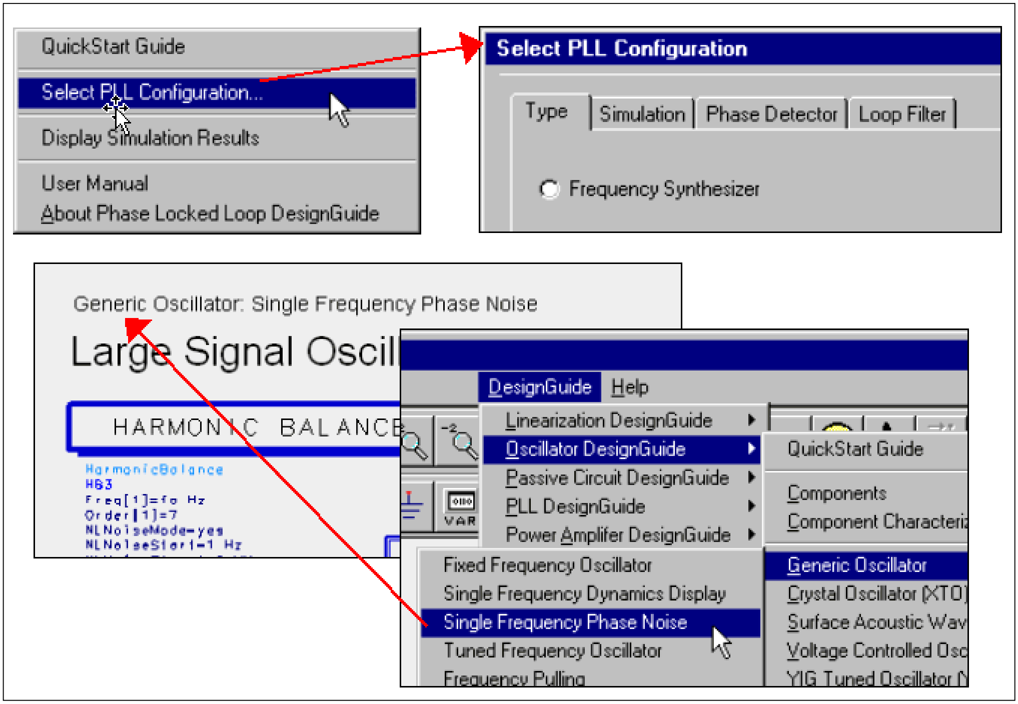
ADS color palette found in the Layer Editor dialog box (Display Control-Color/Pattern)

To create an outline (rectangle):

1. Insert a rectangle into a Schematic window.
2. Select the rectangle (it becomes highlighted).
3. Select **Edit** > **Properties**.
4. In the Properties dialog box, under *Name*, enter: **line_thickness_prop.**
5. Under *Value*, enter: **3.**
6. Click **Add.**
7. Click **OK**.

Consistent Naming of DesignGuide Controls

Providing consistent labeling between user interface components and controls gives users a visual link between an action taken and the result of that action. For example, if a user selects a menu command, the label of the UI component opened should be consistent with the menu label.



Keep data display and schematic names the same if possible. For example, the Power Amplifier Design Guide has the schematic:

HarmZopt1tone.dsn

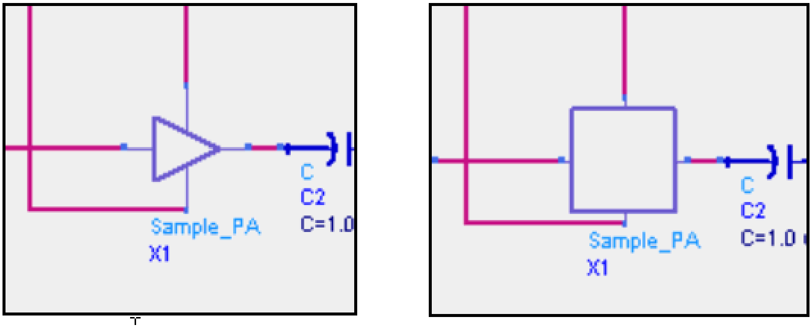
It also has three corresponding data displays

- *HarmZopt1tone.dds*
- *HarmZopt1toneSC.dds*
- *HarmZopt1toneTime.dds*

It should be clear from the DesignGuide names that the files are related.

Labeling and Symbols for Subcircuits in Schematics

When showing sub-circuits in top level schematics, do not use generic boxes, but symbols appropriate to the sub-circuit. In the illustration that follows,the appropriate symbol (amplifier) is shown on the left, with a generic box symbol on the right.



DesignGuide Data Displays

Many Usability Notes reviewed in the section “DesignGuide Schematics” on page 10-14 also apply to DesignGuide data displays. Additional usability guidelines for data displays follow.

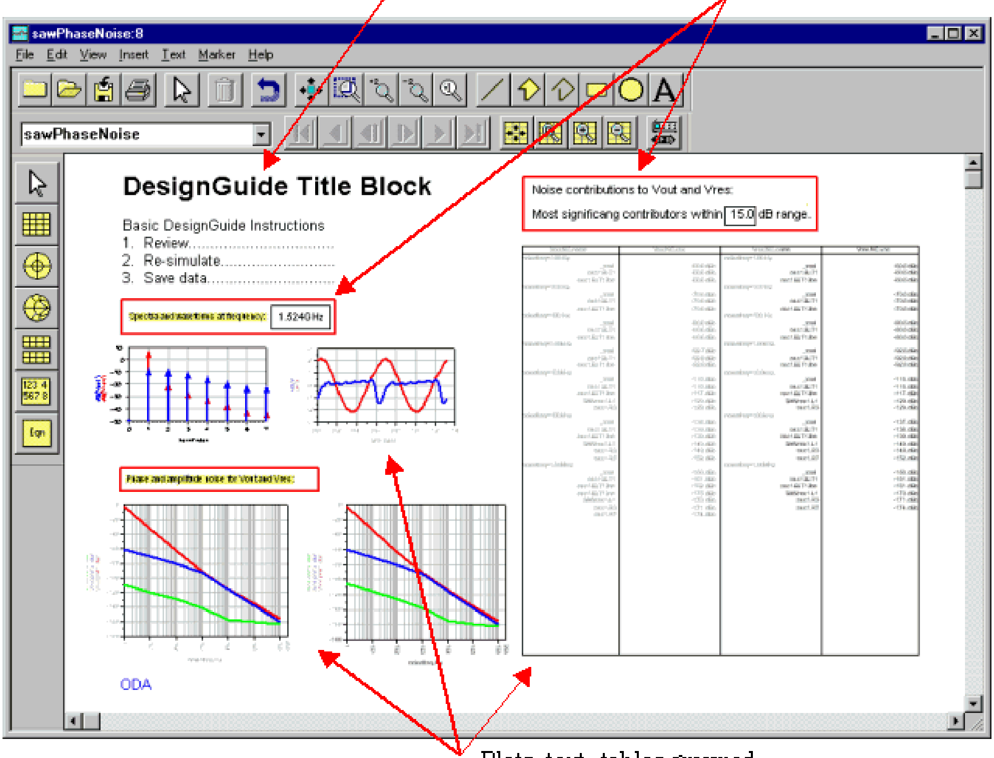
Remember, keep it simple and provide the MVUI (minimum viable user interface).

The following shows a hypothetical DesignGuide data display layout.

.

Title block and basic instructions

Important information boxed



DesignGuide Title Block

Basic DesignGuide Instructions

1. Review..................................
2. Re-simulate..........................
3. Save data.............................

Spectra and waveforms at frequency: 1.5240 Hz

Phase and amplitude noise for Vout and Vres:

Noise contributions to Vout and Vres:

Most significang contributors within 15.0 dB range.

ODA

Plots, text, tables grouped for easy viewing

Data Display Window Usability Notes

Following are tips on usability for the data display windows.

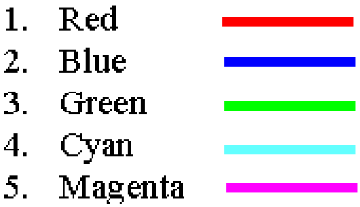
Highlighting Data in Data Display Windows

Do not use highlighting (using a background fill color – available only in data display windows), when creating DesignGuides. Printing the highlighted text can produce poor results. Use the outlining method of highlighting important areas of the data display.

Selecting Color for Data Display Traces

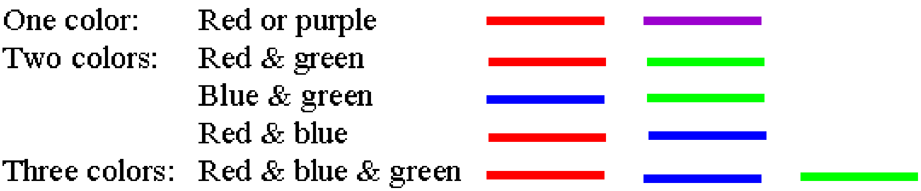
Data display traces should be programmed to provide default values when data displays are created for DesignGuides. Users should be free to easily change trace colors to meet their specific needs. Colors on computers are also viewed on a wide variety of hardware in a wide variety of environments. These factors have a big impact on color perception.

- Use red for traces that indicate the most important data.
- Use the following colors, shown in order of preference, to highlight specific trace data on a white background.



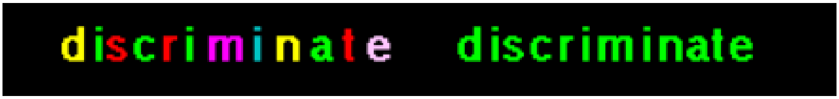
1. Red 🟥
2. Blue 🟦
3. Green 🟩
4. Cyan
5. Magenta 🟪

- The following color combinations are recommended for thin line images (traces) on a white background.

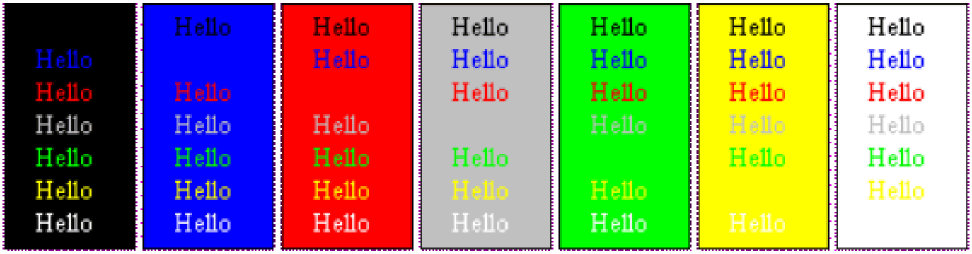


One color:	Red or purple	🟥	🟪	
Two colors:	Red & green	🟥	🟩	
	Blue & green	🟦	🟩	
	Red & blue	🟥	🟦	
Three colors:	Red & blue & green	🟥	🟦	🟩

- Whenever possible, stay away from de-saturated bright colors (yellow, light green)
- Avoid colors that are hard to see on a white background or when printing in color.
- Use colors sparingly.

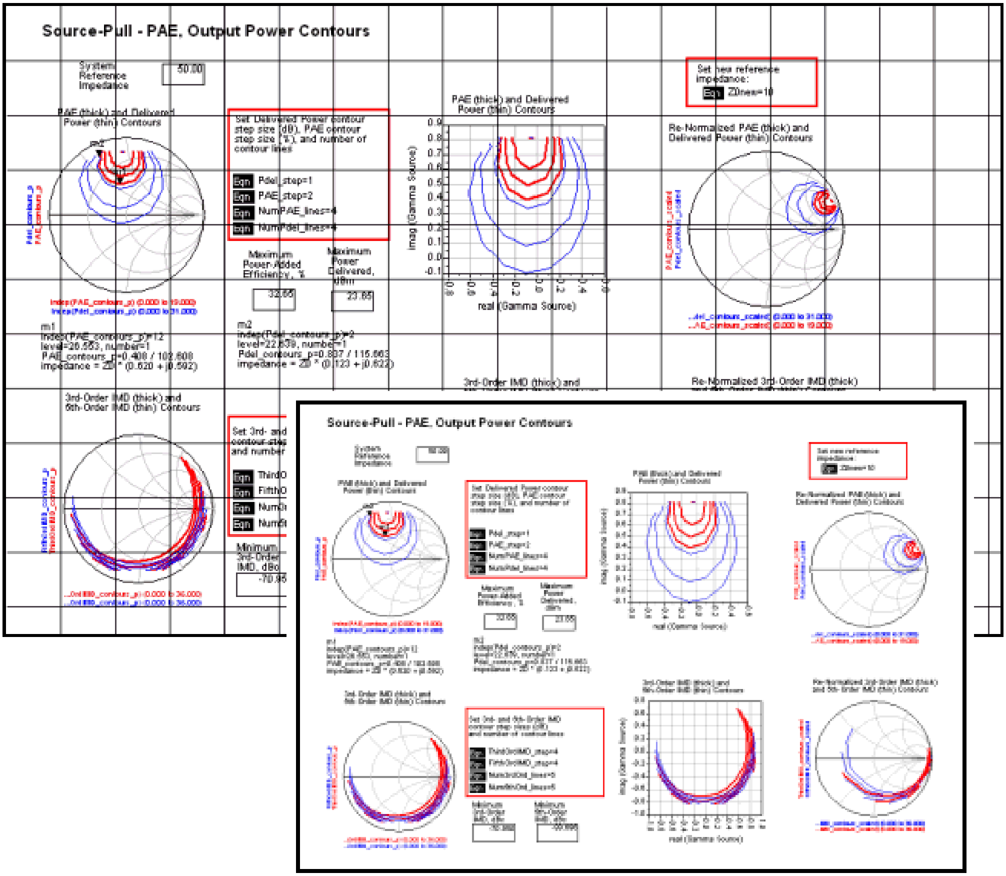


- Black text on a white background is preferred.



