Getting Started with Design Architect

Software Version 8.5_1
Part Number 059912

November 1995

Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777.

TABLE OF CONTENTS

About This Training Workbook xi

Introduction xi
Manual Organization xii
Related Documents xii
Objectives xiii
Workbook Layout xiii
For the Person in a Hurry xv

Module 1
Design Architect
in the Framework Environment 1-1

Module 1 Overview 1-2
Lesson 1
The Electronic Design Data Model 1-3
Design Architect and the EDDM 1-4
Component Structure
(Conceptual View) 1-6
Component Structure
(Iconic View) 1-8
Component Within a Component 1-10
Design Viewpoint
(Conceptual View) 1-12
Lesson 2
Common Elements of the User Interface 1-15
Window Buttons and Navigator Controls 1-16
Menus and Mouse Buttons 1-18
Palettes 1-20
Softkeys 1-22
Command Window 1-24
Prompt Bars 1-26
Strokes 1-28
Quick Help on Strokes 1-30
Transcript Window 1-32

TABLE OF CONTENTS [continued]

Session Setup 1-34

Lesson 3
Using Notepad to Create and Modify ASCII Files 1-37

Editing Files with Notepad 1-38

Lesson 4
Viewing and Searching Online Documentation 1-41

Online Help 1-42

Opening Online Documents 1-44

Searching and Traveling in Documents 1-46

Lesson 5
Using Design Manager to Copy Objects 1-49

Copying Objects in the Navigator Window 1-50

Soft Prefixes and Location Maps 1-52

Design Object References 1-54

Copying Design Objects 1-56

Moving and Deleting Design Objects 1-58

Checking and Changing Design References 1-60

Lab Exercises 1-63

Exercise 1: Copying the Training Data 1-64

Exercise 2: Using Notepad to Create and Modify ASCII Files 1-71

Exercise 3: Viewing and Searching Online Documents 1-77

Module 2
Creating a Schematic 2-1

Module 2 Overview 2-2

Lesson
Creating a Schematic 2-3

Invoking Design Architect 2-4

Opening a Schematic Sheet 2-6

The Schematic Editor Window 2-8

Elements of a Schematic 2-10

schematic_add_route Palette 2-12

Design Architect Strokes 2-14

TABLE OF CONTENTS [continued]

Placing an Instance on a Sheet 2-16
Mentor Graphics Libraries 2-18
Using the Choose Symbol Option 2-20
The Active Symbol 2-22
Adding Nets 2-24
Net Creation Process 2-26
Autorouting Nets 2-28
Net Connection Rules 2-30
Connecting and Disconnecting Net Vertices 2-32
Naming Nets 2-34
Selection Concepts 2-36
Select Filter 2-38
Using the Select Popup Menus 2-40
Manipulating Objects 2-42
Interwindow Copy and Move 2-44
The Context Window 2-46
Checking the Sheet 2-48
Saving the Sheet 2-50
Schematic Editor Window Status Line 2-52
Using the Component Hierarchy Window 2-54
Using the Component Window 2-56
Lab Overview 2-58
Lab Exercises 2-59
Exercise 1: Creating a Schematic 2-60
Exercise 2: Net Connection Rules 2-70
Exercise 3: Changing the Mouse Selection Filter 2-71
Exercise 4: Browsing a Component in the Component Hierarchy Window 2-73
Exercise 5: Browsing a Component in the Component Window 2-74

Module 3
Creating a Symbol and Adding Properties 3-1

Module 3 Overview 3-2
Lesson 1

TABLE OF CONTENTS [continued]

Creating a Symbol 3-3
Elements of a Symbol 3-4
Opening a Symbol 3-6
Symbol Editor Window 3-8
The symbol_draw Palette 3-10
Setting the Symbol Body Defaults 3-12
Adding Pins 3-14
Checking the Symbol 3-16
Changing Required Checks 3-18
Saving the Symbol 3-20
Lesson 2
Adding Properties 3-23
What is a Property? 3-24
Property Ownership 3-26
Property Types 3-28
Property Text Attributes 3-30
Symbol Property Text Switches 3-32
SLD Properties 3-34
Class Property Values 3-36
Examples of Global Nets 3-38
ground 3-39
VCC 3-39
Attaching Nets to PCB Power Planes 3-39
Other Global Nets 3-39
Common Digital Simulation Properties 3-40
Common Analog Simulation Properties 3-42
Common PCB Layout Properties 3-44
Adding Properties to Symbol Graphics 3-46
Adding "Logical Symbol" Properties 3-48
Reporting On and Deleting
"Logical Symbol" Properties 3-50
Changing Property Values 3-52
Selecting Properties 3-53
Changing Property Values 3-53

TABLE OF CONTENTS [continued]

Changing Property Attributes 3-53
Setting Up Property Text Attributes 3-54
Quick Report on Property Text 3-56
Lab Exercises 3-59
Exercise 1: Creating a Symbol 3-60
Exercise 2: Adding Properties to a Schematic 3-70
Exercise 3: Adding Properties to a Symbol 3-75
Exercise 4: Browsing the my_dff Component in the Component Window 3-82

Module 4
For Continued Learning . . . 4-1

Getting Started Training Workbooks 4-1
Personal Learning Programs 4-1
Instructor-Led Workshops 4-1

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES [continued]

About This Training Workbook

Introduction

This manual is for users of Design Architect who have some knowledge about schematic drawing and electronic design and are familiar with the UNIX environment. This training workbook is designed to provide you with an introduction to the concepts and instructions on how to use Design Architect to create a schematic and symbol. An introduction to how to use the Design Manager is also provided.