Using a Grid for a Data Display (and Schematic) Layout Design

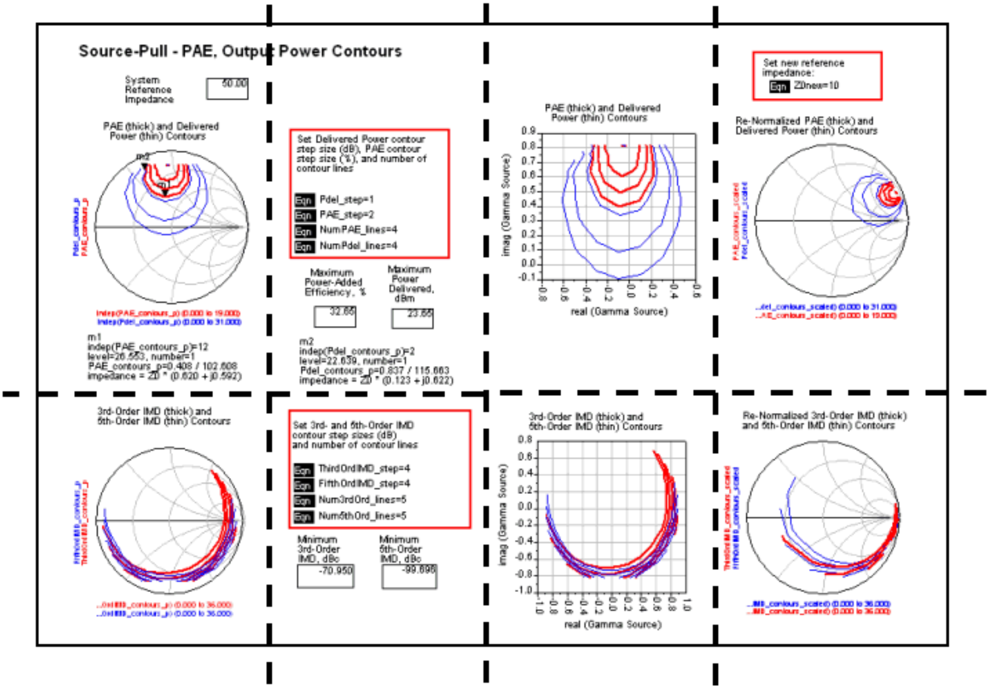
Consider using a grid to help you place components when you are laying out either a Schematic or data display window. Grids will help you provide visual order to the window. Visual order helps users gain a quicker understanding of the information you are presenting. It will also help users when they want to cut, paste and print parts of the data display window.


Source-Pull - PAE, Output Power Contours
System Reference Impedance
50.00
PAE (thick) and Delivered Power (thin) Contours
Set Delivered Power contour step size (dB), PAE contour step size (%), and number of contour lines
Eqn Pdel_step=1
Eqn PAE_step=2
Eqn NumPAE_lines=4
Eqn NumPdel_lines=4
Maximum Power-Added Efficiency, %
32.65
Maximum Power Delivered, dBm
23.85
Set new reference impedance:
Eqn Z0new=10
Re-Normalized PAE (thick) and Delivered Power (thin) Contours
imag (Gamma Source)
real (Gamma Source)
m1
m2
3rd-Order IMD (thick) and 5th-Order IMD (thin) Contours
Re-Normalized 3rd-Order IMD (thick)
Minimum 3rd-Order IMD, dBc
-70.95

Printing DesignGuide Windows

The hardcopy lab notebook is still a mainstay for R&D engineers. Printouts of computer data are routinely pasted into lab notebooks to record design history. With this in mind, the following guidelines should be followed.

- Because most documents are printed in portrait mode, consider the grouping of data display components so that they can be easily printed in portrait mode.
- When locating specific data display components, allow room between components so that individual components or groups of components can be easily selected for printing without overlapping other components.



Source-Pull - PAE, Output Power Contours

System Reference Impedance: 50.00

Set Delivered Power contour step size (dB), PAE contour step size (%), and number of contour lines

Eqn Pdel_step=1
Eqn PAE_step=2
Eqn NumPAE_lines=4
Eqn NumPdel_lines=4

Maximum Power-Added Efficiency, %: 32.65

Maximum Power Delivered, dBm: 23.65

m1
indep(PAE_contours_p)=12
level=26.553, number=1
PAE_contours_p=0.408 / 102.608
impedance = Z0 * (0.620 + j0.592)

m2
indep(Pdel_contours_p)=2
level=22.639, number=1
Pdel_contours_p=0.837 / 115.663
impedance = Z0 * (0.123 + j0.622)

Set new reference impedance:

Eqn Z0new=10

Set 3rd- and 5th-Order IMD contour step sizes (dB) and number of contour lines

Eqn ThirdOrdIMD_step=4
Eqn FifthOrdIMD_step=4
Eqn Num3rdOrd_lines=5
Eqn Num5thOrd_lines=5

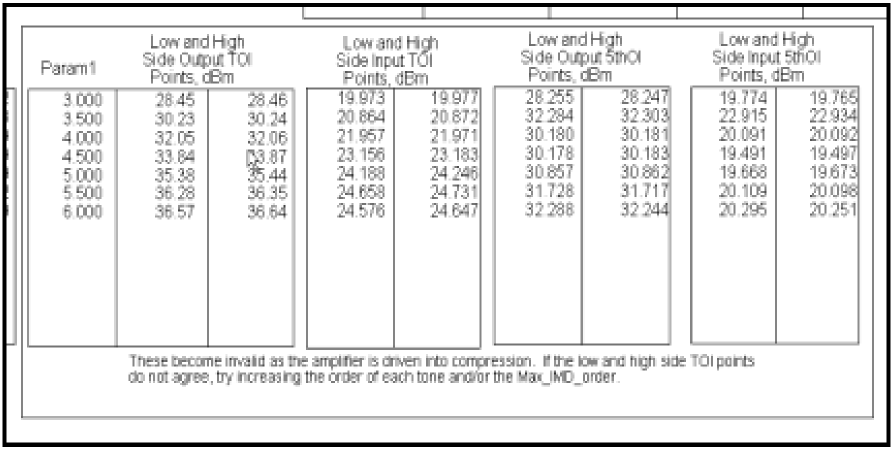
Minimum 3rd-Order IMD, dBc: -70.950

Minimum 5th-Order IMD, dBc: -99.696

PAE (thick) and Delivered Power (thin) Contours
m2
m1
Pdel_contours_p
PAE_contours_p
indep(PAE_contours_p) (0.000 to 19.000)
indep(Pdel_contours_p) (0.000 to 31.000)
PAE (thick) and Delivered Power (thin) Contours
imag (Gamma Source)
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
real (Gamma Source)
Re-Normalized PAE (thick) and Delivered Power (thin) Contours
PAE_contours_scaled
Pdel_contours_scaled
...del_contours_scaled) (0.000 to 31.000)
...AE_contours_scaled) (0.000 to 19.000)
3rd-Order IMD (thick) and 5th-Order IMD (thin) Contours
FifthOrdIMD_contours_p
ThirdOrdIMD_contours_p
...OrdIMD_contours_p) (0.000 to 36.000)
...OrdIMD_contours_p) (0.000 to 36.000)
3rd-Order IMD (thick) and 5th-Order IMD (thin) Contours
imag (Gamma Source)
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
real (Gamma Source)
Re-Normalized 3rd-Order IMD (thick) and 5th-Order IMD (thin) Contours
ThirdOrdIMD_contours_scaled
FifthOrdIMD_contours_scaled
...IMD_contours_scaled) (0.000 to 36.000)
...IMD_contours_scaled) (0.000 to 36.000)

Linking Listing Columns for Simultaneous Scrolling

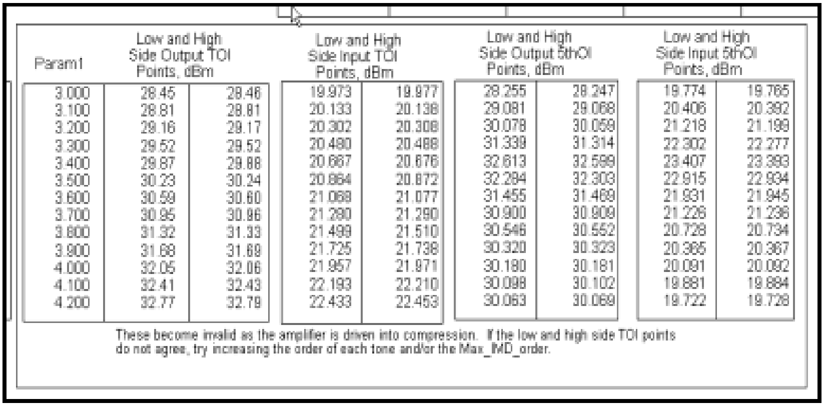
Listing columns of related data should be linked so that scrolling of the data works simultaneously for all columns. In the figure that follows, separate listings items, as shown, are acceptable when scrolling is not needed (list columns not full).



Param1	Low and High Side Output TOI Points, dBm		Low and High Side Input TOI Points, dBm		Low and High Side Output 5thOI Points, dBm		Low and High Side Input 5thOI Points, dBm	
3.000	28.45	28.46	19.973	19.977	28.255	28.247	19.774	19.765
3.500	30.23	30.24	20.864	20.872	32.284	32.303	22.915	22.934
4.000	32.05	32.06	21.957	21.971	30.180	30.181	20.091	20.092
4.500	33.84	33.87	23.156	23.183	30.178	30.183	19.491	19.497
5.000	35.38	35.44	24.188	24.246	30.857	30.862	19.668	19.673
5.500	36.28	36.35	24.658	24.731	31.728	31.717	20.109	20.098
6.000	36.57	36.64	24.576	24.647	32.288	32.244	20.295	20.251

These become invalid as the amplifier is driven into compression. If the low and high side TOI points do not agree, try increasing the order of each tone and/or the Max_IMD_order.

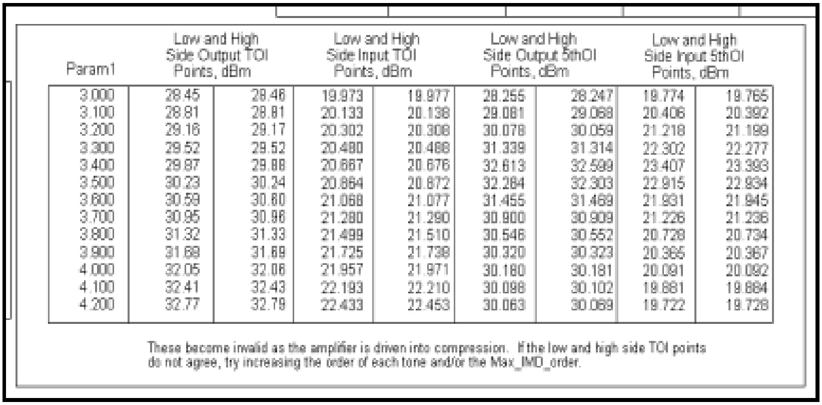
When listing columns are filled and scrolling is needed to view all data, separate columns do not allow the user to view correlated data across all lists.



Param1	Low and High Side Output TOI Points, dBm		Low and High Side Input TOI Points, dBm		Low and High Side Output 5thOI Points, dBm		Low and High Side Input 5thOI Points, dBm	
3.000	28.45	28.46	19.973	19.977	28.255	28.247	19.774	19.765
3.100	28.81	28.81	20.133	20.138	29.081	29.068	20.406	20.392
3.200	29.16	29.17	20.302	20.308	30.078	30.059	21.218	21.199
3.300	29.52	29.52	20.480	20.488	31.339	31.314	22.302	22.277
3.400	29.87	29.88	20.667	20.676	32.613	32.599	23.407	23.393
3.500	30.23	30.24	20.864	20.872	32.284	32.303	22.915	22.934
3.600	30.59	30.60	21.068	21.077	31.455	31.469	21.931	21.945
3.700	30.95	30.96	21.280	21.290	30.900	30.909	21.226	21.236
3.800	31.32	31.33	21.499	21.510	30.546	30.552	20.728	20.734
3.900	31.68	31.69	21.725	21.738	30.320	30.323	20.365	20.367
4.000	32.05	32.06	21.957	21.971	30.180	30.181	20.091	20.092
4.100	32.41	32.43	22.193	22.210	30.098	30.102	19.881	19.884
4.200	32.77	32.79	22.433	22.453	30.063	30.069	19.722	19.728

These become invalid as the amplifier is driven into compression. If the low and high side TOI points do not agree, try increasing the order of each tone and/or the Max_IMD_order.

If all listed data must be viewed while one column is scrolled, listing columns must be linked.

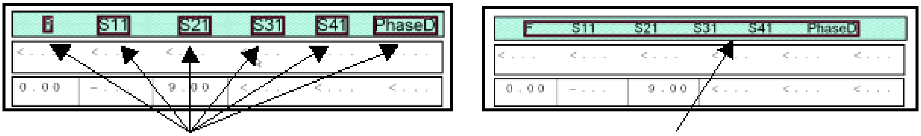


Param1	Low and High Side Output TOI Points, dBm		Low and High Side Input TOI Points, dBm		Low and High Side Output 5thOI Points, dBm		Low and High Side Input 5thOI Points, dBm	
3.000	28.45	28.46	19.973	19.977	28.255	28.247	19.774	19.765
3.100	28.81	28.81	20.133	20.138	29.081	29.068	20.406	20.392
3.200	29.16	29.17	20.302	20.308	30.078	30.059	21.218	21.199
3.300	29.52	29.52	20.480	20.488	31.339	31.314	22.302	22.277
3.400	29.87	29.88	20.667	20.676	32.613	32.599	23.407	23.393
3.500	30.23	30.24	20.864	20.872	32.284	32.303	22.915	22.934
3.600	30.59	30.60	21.068	21.077	31.455	31.469	21.931	21.945
3.700	30.95	30.96	21.280	21.290	30.900	30.909	21.226	21.236
3.800	31.32	31.33	21.499	21.510	30.546	30.552	20.728	20.734
3.900	31.68	31.69	21.725	21.738	30.320	30.323	20.365	20.367
4.000	32.05	32.06	21.957	21.971	30.180	30.181	20.091	20.092
4.100	32.41	32.43	22.193	22.210	30.098	30.102	19.881	19.884
4.200	32.77	32.79	22.433	22.453	30.063	30.069	19.722	19.728

These become invalid as the amplifier is driven into compression. If the low and high side TOI points do not agree, try increasing the order of each tone and/or the Max_IMD_order.

Titles for Listing Columns

When using text as titles for listing columns, use separate text entries for each title. Though this is time consuming, when one long text entry is used for a group of the titles, scaling of text is a problem when a user zooms in and out of the data display window.



Reducing DesignGuide Datasets

DesignGuides are capable of generating a great amount of simulation data, to the point that it might have an adverse impact on a user's disk space. To minimize the impact of large amounts of dataset data, DesignGuide developers should consider the following guidelines when setting up simulations.

- Set up simulations with coarse sweeps. Let the user decide how to refine their sweep parameters in subsequent simulations.

- Don’t sweep over larger ranges than necessary. Again, start with a minimal range of parameters. Let the user decide to expand the range is necessary.
- Delete unnecessary datasets.

Hiding Equations

A few assumptions can be made about the need to either show or hide equations.

- The average user is not interested in seeing every equation and might in fact be put off by a window full of equations.
- Power users want to have access to equations to understand the inner workings of a DesignGuide process.
- Showing equations may compromise Agilent intellectual property.
- Equations take up valuable space in schematic and data display windows.

Based on these assumptions, two approaches for equations are suggested.

1. If there are 10 or fewer equations, don’t hide them. Make them small, using a text size of 6 point Arial for CAE. Include some basic documentation and place near the equation.
2. For DesignGuides with many equations, hide the equations using the zero font approach. This is done by placing equations on top of each other and reducing the font size to zero. Place them below the lower left-hand plot or listing column. Other suggestions are placing the hidden equations under the data display title or by placing the data display title over the equations.

Designing Dialog Boxes

This section provides guidelines on user-friendly dialog box design.

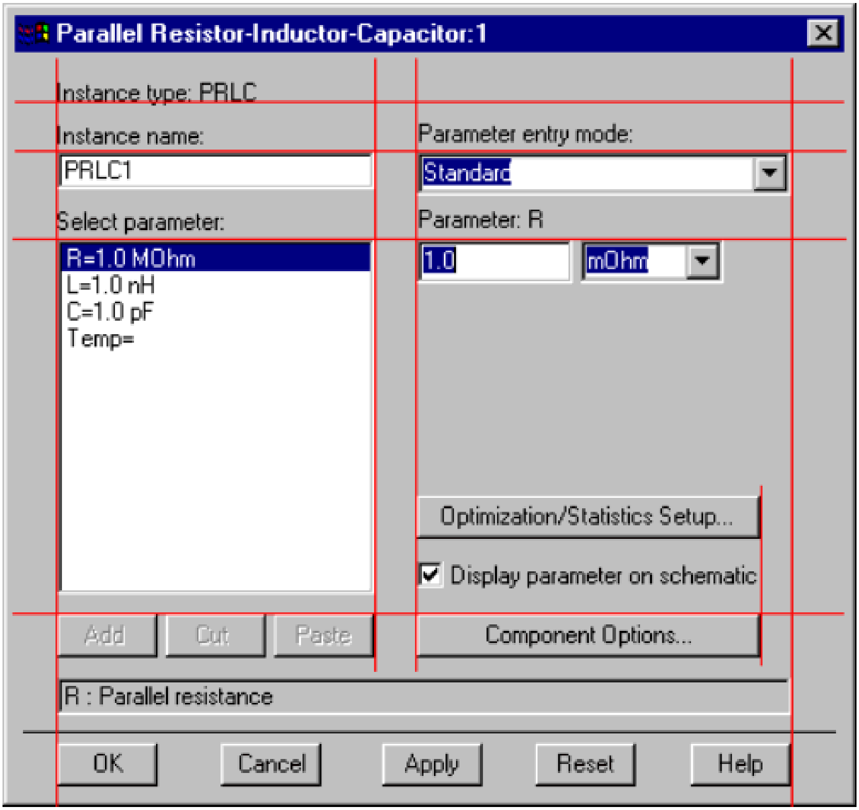
Dialog Box Usability Notes

DesignGuides will see an ever-increasing use of dialog boxes. When designing dialog boxes, a few usability rules will help make them use easier to learn and easier to use. The goal in dialog box design is to let the user make intelligent choices, quickly.

- **Controls**. Use the appropriate control. Controls are optimized for certain types of functions. The wrong control not only affects the user's efficiency, but also confuses the user about the purpose of your design.
- **Layout conventions**. Use recommended layout conventions. For example, buttons such as *OK* and *Cancel* or *Yes* and *No* should be aligned either at the top right or bottom right of the dialog box. *OK* is always the first button, followed by *Cance*l, and then any other buttons. If you don't have an *OK* button, then *Cancel* follows all the other buttons.
- **Labeling.** Use appropriate labeling. Always use the appropriate capitalization and access key assignments. Include colons when you use static text to label another control. This not only identifies the text as a label, but also provides a cue for screen reader utilities.
- **Control alignment.** Use appropriate alignment. Alignment affects readability and therefore usability and efficiency. It also affects the user's overall impression of the quality of your application.
- **Control spacing**. Use appropriate sizing, spacing and margins. Make good use of overall space. Avoid cramming too many controls together if you have additional space.
- **Default dialog box location.** When opening a dialog box for the first time, default the location to the center of the active window. If the user moves the dialog box, display the it at the new location the next time the user opens the window, adjusted as necessary to the current display configuration.
- **Button size.** Make buttons a consistent length for readability. However, if maintaining this consistency greatly expands the space required for a set of buttons, it might be reasonable to have one button larger than the rest.

- **Tabs.** Similarly, if you use tabs, try to maintain a consistent width for all tabs in the same window (and in the same dimension). However, if a particular tab's label makes this unworkable, size it larger and maintain a smaller, consistent size for the other tabs. If possible, avoid designing dialog boxes that require the user to scroll left or right (using arrow controls) to view tabs.
- **Grouping controls**. Group related components – you can use group box controls, separator lines, or spacing. Avoid using a group box when you have only one set of related items or where the group box may take too much space or add visual clutter rather than structure. Instead, consider using separators to group related items.

- **Dialog box task flow.** Design the dialog box so that the user works from right to left and top to bottom when making choices.



Parallel Resistor-Inductor-Capacitor:1

Instance type: PRLC

Instance name: PRLC1

Select parameter:

R=1.0 MOhm
L=1.0 nH
C=1.0 pF
Temp=

Parameter entry mode: Standard

Parameter: R 1.0 mOhm

☑ Display parameter on schematic

R : Parallel resistance

Designing Icons for Toolbars and Palettes

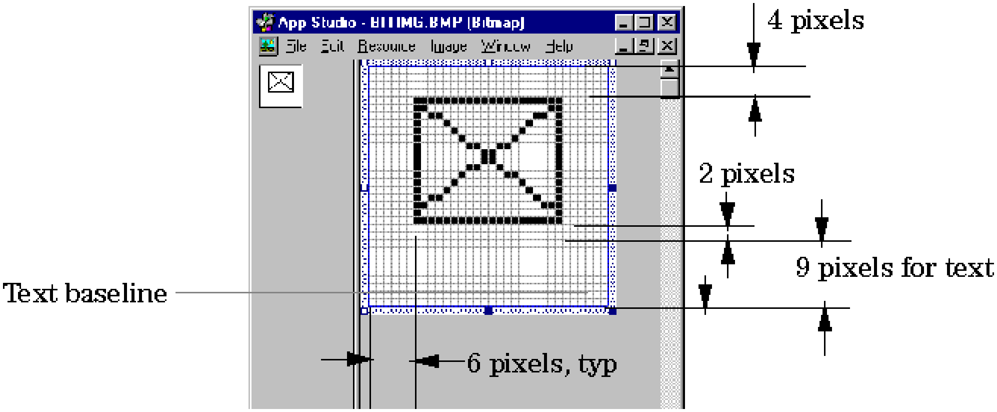
To design pictorial representations, such as icons or other graphics, begin by defining the graphic's purpose and use. How will the graphics help the users finish a task? Graphics are used to support or illustrate the user's task rather than compete with or distract from the task.

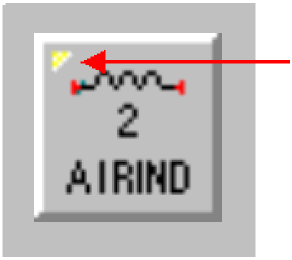
Consistency is important in the design of graphic images. Make the scale, orientation, and color consistent with other related objects, and fit the graphics into the overall environment in which they appear. In addition, make sure you provide sufficient contrast for your images so that users can identify different elements or details of the images.

Microsoft defines three icon sizes: 16 x 16 pixels, 32 x 32 pixels, and 48 x 48 pixels.

Palette Icons

ADS palette bitmaps are 32 x 32. Please review the following guideline for creating palette bitmaps.


App Studio - BITIMG.BMP (Bitmap)
File Edit Resource Image Window Help
4 pixels
2 pixels
9 pixels for text
Text baseline
6 pixels, typ



The white/yellow dithered triangle in upper left corner is the graphic used on Circuit bitmaps to distinguish them from Signal Processing. On a 32 X 32 bitmap the triangle is 4 X 4 pixels and bleeds off the bitmap on the top left.

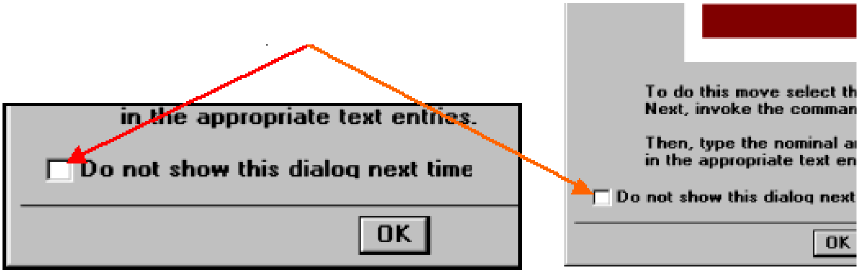
Helping New Users

Not all DesignGuides users will be knowledgeable ADS users. It might be necessary to provide these users with additional help as they learn ADS. Help documentation is available for new users, but many users will start using ADS without referring to documentation. The user interface must guide them through the process.

Review the following guidelines for helping users through their “first hour” experience with ADS and DesignGuides.

Throwaway Dialog Boxes

Use throwaway dialog boxes to help users navigate through the interface. A throwaway dialog box contains a checkbox that allows the user to hide the dialog box from view.



in the appropriate text entries.

□ Do not show this dialog next time

To do this move select th
Next, invoke the comman

Then, type the nominal a
in the appropriate text en

□ Do not show this dialog next

Following are locations in the DesignGuide user interface where throwaway dialog boxes might be useful.