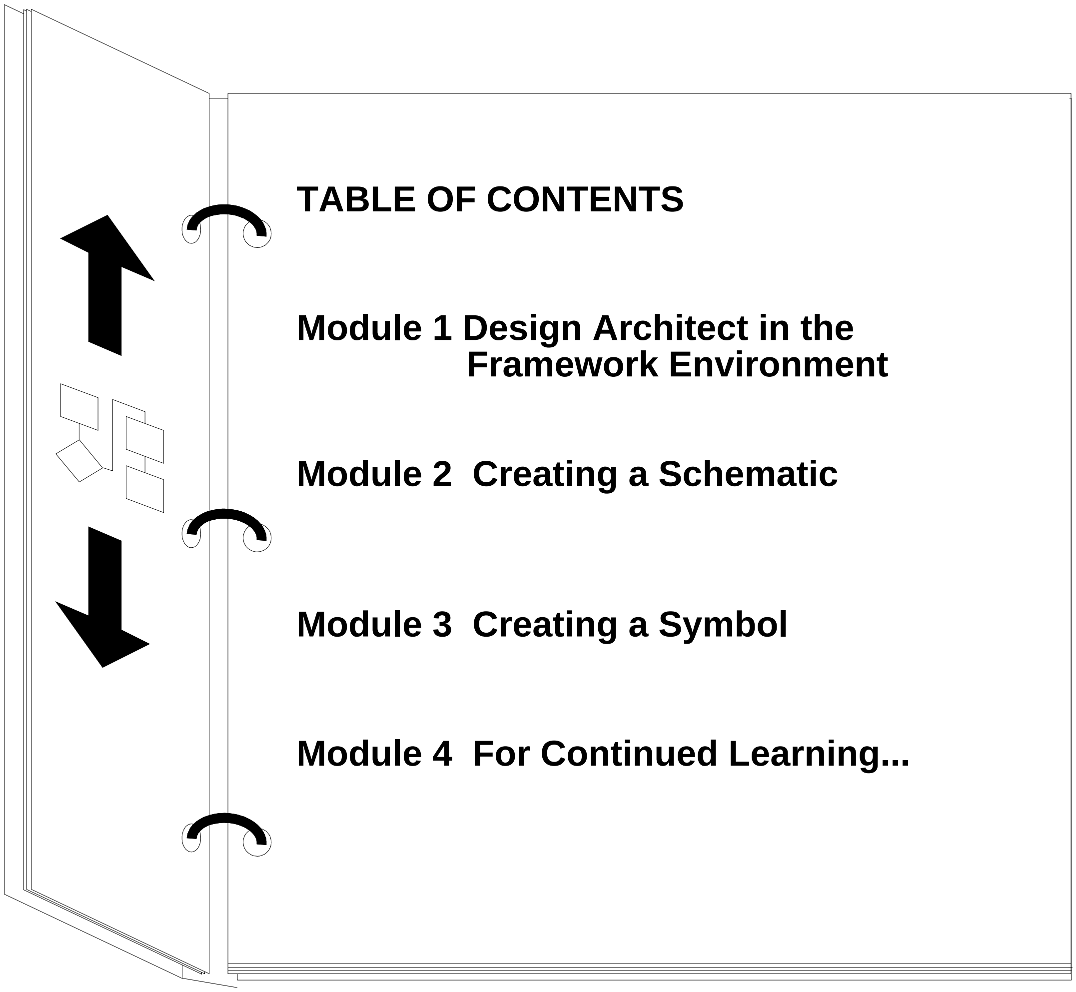
In cases where more information about another tool or application may be desired, a hypertext link is provided for immediate access to that information through the BOLD Browser on-line document viewer.

Note If you are reading this on a color workstation, and you invoked the Bold Browser using the -Color switch, hypertext links are shown in the active color defined for your workstation; by default, the color is blue. If you are reading this on a monochrome workstation, or you invoked the Bold Browser in default mode, hypertext links are enclosed within a box.

Hypertext links may also be contained in pictures and tables, but are not normally shown as hypertext links. In most cases, introductory text will instruct you on using these hypertext links.

When you position the graphic pointer over the hypertext link, you can click the Select mouse button on the hypertext link to go to its destination. For more information about using hypertext links within online manuals, see Navigating Online Documents in the *BOLD Browser User's Manual.*

Manual Organization



Related Documents

Design Architect Training Workbook

Design Architect User's Manual

Design Architect Reference Manual

Objectives

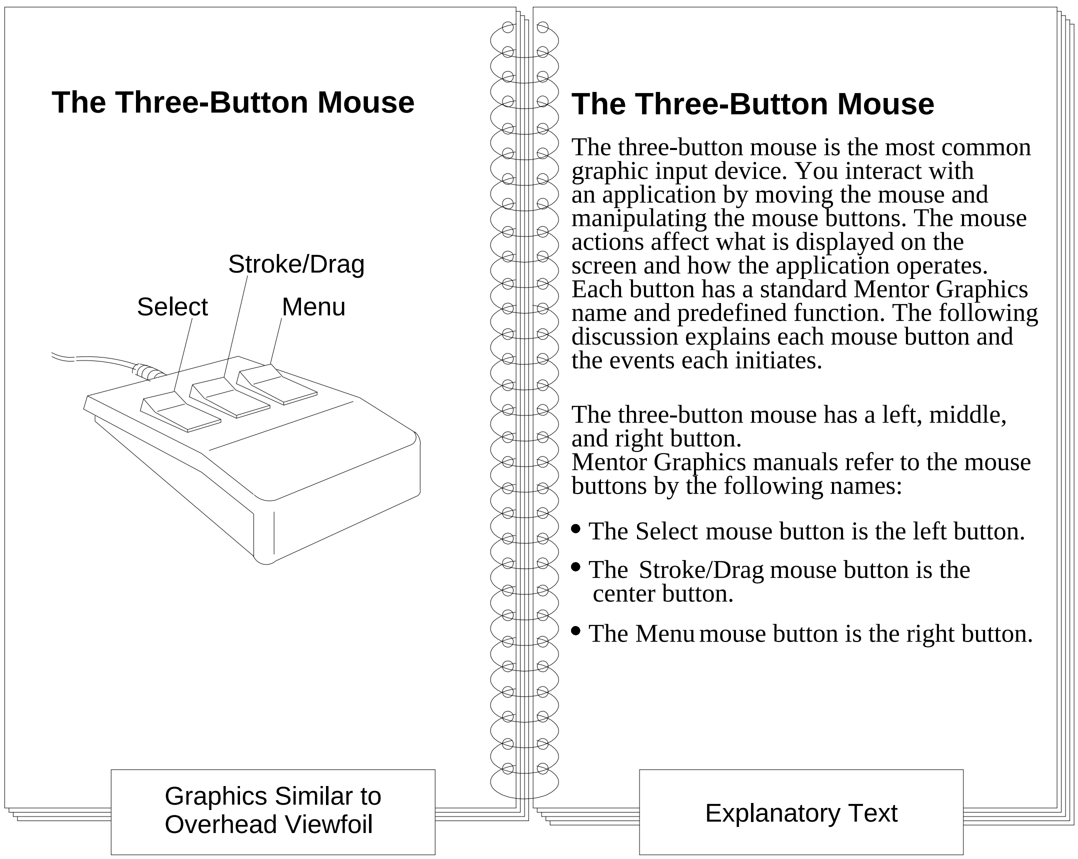
After reading this training workbook and completing the lab exercises, you should be able to do the following:

- Invoke Design Manager and copy design data
- Use the Notepad Text Editor
- Use the BOLD Browser to view online documentation
- Create a Schematic with Design Architect
- Create a Symbol with Design Architect

Workbook Layout

Each module is divided into one of more “Lessons” followed by “Lab Exercises”. A Lesson presents conceptual information. Each concept is presented on two facing pages, as shown in the following illustration. The left page is similar to an overhead viewfoil that an instructor in training class might use. The right page contains written text similar to what an instructor might say about the viewfoil.

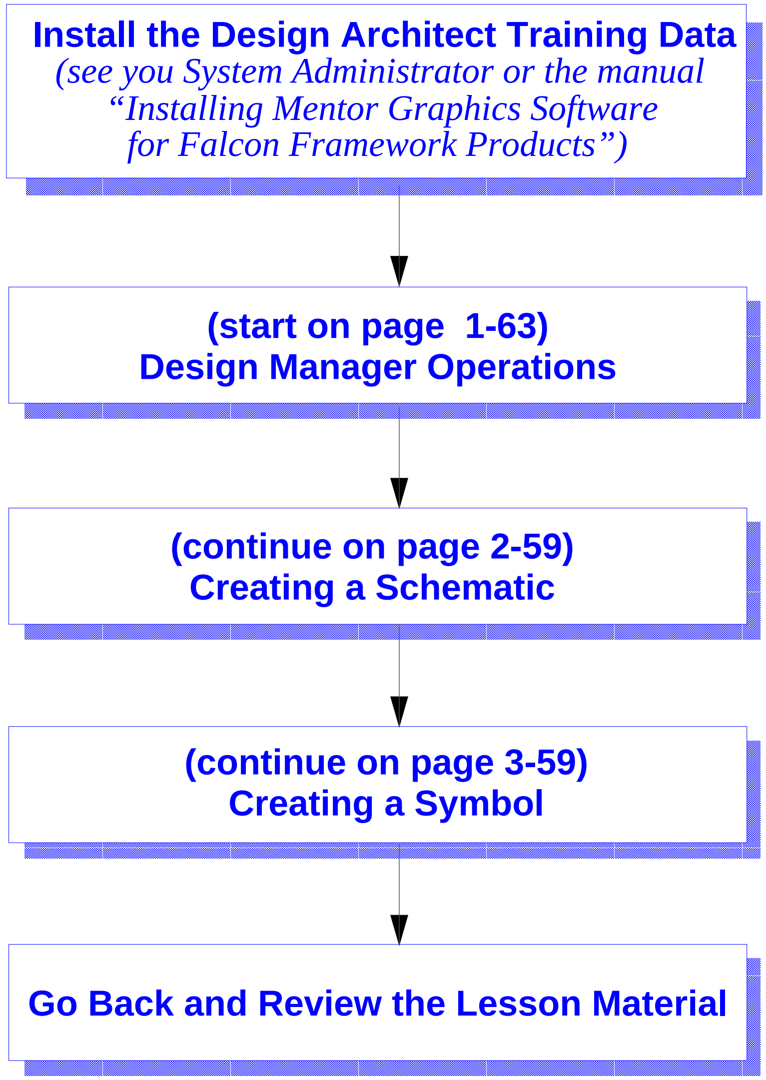
When your bookl is open to any concept, you can see all of this information at once.