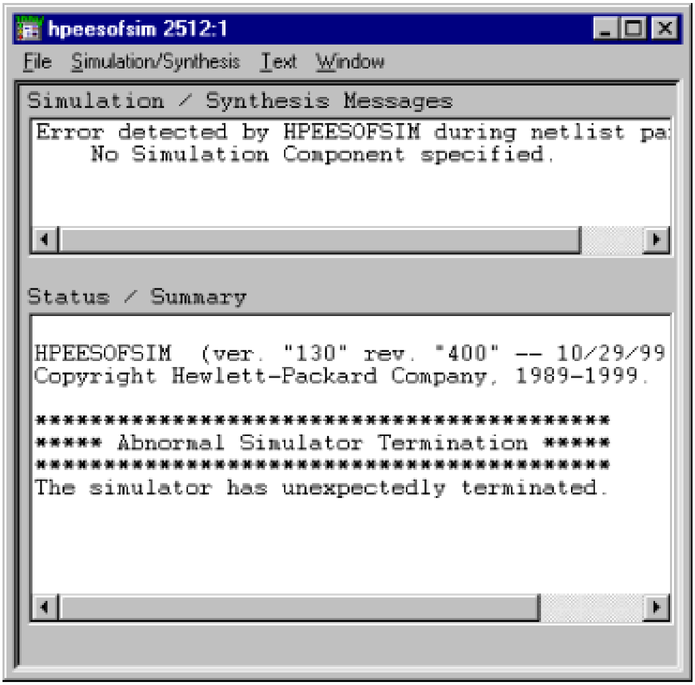
- Telling users how to edit component parameters (both onscreen and dialog box editing)
- Locating and inserting components from palettes
- Running a simulation
- Locating a dataset and applying it to an open data display window
- Moving markers in data display windows
- Undoing the *sticky cursor* after placing components
- Editing color and line weight parameters for polygons

Throwaway dialog boxes are not the same as wizards. Wizards are self-contained tools that guide users through a process with little or no interaction with other parts of the user interface. Wizards are not educational tools.

Information Dialog Box

If users start processes that do not work within the DesignGuide environment, providing informational messages will help them learn to work within the DesignGuide framework. For example, a user with ADS experience, while within the Passive DesignGuide, may start a simulation using the standard ADS menu commands.

The following shows first a current message, then a possible replacement message



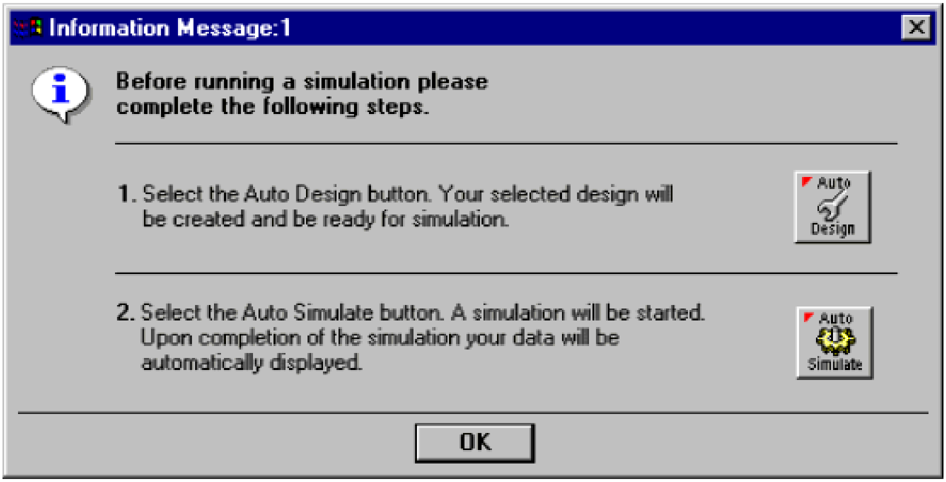
Simulation / Synthesis Messages

```
Error detected by HPEESOFSIM during netlist pa
    No Simulation Component specified.
```

Status / Summary

```
HPEESOFSIM  (ver. "130" rev. "400" -- 10/29/99
Copyright Hewlett-Packard Company, 1989-1999.

******************************************
***** Abnormal Simulator Termination *****
******************************************
The simulator has unexpectedly terminated.
```



Before running a simulation please complete the following steps.

1. Select the Auto Design button. Your selected design will be created and be ready for simulation.

2. Select the Auto Simulate button. A simulation will be started. Upon completion of the simulation your data will be automatically displayed.

Release Checklist

Before checking in a DesignGuide, review the following checklist.

- If you make schematic or data display windows that use an entire 1280x1024 display, the windows will come up off the screen on smaller displays
- Delete any schematic, data displays and datasets that are not important to your DesignGuide. Less is better than more.
- All schematic and data displays should have a title in black.
- On data displays, avoid using full path names. They make plot axes harder to read, and they make data display files difficult to re-use.
- Change the labels on axis labels. 10 ns is better than 0.000000010 sec. Also, take the time to make axis labels larger so they may be read more easily.
- Every data display should have the same name as the schematic from which the data was generated. Dataset names should be no longer than 15 characters.
- Whenever possible, minimize dataset sizes so that users will not fill up their disk drives with ADS data. Use coarse sweeps and don't sweep over larger ranges than is necessary.
- Check for any of the following files, and remove them if found:
  - Core files
  - Debug files
  - Files with a .bak suffix (especially in the networks directory)
  - Files with a .sync suffix
- It is best to create archive file names that are the same as project names. For example, *ModSources_prj* should be archived as *ModSources_prj.zap*.
- Ideally, all DesignGuides should have note saying how long they take to simulate on a particular platform. This is important for DesignGuides that take longer than several minutes to run. Simulation time is a key issue with customers.
- Turn off the grid on schematics. You will have to save the schematic *.prf* file after turning off the grid. This makes schematics much easier to read.

Writing for DesignGuide Screens

Use a conversational, rather than instructional, writing style for the text you provide on DesignGuide screens. The following guidelines can be used to assist you in writing the textual information.

- Use words like *you* and *your*.
- Start most questions with phrases like “Which option do you want…” or “Would you like…” Users respond better to questions that enable them to do a task than being told what to do. For example “Which layout do you want?” works better in wizards than “Choose a layout.”
- Use contractions and short common words. In some cases, it may be acceptable to use slang, but you must consider how this affects localization when doing so.
- Avoid using technical terminology that may be confusing to a novice user.
- Try to use as few words as possible. For example, the question “Which resistor do you want to use for this schematic?” could be written simply as “Which resistor do you want?”
- Keep the writing clear, concise, and simple, but remember not to be condescending.

DesignGuide Use Models

While DesignGuides have been created to bypass some of the more labor-intensive areas of ADS, they should not stray too far from the basic ADS use model (create a circuit, analyze the circuit and view the results of the analysis). Review the following use models before you create your DesignGuide.

DesignGuide Use Models Examples

Following are examples of the various types of use models.

Load-and-go-use model

1. Users make selections from the DesignGuide pulldown menu.
2. DesignGuide schematic and data display windows are open.
3. Users modify circuit parameters and/or insert new circuit components.
4. Users run simulations using the standard ADS simulation controls.
5. Users view the results in the data display windows

Note In some cases, users must select dataset information before viewing simulation results

Specify-and-go-use-model

1. Users open a DesignGuide and are asked to specify the type of circuit desired via a tab format dialog box
2. A Schematic window opens, displaying the specified circuit
3. Users enter and/or modify circuit parameters based on DesignGuide instructions or current knowledge
4. Users run simulations from the DesignGuide menu or from ADS simulation commands.
5. Users view simulation results in the DesignGuide Data Display window using the *View Simulation Results* menu selection.
6. Experienced users may create Data Displays to meet their specific needs. The PLL DesignGuide uses this model.

The Passive DesignGuide model

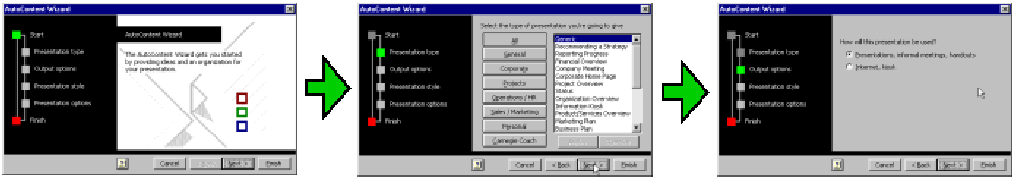
1. Users select a palette from the Passive DesignGuide menu.
2. Users insert circuit components from the Passive Structures palette.
3. Users enter and/or modify the component circuit parameters.
4. Users select the *Auto Design* palette button and create a circuit.
5. Users may view a sub-circuit of the new design.
6. Users select the *Auto Simulate* palette button to run a simulation.
7. A data display window automatically opens showing the simulation results.

DesignGuide Wizards

Wizards are being considered for and implemented in many Agilent EEsof applications to help users through complex and/or infrequently performed tasks. As wizards are designed and implemented for our products, it is important to provide a well designed and consistent user interface.

What is a Wizard?

A *wizard* is a special form of user assistance that automates a task through a dialog with the user. Wizards help the user accomplish tasks that can be complex and require experience. Wizards can automate almost any task. They are especially useful for complex or infrequent tasks that the user may have difficulty doing.



The AutoContent Wizard gets you started by providing ideas and an organization for your presentation.

Select the type of presentation you're going to give

How will this presentation be used?

● Presentations, informal meetings, handouts

○ Internet, kiosk

When Not to Use a Wizard

Wizards are not well-suited to teach a user how to do something. Although wizards assist the user in accomplishing a task, they should be designed to hide many of the steps and much of the complexity of a given task. Similarly, wizards are not intended to be used for tutorials; wizards should operate on real data. For instructional user assistance, consider task Help or tutorial-style interfaces.

Wizards Should Not Mask Poor Designs

Do not rely on wizards as a solution for ineffective designs; if the user relies on a wizard too much it may be an indication of an overly complicated interface, not good wizard design. In addition, consider using a wizard to supplement, rather than replace, the user's direct ability to perform a specific task. Unless the task is fairly simple or done infrequently, experienced users may find a wizard to be inefficient or not provide them with sufficient access to all functionality.

Designing Wizards

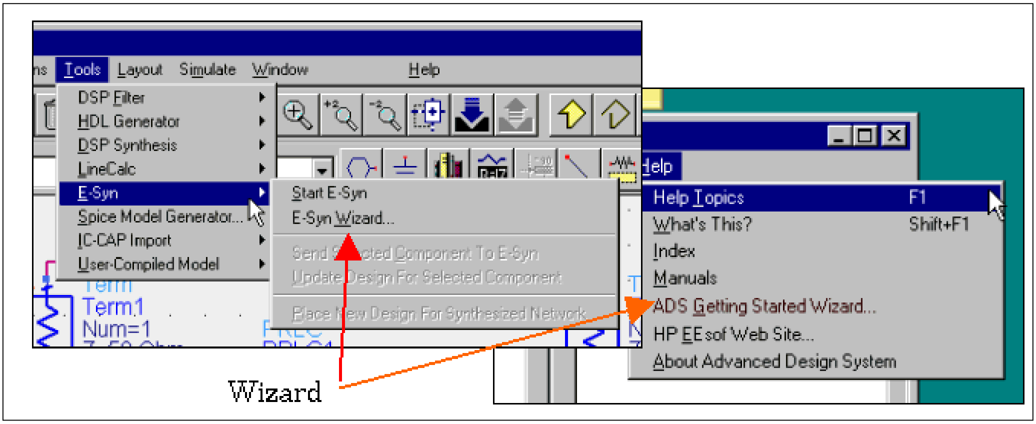
The following sections provide guidelines on designing effective wizards.

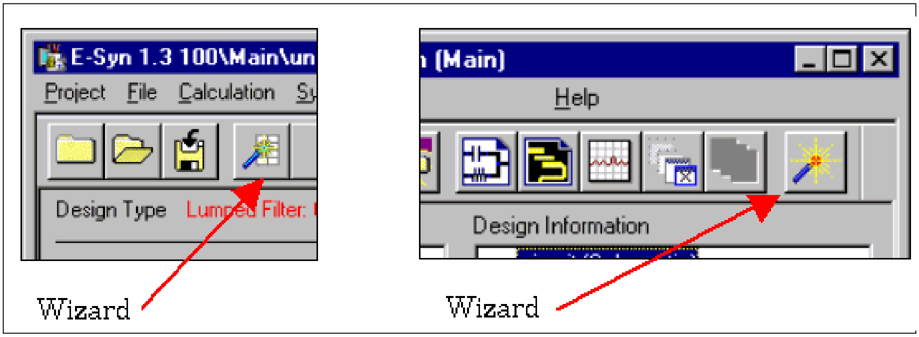
Locating wizards

Wizards may not always appear as an explicit part of the Help interface. You can provide access to them in a variety of ways, including toolbar buttons or even specific icons.

- Where to access wizards
  - Startup dialog boxes
  - Menus
  - Toolbar icons
  - Templates
- Don't force users to use wizards. Wizards should be accessed at the users discretion.

.