The back portion of each module contains hands-on lab exercises that give you experience using QuickPath to analyze the timing of a design. Although it is helpful to read the Lesson material first, you can perform the lab exercises in each module without reading the lesson material.

For the Person in a Hurry

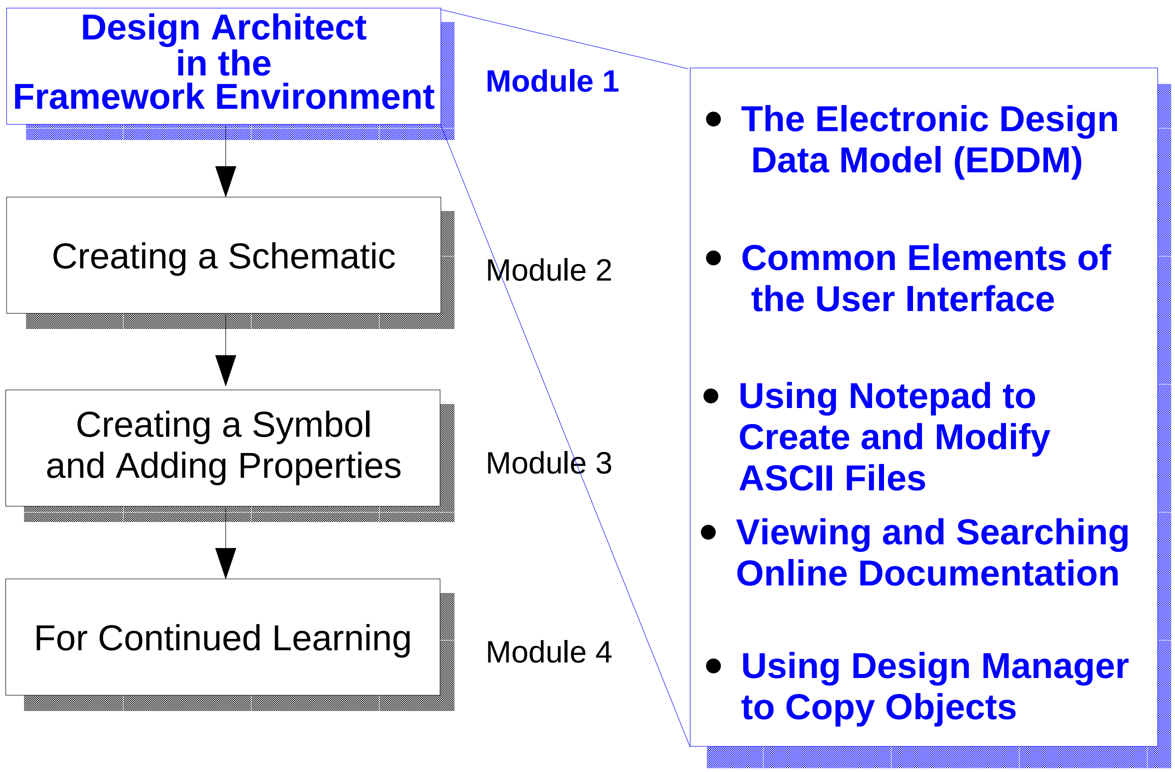
If you learn best by doing rather than reading, follow the path illustrated below:



Module 1
Design Architect
in the Framework Environment

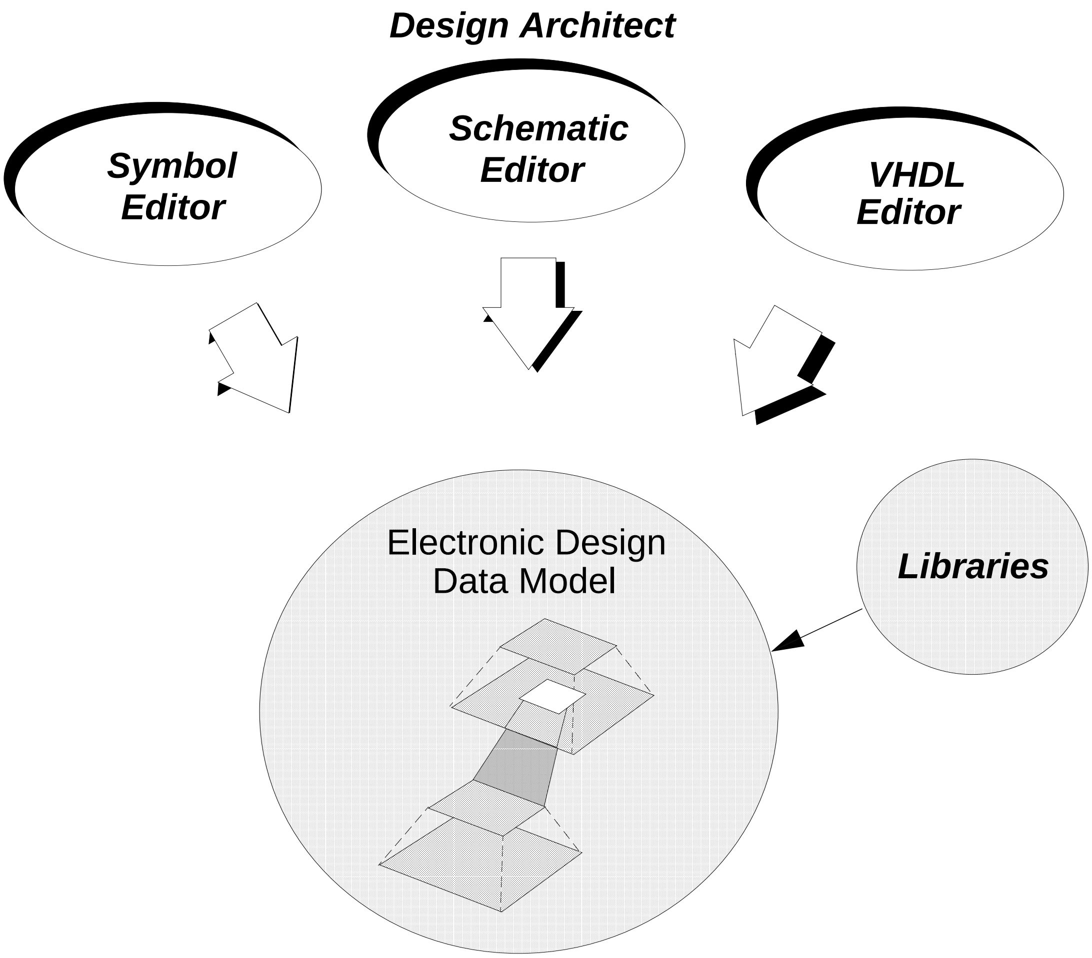
Lesson 1 The Electronic Design Data Model 1-3
Lesson 2 Common Elements of the User Interface 1-15
Lesson 3 Using Notepad to Create and Modify ASCII Files 1-37
Lesson 4 Viewing and Searching Online Documentation 1-41
Lesson 5 Using Design Manager to Copy Objects 1-49
Lab Exercises 1-63