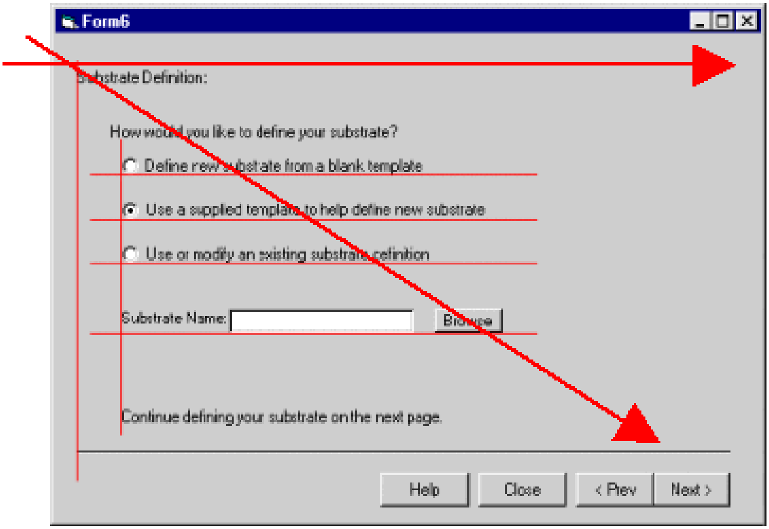
Layout of Controls

The layout of controls should follow standard dialog box design guidelines:

- Controls are aligned to read from left to right.
- Controls are aligned and spaced to provide logical grouping that visually guide users through the tasks.

- The number of controls in a window should be minimized

.



Substrate Definition:

How would you like to define your substrate?

○ Define new substrate from a blank template

● Use a supplied template to help define new substrate

○ Use or modify an existing substrate definition

Substrate Name:

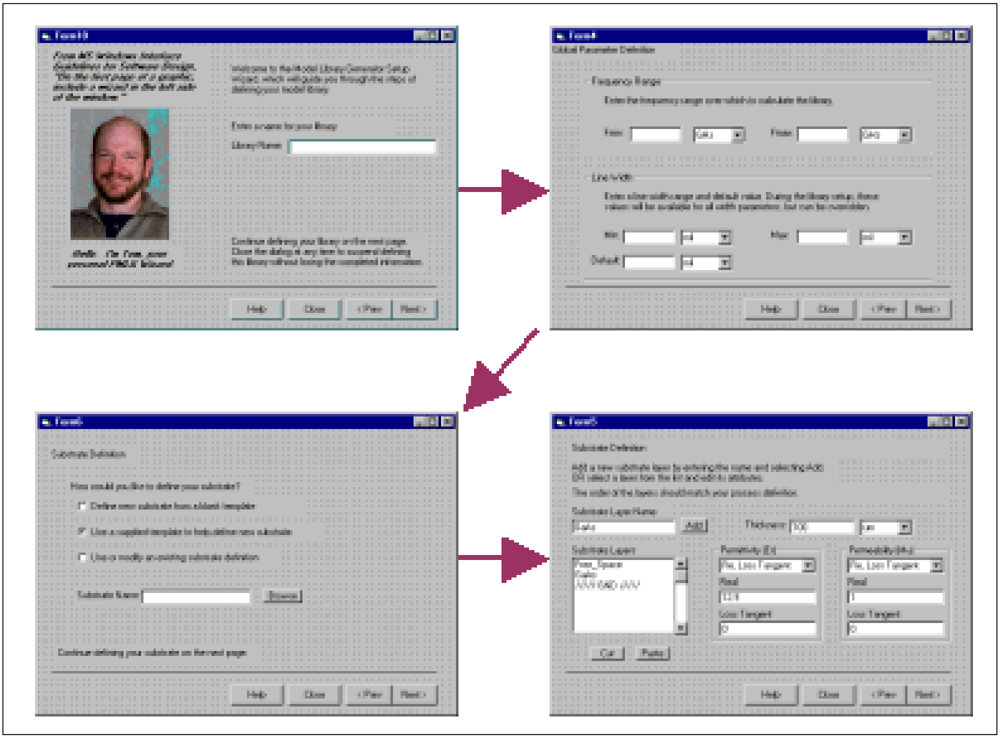
Continue defining your substrate on the next page.

Page Layout Consistency

Provide users with a consistent view of the wizard:

- Wizard windows should be a consistent size throughout.
- Include default values or settings for all controls where possible.
- Resist the urge to take users out of the wizard into other user interfaces to complete a task

.



Form10

From MS Windows Interface Guidelines for Software Design: "On the first page of a graphic, include a wizard in the left side of the window."

Hello - I'm Tom, your personal PMLG Wizard!

Welcome to the Model Library Generator Setup wizard, which will guide you through the steps of defining your model library.

Enter a name for your library

Library Name:

Continue defining your library on the next page. Close the dialog at any time to suspend defining the library without losing the completed information.

Form4

Global Parameter Definition

Frequency Range

Enter the frequency range over which to calculate the library.

From: GHz Thru: GHz

Line Width

Enter a line width range and default value. During the library setup, these values will be available for all width parameters, but can be overridden.

Min: mil Max: mil

Default: mil

Form6

Substrate Definition

How would you like to define your substrate?

○ Define new substrate from a blank template

● Use a supplied template to help define new substrate

○ Use or modify an existing substrate definition

Substrate Name

Continue defining your substrate on the next page.

Form5

Substrate Definition

Add a new substrate layer by entering the name and selecting Add. OR select a layer from the list and edit its attributes.

The order of the layers should match your process definition.

Substrate Layer Name: GaAs

Thickness: 100 um

Substrate Layers:
Free_Space
GaAs
//// GND ////

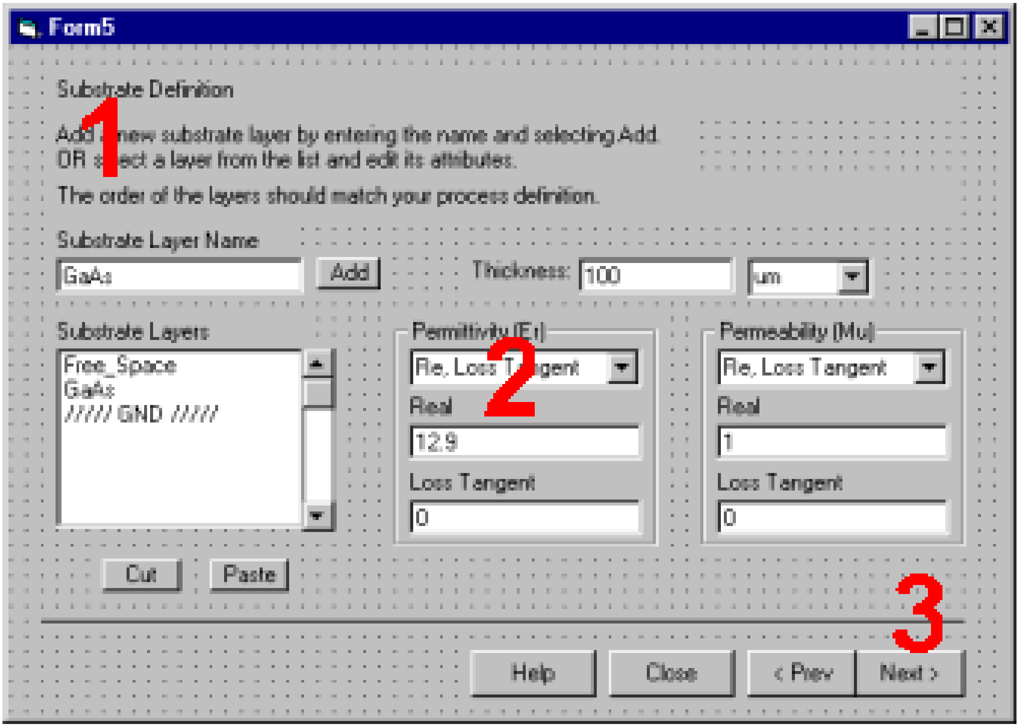
Permittivity (Er): Re, Loss Tangent
Real: 12.9
Loss Tangent: 0

Permeability (Mu): Re, Loss Tangent
Real: 1
Loss Tangent: 0

Page Layout Task Flow

As shown in the illustration that follows, the recommended page layout and task location flow is as follows:

1. The user is provided with information about the task and directions for completing the task.
2. The user enters data relevant to completing this part of the task.
3. The user makes decisions to move on to the next step, review previous steps, or exit the wizard. The user can get help at any time.



Substrate Definition

Add a new substrate layer by entering the name and selecting Add OR select a layer from the list and edit its attributes.

The order of the layers should match your process definition.

Substrate Layer Name: GaAs

Thickness: 100 um

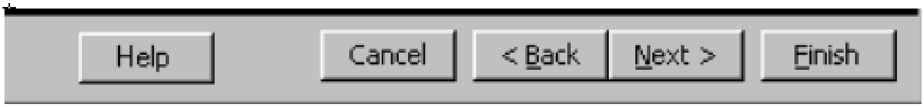
Substrate Layers:
Free_Space
GaAs
///// GND /////

Permittivity (Er): Re, Loss Tangent
Real: 12.9
Loss Tangent: 0

Permeability (Mu): Re, Loss Tangent
Real: 1
Loss Tangent: 0

Command Buttons

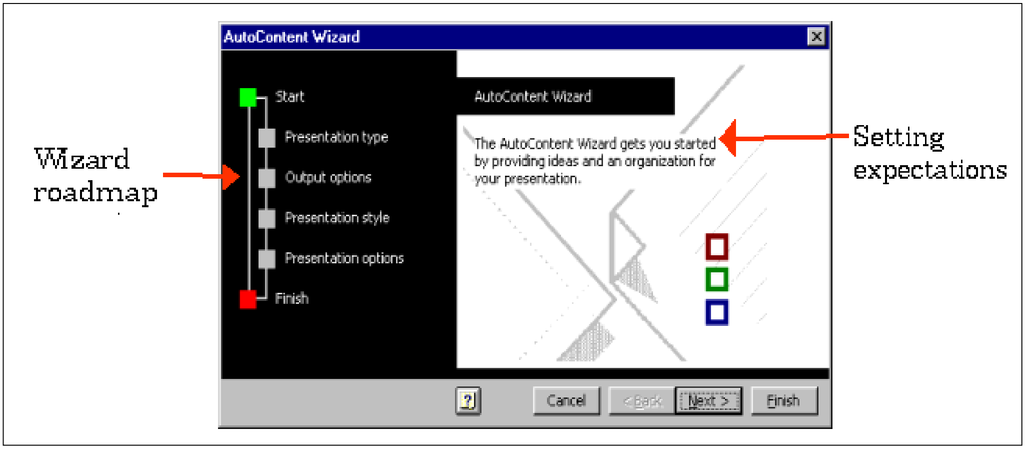
The following command buttons should be used to allow the user to navigate through the wizard. Resist the urge to add additional buttons to the command button location.



- **< Back** returns to the previous page. (Remove/disable the button on first page.)
- **Next >** moves to the next page in sequence, maintaining whatever settings the user provides in the previous pages.
- **Finish** applies user-supplied or default settings from all pages and completes the task.
- **Cancel** discards any user-supplied settings, terminates the process, and closes the wizard window.

Designing the First Wizard Page

The first page should provide a point of reference (a graphic or roadmap) for users. Set positive user expectations for the upcoming task.



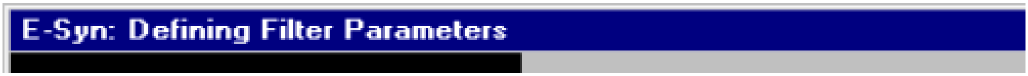
The AutoContent Wizard gets you started by providing ideas and an organization for your presentation.

Wizard roadmap

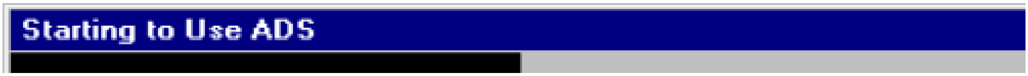
Setting expectations

Wizard Identification

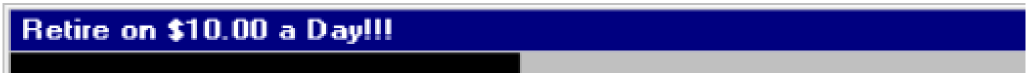
- Use the title text of the wizard page to clearly identify the purpose of the wizard.
- Because wizards are secondary windows, they should not appear in the taskbar



E-Syn: Defining Filter Parameters



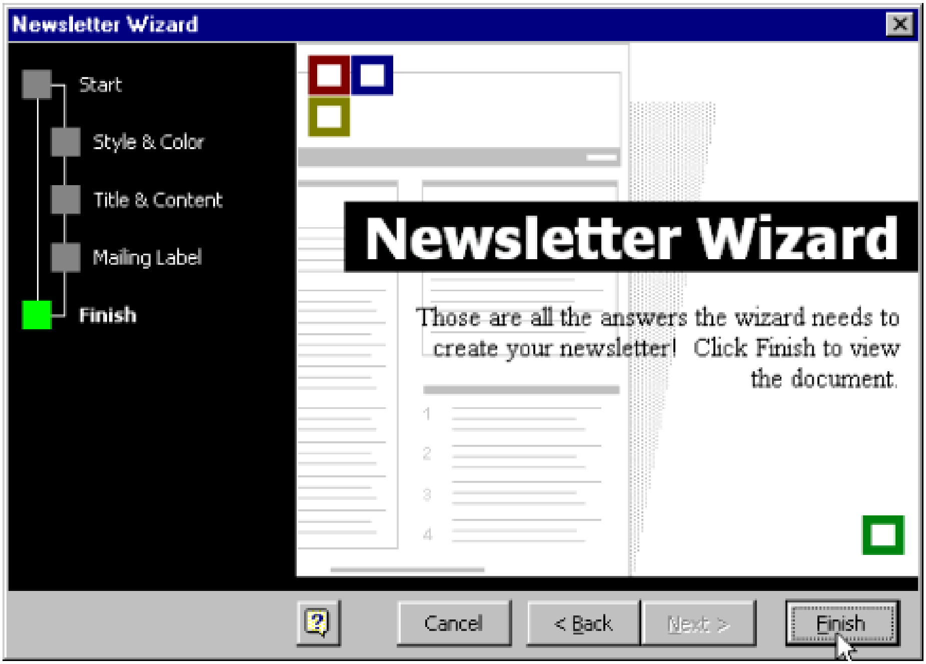
Starting to Use ADS



Retire on $10.00 a Day!!!