Module 1 Overview



Lesson 1
The Electronic Design Data Model

Design Architect and the EDDM

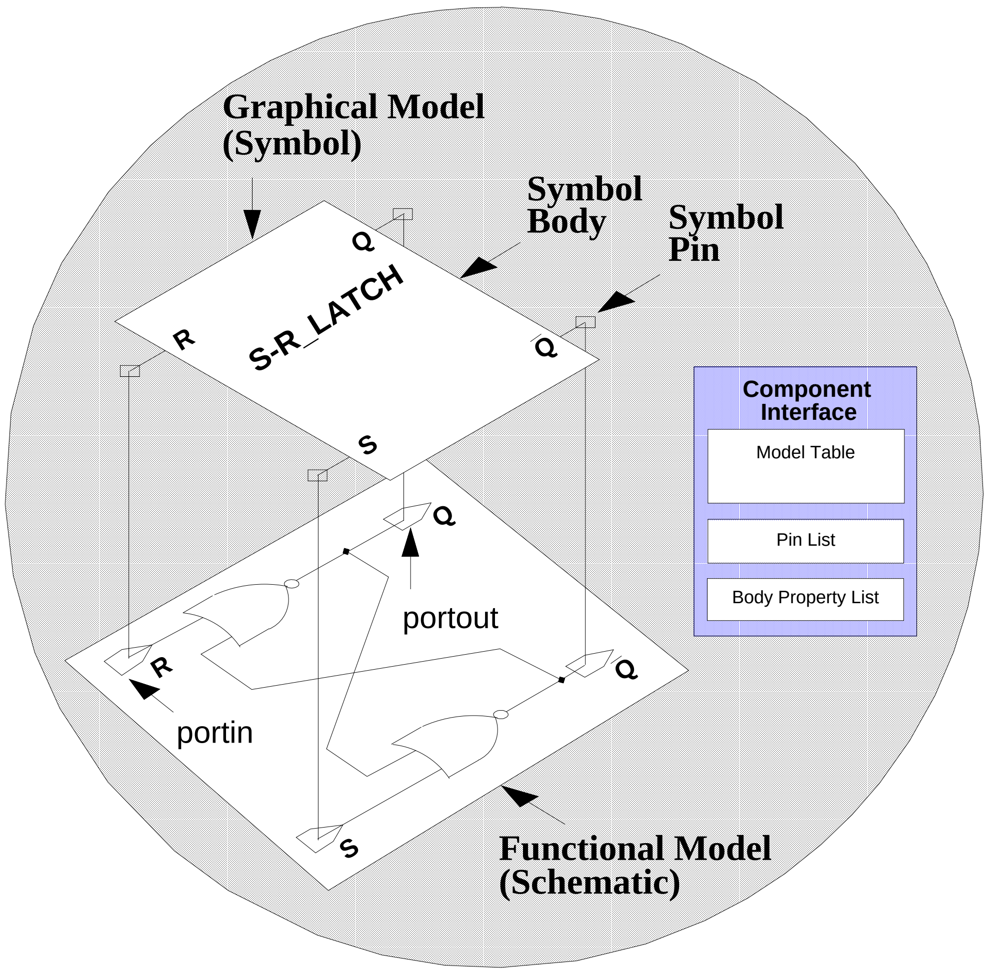


Design Architect and the EDDM

Design Architect - a Tool Box You can think of Design Architect as a tool box containing a collection of tools (graphic and text editors) that are used to create and modify the data that is modeling your design. The three primary tools in Design Architect are the Symbol Editor, the Schematic Editor, and the VHDL Editor. Many other optional tools can be added to Design Architect through the use of Personality Modules.

The Electronic Design Data Model (EDDM) When you use the editors in Design Architect to enter a design, a set of design files and directories is created called the Electronic Design Data Model (EDDM). Typically, the basic building blocks of your design will be pre-built components that reside in libraries located somewhere in your file system.

Component Structure (Conceptual View)

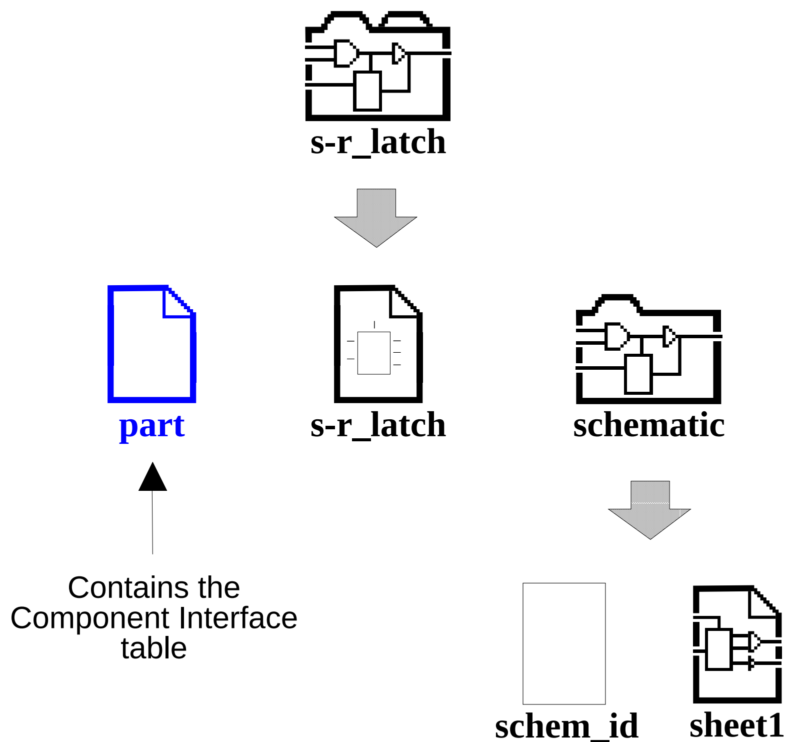


Simple Component Structure (Conceptual View)

The basic unit structure of the EDDM is called a component. In other systems, this unit structure might be called a cell. A component can be thought of as a sphere containing a collection of models that represent a “chunk” of electronics. The simple structure illustrated on the facing page shows a component that models an S-R latch. The component contains a graphical model called a symbol and a functional model in the form of a schematic. The control center of the component is called the “component interface” and may be thought of as the “nucleus” of the cell.

- **Graphical Model (symbol).** Composed of symbol body graphics, symbol pins, and properties. A component may have more than one symbol model associated with it.
- **Functional Model (schematic)**. Describes the functional behavior of the electronics being modeled. More than one functional model may be present. A non-schematic functional model is called a “primitive”. Each functional model must have input and output “ports” that match the input and output “pins” on the symbol.
- **Component Interface.** Acts as the control center for the component structure. This design object takes the form of a table that collects and retains information about the symbol pins, symbol body properties and each functional and timing model associated with the component. Placing a model in the model table of the Component Interface “registers” the model with the component.

Component Structure (Iconic View)



- **Component Icon - represents the directories and files in a component structure**
- **Design Object Icons - represent other objects within the component structure (symbols, schematics, sheets**
- **Part Object - a binary object that contains the Component Interface table**

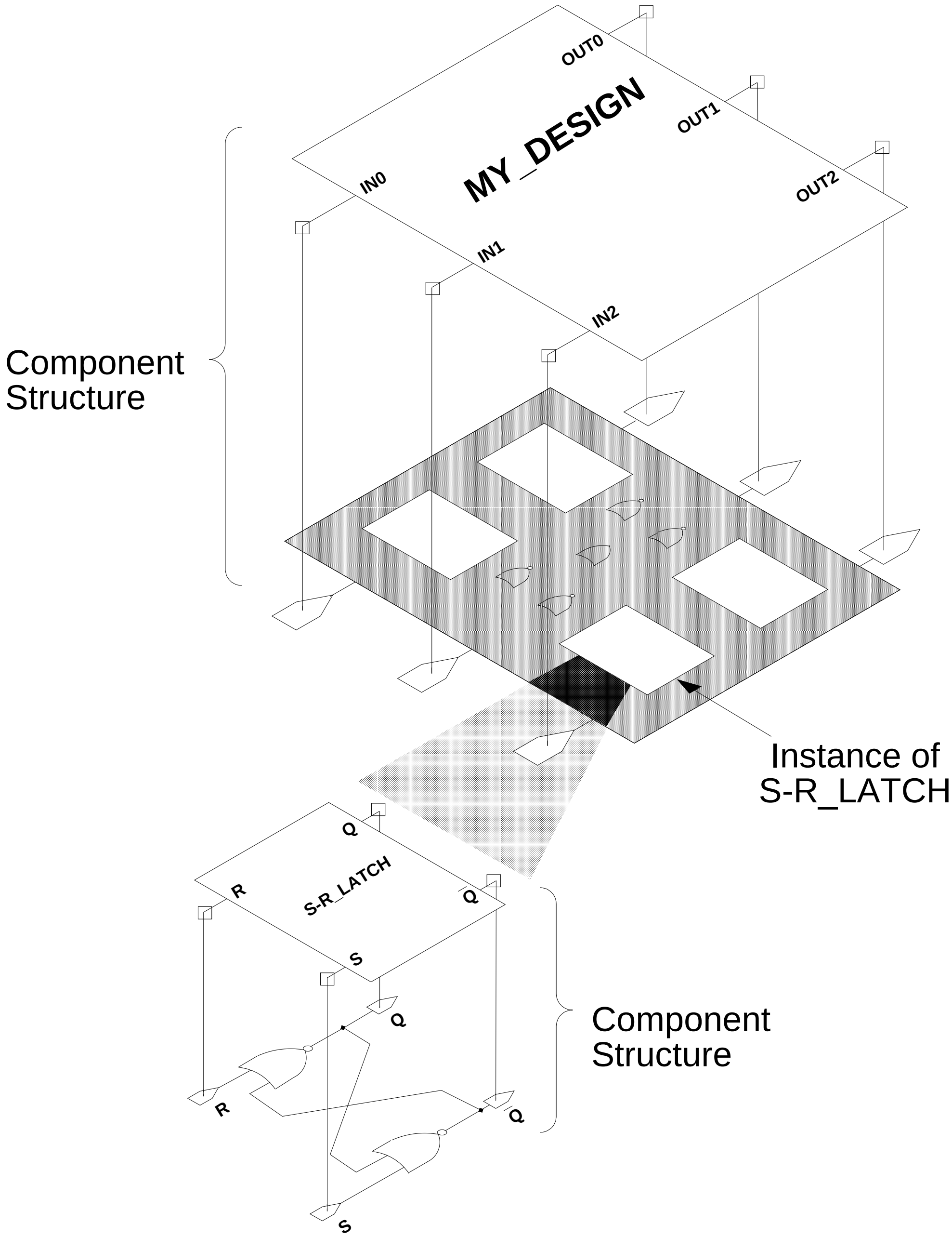
Component Structure (Iconic View)

Your design data is modeled by an object-oriented database and each “object” is represented by one or more directories or files in the Unix file system. To make things simpler, these objects are represented by icons when you view them through a window such as the Navigator.

If an object is really a Unix directory in the file system, it is called a “container.” A container can contain other objects. The icon at the top of the illustration on the facing page shows the icon for a component structure. Because a component is a container, it contains several other objects such as the part object, the symbol object, and the schematic object. The schematic object is also a container and can contain one or more “sheet” objects. The “schem_id” object is a special object that helps the system manage the assignment of system identifiers (handles) to various objects on the schematic sheets. (The subject of handles will be covered in a later module.)

The part object is a special object that contains the Component Interface table. Remember that this table acts like the control center of the component where information about the symbol pins, symbol body properties and each model is collect and retained.

Component Within a Component



Component within a Component

The picture on the facing page illustrates how a new design is created by placing components within a component. Two component structures are shown: one named *S-R_LATCH* and the other named *MY_DESIGN*. The component named *S-R_LATCH* contains a symbol model and a functional model and is typical of a component that might be found in a library.