Finishing Wizards

You can include the *Finish* button at any point that the wizard can complete the task. On the last page, indicate that the task is completed and instruct the user to click the *Finish* button



Those are all the answers the wizard needs to create your newsletter! Click Finish to view the document.

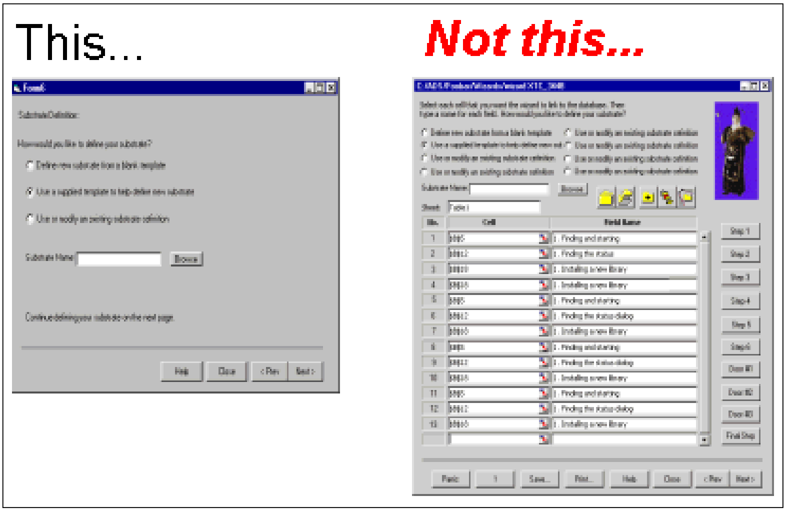
Wizard Do’s and Don’ts

- Minimize the number of pages that require the display of a secondary window. Novice users are often confused by the additional complexity of secondary windows.
- Avoid a wizard design that requires the user to leave the wizard to complete a task. Less experienced users are often the primary users of a wizard. Asking them to leave the wizard to perform a function can make them lose their context.

- Make it visually clear that user-interface elements that are part of a graphic illustration on a wizard page are not interactive.
- Avoid advancing pages automatically. Wizards are intended to allow the user to be in control of the process.
- Display a wizard page so that the user can recognize it as the primary point of input. The page may need to be displayed over its parent window.
- Make certain that the design alternatives offered by the wizard provide the user with positive results.
- Make certain that it is obvious how the user can proceed when the wizard has completed the process.
- If a large amount of data is entered during the task, consider providing the user with a summary view of all the data.

Final Thought on Designing Wizards

Design your wizard pages to be easy to understand. It is important that users immediately understand what a wizard is about so they don't feel like they have to read it very carefully to understand what they have to answer. It is better to have a greater number of simple pages with fewer choices than a smaller number of complex pages with too many options or text.



Substrate Definition:

How would you like to define your substrate?

○ Define new substrate from a blank template

● Use a supplied template to help define new substrate

○ Use or modify an existing substrate definition

Substrate Name: [ ] Browse

Continue defining your substrate on the next page.

ADS Schematic Symbol & Bitmap Creation

Standards have been established for the creation of schematic symbols for ADS schematics and the bitmaps that represent each symbol. The bitmaps are located in the component palettes. To maintain a consistent look and feel throughout the product, designers should follow these standards when creating schematic symbols and bitmaps for DesignGuides.

Creating the Analog RF Symbol

Following are guidelines on creating the symbols for Analog RF components.

13-Step Procedure for Analog RF Symbol Creation

1. When creating an Analog RF Symbol, try to leverage a similar existing symbol.
2. Open the existing similar symbol file.
3. Select the command **Select >Select All**.
4. Copy the symbol to the buffer (**Edit** > **Copy/Paste** > **Copy to Buffer**).
5. Create a new file (**File** > **New**).
6. Change the mode to *Edit Schematic* (**View** > **Create/Edit Schematic**).
7. Paste the Symbol to grid (**Edit** > **Copy/Paste** > **Paste from Buffer**).
8. *Important*: Align pin 1 on a grid point.
9. Set origin to Pin 1 (**Edit** > **Modify** > **Set Origin**).
10. Using the ADS design tools (Layer Editor and drawing tools), edit the symbol.
11. Check to make sure all pins are on gridpoints.
12. Save the Symbol (**File** > **Save As**...) using the symbol naming convention –SYM_*Filename*.dsn.

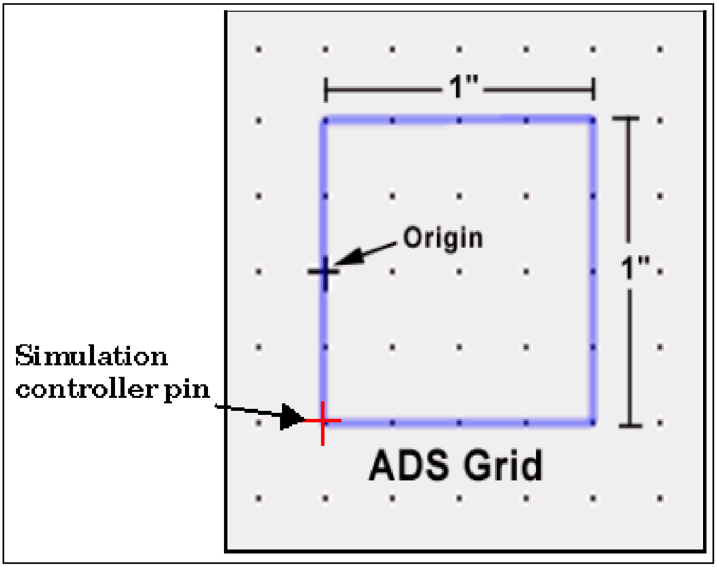
Once you have saved and closed a newly created symbol, to be able to view it again, follow these steps.

1. Open the symbol file (**File** > **Open <new filename**>) You should be able to see the symbol. If you cannot, the design type may not be set correctly if an existing design was not leveraged. You will need to modify the design type to be -1 in the design file. To do this, make sure the number after the line starting with 10 is -1, e.g. (10 -1 "SYM_abc" 2 954523902 0 0 0 0).
2. To view the symbol, change the mode to *Edit Schemati*c (**View > Create/Edit Schematic**)

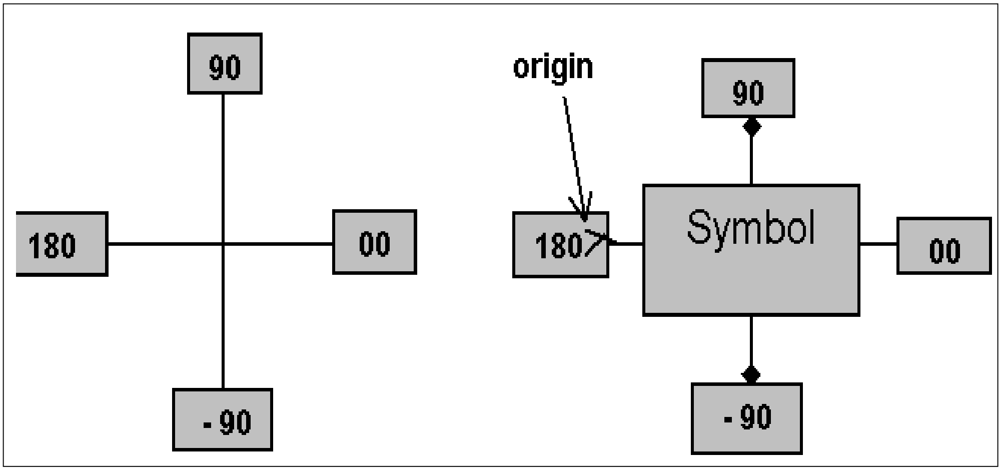
Symbol Grid

Following are guidelines on the symbol grid.

- Symbols most commonly fit into a 1-inch by 1-inch square area.
- The origin of the symbol is always on a grid point.
- The origin is where pin 1 is placed.
- Simulation controller pin is located on the lower left corner (anything without pins) of the symbol, as shown in the following illustration.



The following illustration shows pin orientation angles.



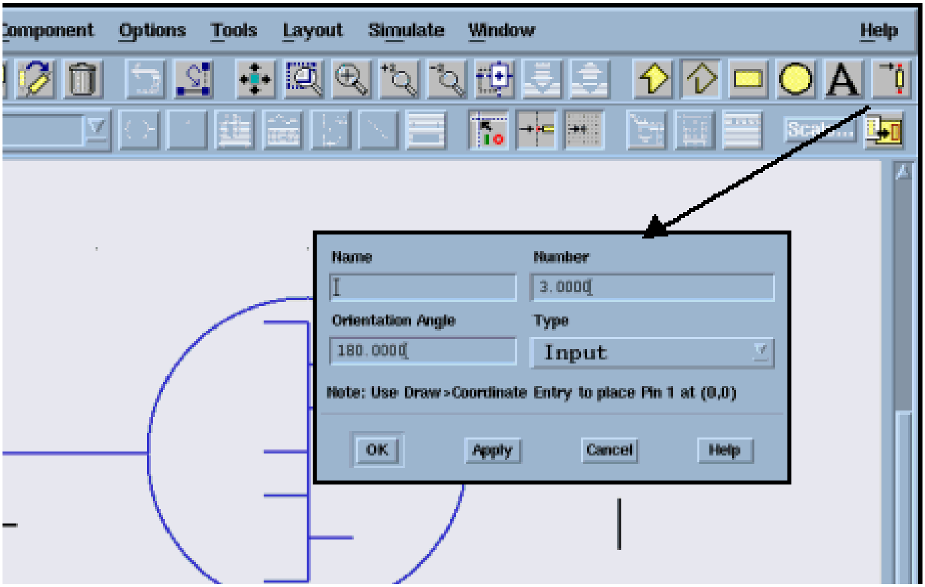
Symbol Pin-Out

Following are guidelines on the symbol pin-out.

- Symbols most commonly fit into a 1-inch by 1-inch square area.
- The origin of the symbol is always on a grid point.
- the origin is where pin 1 is placed.
- Simulation controller pin is located on the lower left corner (anything without pins) of the symbol.

Symbol Pin Dialog Box (Analog RF)

Following are guidelines for use of the symbol pin dialog box for Analog RF.



Name

Number 3.0000

Orientation Angle 180.0000

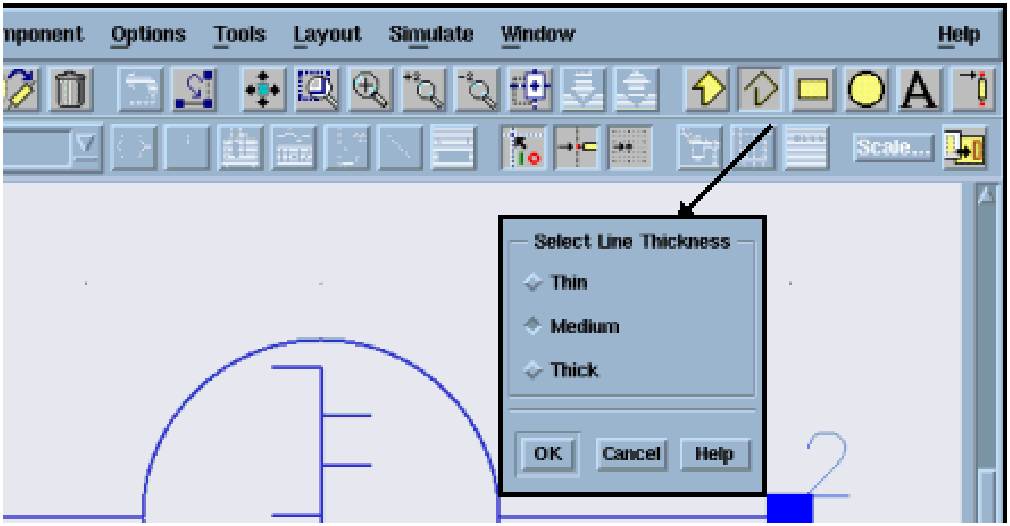
Type Input

Note: Use Draw>Coordinate Entry to place Pin 1 at (0,0)

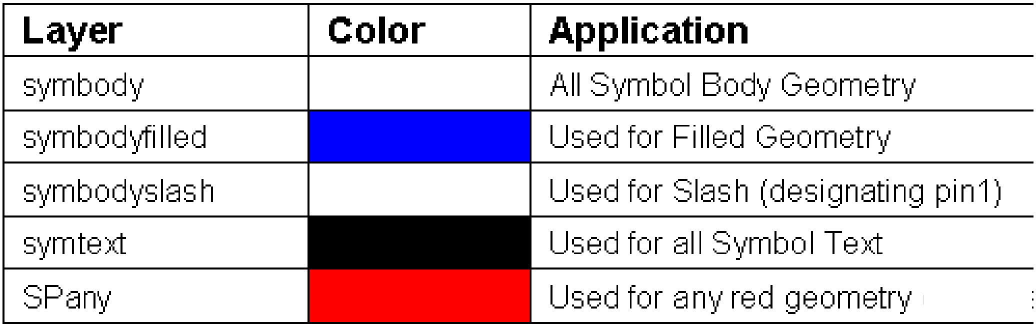
1. Pins are not named unless otherwise specified
2. See Pin Orientation angle in the preceding section “Symbol Grid” on page 10-54
3. See Symbol Pinout in the preceding section “Symbol Pin-Out” on page 10-55
4. Label Each pin Input/Output unless otherwise specified

Line Thickness (Analog RF)

When designing geometry for Analog RF symbols, always uses line thickness of Medium.



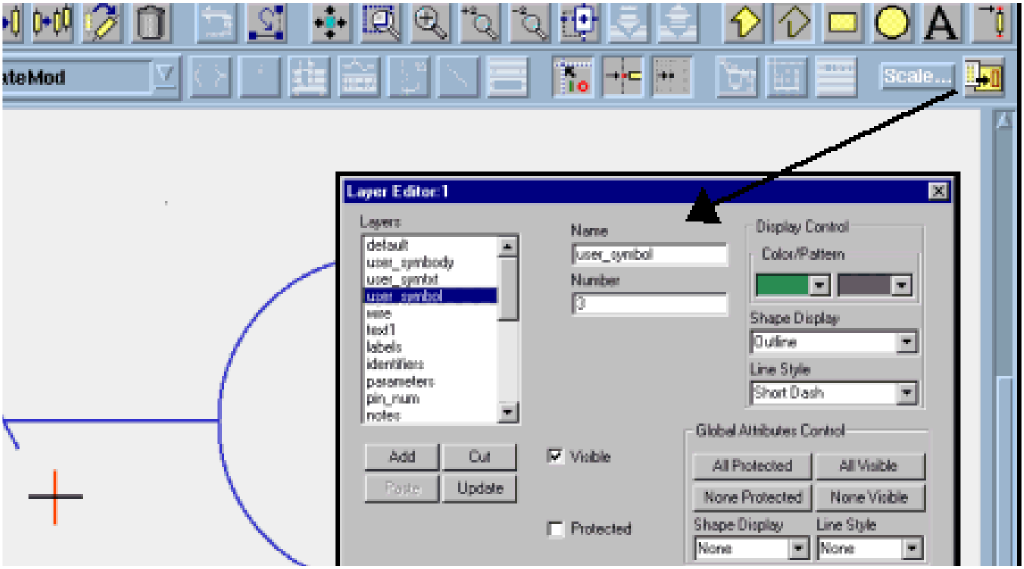
Note The Polyline Thickness Dialog Box Automatically comes up when using the geometry tools. (select *Medium*).



Layer	Color	Application
symbody		All Symbol Body Geometry
symbodyfilled	🟦	Used for Filled Geometry
symbodyslash		Used for Slash (designating pin1)
symtext	⬛	Used for all Symbol Text
SPany	🟥	Used for any red geometry

Changing Layer/Color

The Layer Editor dialog box is used for changing layer and color.



Layer Editor:1

Name: user_symbol

Number: 0

Shape Display: Outline

Line Style: Short Dash

☑ Visible

□ Protected

Global Attributes Control

Shape Display: None

Line Style: None

Creating the DSP Symbol

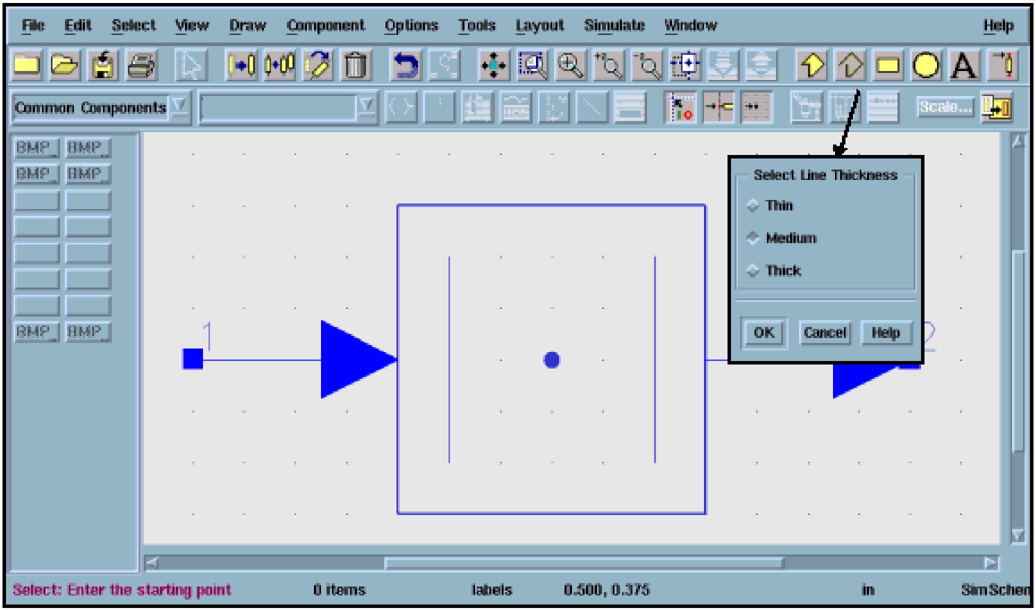
13-Step Procedure for DSP Symbol Creation

1. When creating a DSP Symbol, try to leverage a similar existing symbol.
2. Open the existing similar symbol file.
3. Select the command *Select All* (**Select > Select All**).
4. Copy the symbol to the buffer (**Edit** > **Copy/Paste** > **Copy to Buffer**).
5. Create a new file (**File** > **New**).
6. Change the mode to *Edit Schematic* (**View** > **Create/Edit Schematic**).
7. Paste the Symbol to grid (**Edit** > **Copy/Paste** > **Paste from Buffer**).
8. *Important*: Align pin 1 on a grid point.
9. Set origin to Pin 1 (**Edit** > **Modify** > **Set Origin**).
10. Using the ADS design tools (Layer Editor and drawing tools), edit the symbol.
11. Check to make sure all pins are on gridpoints.
12. Save the Symbol (**File** > **Save As…**) using the symbol naming convention –SYM_*Filename*.dsn
13. Once you have saved and closed a newly created symbol, to be able to view it again, follow these steps.
    - Open the symbol file (File > *Open <new filename>*). You should be able to see the symbol. If you cannot, the design type may not be set correctly if an existing design was not leveraged. You will need to modify the design type to be -1 in the design file. To do this, make sure the number after the line starting with 10 is -1 e.g.

    (10 -1 “SYM_abc” 2 954523902 0 0 0 0).

    - To view the symbol, change the mode to Edit Schematic (View > Create/Edit Schematic).

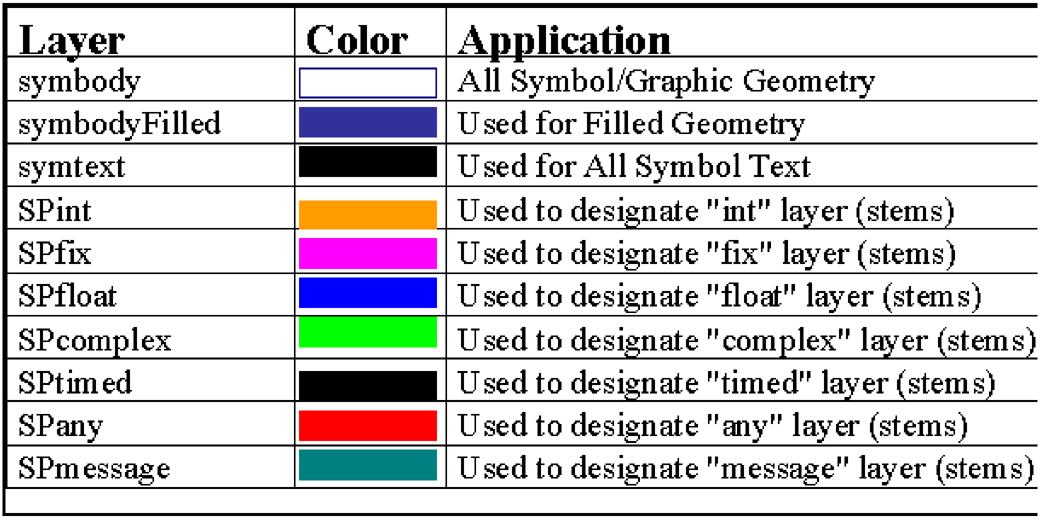
Line Thickness (DSP)



Select: Enter the starting point 0 items labels 0.500, 0.375 in SimSchem

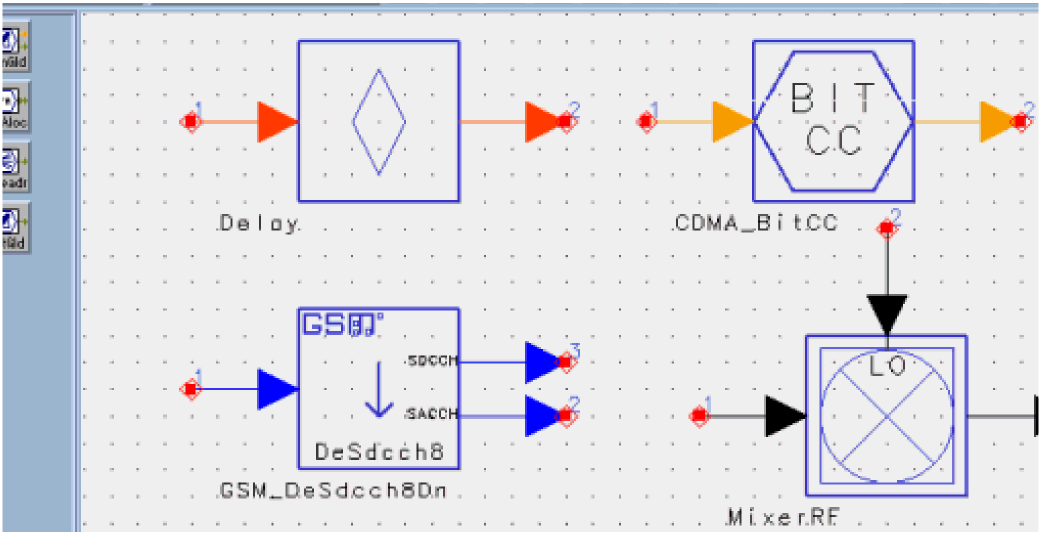
1. When designing geometry for DSP symbols, Use a line thickness of **Thin** for internal symbol geometry.
2. Always use line thickness of *Medium* for symbol bodies.

Layer/Color Scheme (DSP Symbols)



Layer	Color	Application
symbody	□	All Symbol/Graphic Geometry
symbodyFilled	🟦	Used for Filled Geometry
symtext	⬛	Used for All Symbol Text
SPint	🟧	Used to designate "int" layer (stems)
SPfix	🟪	Used to designate "fix" layer (stems)
SPfloat	🟦	Used to designate "float" layer (stems)
SPcomplex	🟩	Used to designate "complex" layer (stems)
SPtimed	⬛	Used to designate "timed" layer (stems)
SPany	🟥	Used to designate "any" layer (stems)
SPmessage	🟦	Used to designate "message" layer (stems)

Sample Naming Conventions for Symbols (DSP)



1. Normal (SYM_*filename*.dsn)
2. CDMA (SYM_CDMA_*filemame*.dsn)
3. GSM (SYM_GSM_ *filemame*.dsn)
4. Hierarchical (SYM_DSN_ *filemame*.dsn)

Creating New Projects

When creating DSP symbol libraries it is often desirable to create a new project. This allows separate DSP symbol libraries to be separated by project.

Note Analog RF components are designed in a single project gemini_prj project.

1. In the ADS Main window, select **File** > **Create New Project.**
2. Add custom setup files to new project directory- (these setup files contain toolbar and layout preferences that have been optimized for symbol creation)
   - *schematic.lay*
   - *schematic.prf*

   Both of these files are on the zip disk in the directory labeled custom.

Symbol Text

1. Symbol text must only be created in single line blocks (no carriage returns).
2. Symbol text that should not rotate needs to be created with Center/Middle justification and with the non-rotating flag ON. After component placement, check that Non-rotating text looks appropriate by rotating the component at different angles (0, 90, 180, and -90).

   Note The following scripts were written for ADS 1.3. There is no guarantee that these scripts will continue to work in future releases of ADS. They should not be necessary if the text was created correctly by following step 1 & 2 above.

These scripts are located on the zip disk in the directory labeled scripts.

1. findtext script (script that checks a batch of symbols for multiple line text blocks)

- Usage ./findtext *filename* or *.dsn
- Returns list of filenames with multiple line text blocks.
- Manually correct text problems.
- Re-center script (script that fixes that changes Bottom/Left justification to Center/Middle and turns Non-rotating flag ON) is not necessary if the text is created correctly to begin with.
- Usage ./recenter *filename* or *.dsn

Hints and Tips

- ADS is not designed as a graphics tool and is a cumbersome interface for drawing. Creating special shapes and geometry often requires creativity.
- Try to become familiar with existing symbols as they may be leveraged to create new symbols.
- When possible try to have engineers supply accurate graphical information (mockup) for each symbol.
- After creation double check symbols to insure quality.

ADS Bitmap Creation

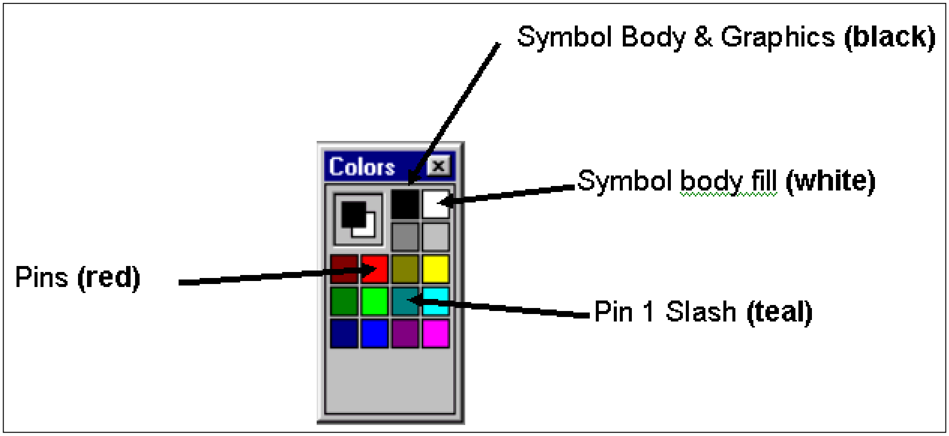
The following sections provide guidelines on creation of bitmaps.

Procedure for Bitmap Creation

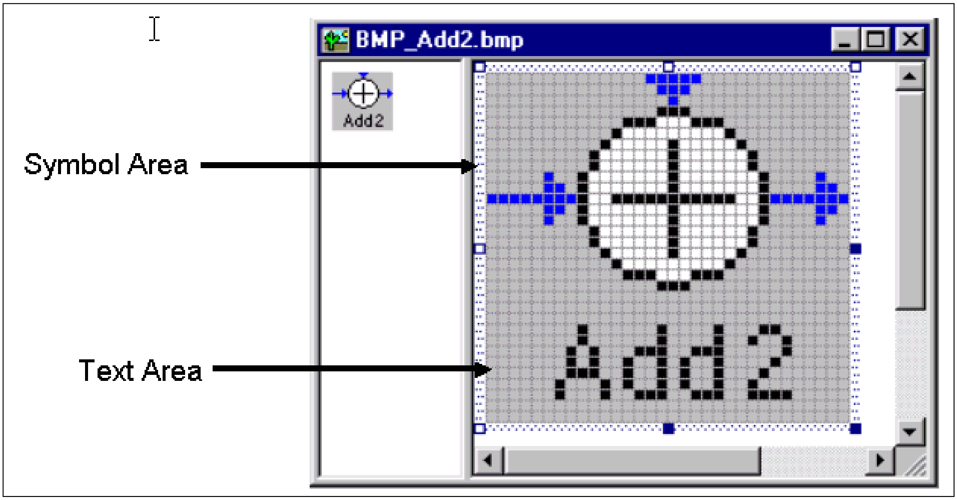
1. When creating a new bitmap, try to leverage a similar existing Bitmap. Be consistent with other Bitmaps that may be in similar family grouping.
2. Add graphical content [*Design Tools*]
3. Remember Analog RF symbols need corner tab.
4. Save the bitmap
5. When finished with new Bitmaps, place them in the appropriate engineers directory on the WLV ‘N’ Drive. The path is:

   *N:\RnD\PUBLIC\*__________

Colors Used in Analog RF Bitmaps

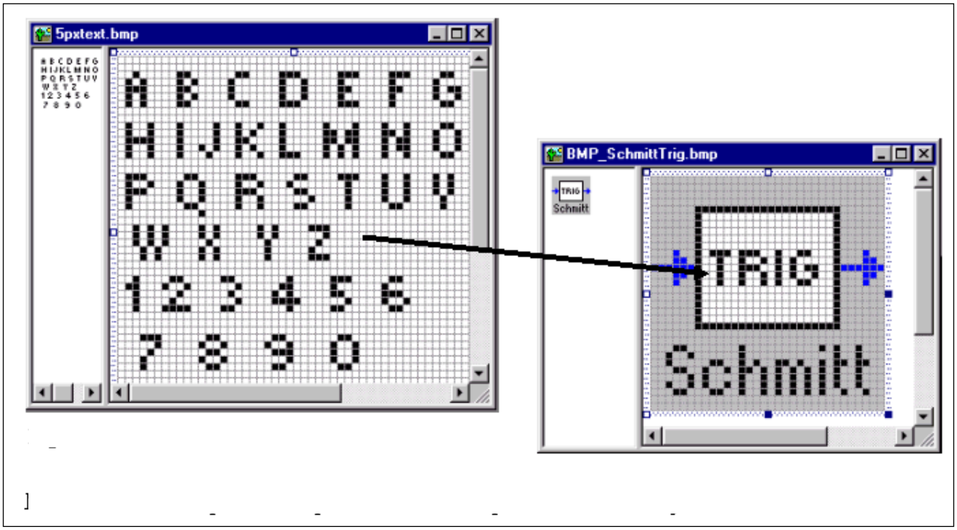

Symbol Body & Graphics **(black)**
Colors
Symbol body fill **(white)**
Pins **(red)**
Pin 1 Slash **(teal)**

32 X 32 Bitmap Layout (DSP)



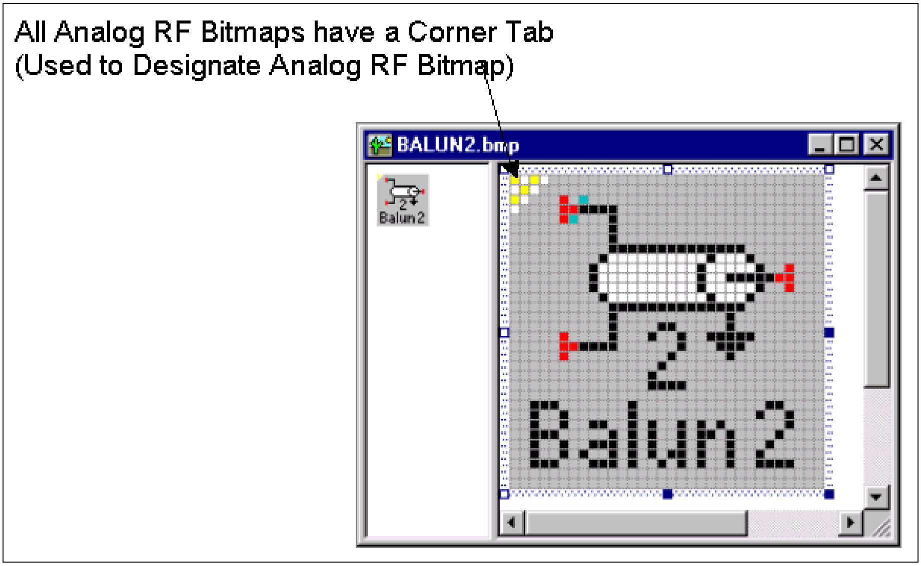
Note Text files 7PXCAPS.bmp and 7PXTXT.bmp are located on zip disk in directory labeled “text”

Bitmap Internal Graphics Use 5 Pixel Text



Note Text files 7pxtext.bmp is located on zip disk in directory labeled “text.”

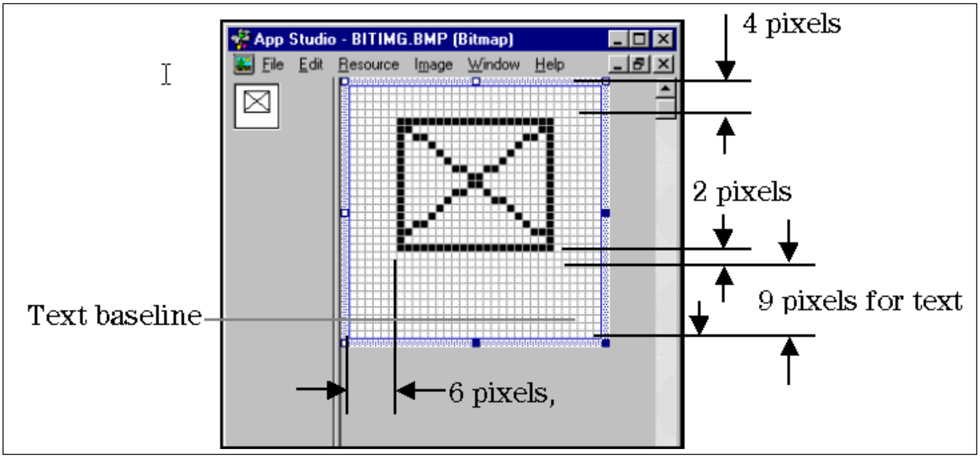
Corner Tab on Analog RF Bitmaps



All Analog RF Bitmaps have a Corner Tab
(Used to Designate Analog RF Bitmap)

32 X 32 DSP Bitmap Grid

This grid is to be used for Analog bitmaps also. The image area can be larger than the box. The text area is to remain the same.



User Testing

User feedback is an important part of the design process. This is especially true for DesignGuides. DesignGuides, along with examples and templates, may be a customer's first exposure to our tools. This "first hour" exposure can influence a users long term view of a product. Please review the following usability processes and let us know how we can help you get user feedback.

Following are user-testing techniques:

- **Heuristic Analysis.** Heuristic evaluation is a systematic inspection of the wizard using a recognized set of design principles. The evaluation is performed by having each evaluator inspect the DesignGuide alone. Findings are reviewed by the project team.
- **Cognitive Walkthrough**. A cross-functional team reviews the usability of the wizard. The results are reviewed by the project team and defects are ranked according to their severity.
- **Low-fidelity user testing**. User testing of an interface using paper prototypes. Good test for task flow and general UI navigation. (3 to 6 users)
- **Medium-fidelity user testing**. First test of the computer interface. Full functionality may not be implemented, but major features may be tested. Good for task flow and specific feature operation. Include documentation if available. (3 to 6 users)
- **High-fidelity user testing**. Final user testing of wizard. Full product functionality available. Users may test any and all parts of the user interface. (3 to 7 users)

Usability Resources

There are many usability resources available, especially on the web. The following resources are recommended.

- Microsoft Windows Usability
    - Microsoft Usability Resources
      http://msdn.microsoft.com/ui/
    - *Windows User Experience: Official Guidelines for User Interface Developers and Designer*s
      http://www.amazon.com/
- UNIX Motif Usability
    - Motif User Interface StyleGuide
      http://www.premier.sco.com/guide/MotifStyleGuide/en_US/TOC.html
- Other usability resources
    - *Macintosh Human Interface Guidelines*
      http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-2.html
    - *IBM – User Centered Design*
      http://www-3.ibm.com/ibm/easy/eou_ext.nsf/publish/2
- ACM SIGCHI (Special Interest Group – Computer Human Interface)
  http://www.acm.org/sigchi/about.html
- Usability Professionals Association
  http://www.upassoc.org/
- Human Factors and Ergonomics Society
  http://hfes.org/
- Don Norman (usability guru)
  http://www.jnd.org/

Bibliography

[1] Apple Computer (1996), *Macintosh Human Interface Guidelines*. Addison-Wesley, Reading, MA. ISBN 0201622165

[2] Helander, M. (Ed.) (1993), *Handbook of Human-Computer Interaction*. Elsevier Science Publishers, Amsterdam, The Netherlands, ISBN 0-444-70536-8

[3] Mayhew, D. J. (1992), *Principles and Guidelines in Software User Interface Design*. Prentice Hall, Englewood Cliffs, NJ, ISBN 0-130721929-6

[4] Microsoft Corporation (1999), *Microsoft Windows User Experience: Official Guidelines for User Interface Developers and Designers*. Microsoft Press, Redmond, WA. ISBN 0-7356-0566-1

[5] Nielsen, Jakob (1993), *Usability Engineering*, AP Professional, Chestnut Hill, MA. ISBN 0-12-518406-9

[6] Norman, D. A. (1988). *The Psychology of Everyday Things*, Basic Books, New York, NY. ISBN 0-465-06709-3

[7] Preece, J. (Ed.) (1998), *A Guide to Usability: Human Factors in Computing*. Addison-Wesley, Harlow, England. ISBN 0-201-62768-X

Index