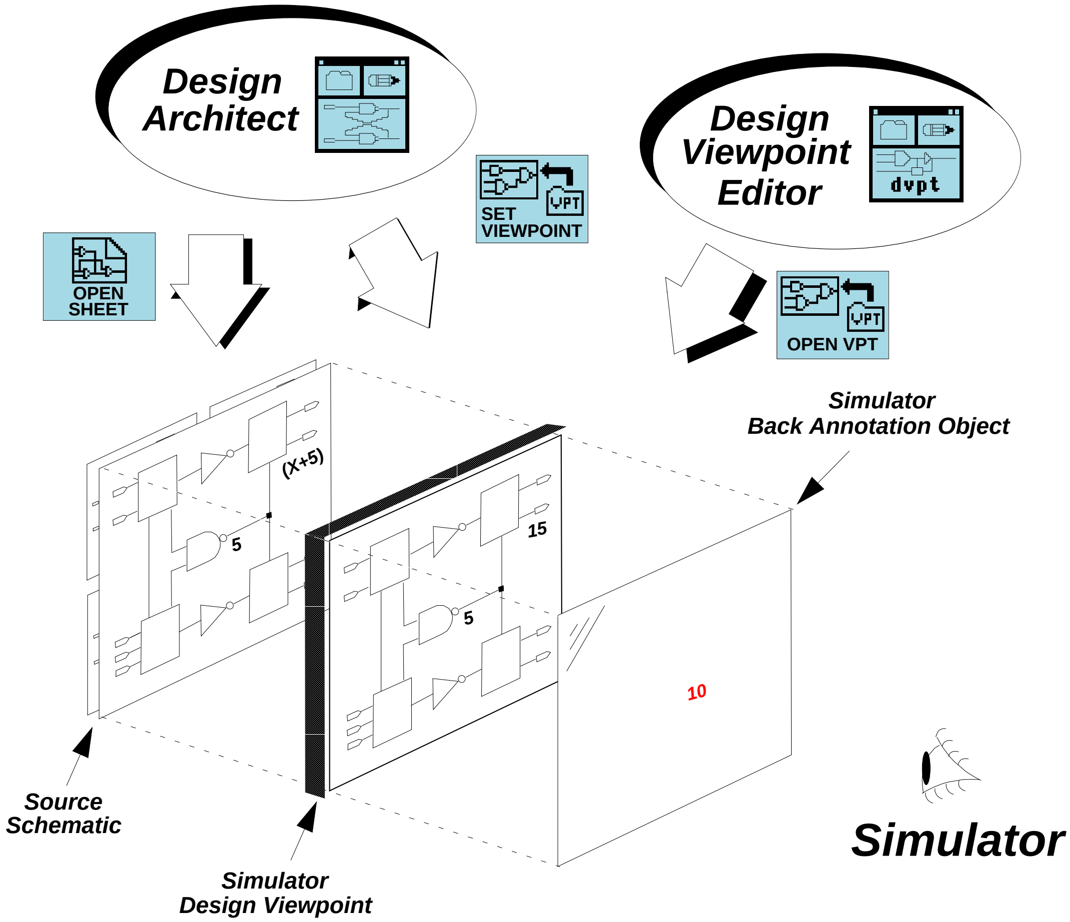
A new component structure is created when you open a new schematic sheet with Design Architect. The new component in the picture is called *MY_DESIGN*. In this case, the image of the *S-R_LATCH* component symbol is placed on the new schematic. Each image is called an “instance” of the *S-R_LATCH* and can be thought of as an active reflection of the *S-R_LATCH* symbol located in the library.

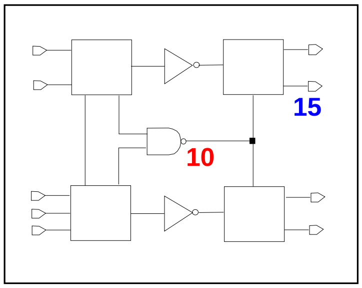
Each instance of the *S-R_LATCH* is considered different than the symbol body, because some of the characteristics of each instance can be changed once the instance is placed on the new sheet. Additionally, each instance can also be viewed as “linked” to the symbol body, which allows each instance to be collectively or individually updated after changes are made to the symbol body in the library.

Once you create a new schematic design, you can create a symbol to represent it. In this case, the symbol *MY_DESIGN* is created with the Design Architect symbol editor and registered with the *MY_DESIGN* component structure. Notice that the new symbol contains three input pins and three output pins to match the three input ports and three output ports on the schematic.

The new *MY_DESIGN* symbol can now be instantiated on a new higher-level schematic sheet that may represent the electronics of a more complex system.

Design Viewpoint (Conceptual View)





What the Simulator Sees

Design Viewpoint(Conceptual View)

Schematics are represented by files and directories in a software environment, so they can take on some of the characteristics of a software program. For example, a timing value can be represented by a numeric expression such as (X + 5) as shown in the figure on the left. This expression must be evaluated to a constant before a downstream tool like a simulator can operate on it. The object in the data model that allows a downstream tool to view the source schematic as fully evaluated data is called a *design viewpoint*.

You may conceptually think of a design viewpoint object as a picture frame through which the downstream tool views the schematic. In your mind's eye, think of the image of the source schematic as being reflected onto the back of the glass in the picture frame. Notice in the diagram that the simulator sees the fully evaluated data through the viewpoint (15 in this case) even though the expression on the source schematic (X + 5) doesn't change. The value of X can be defined elsewhere on the schematic or defined in the viewpoint itself.

Because the glass in the viewpoint protects the source schematic, you can't change the source schematic from the downstream tool. You can appear to change the schematic, however, by selecting a property in the simulator Schematic View Window and making a change. The change is recorded in a Back Annotation object, which is conceptually represented as a transparent sheet laid over the top of the glass in the viewpoint. In the figure, the timing value in front of the center **and** gate is changed from 5 to 10 nanoseconds. The simulator sees 10 ns, as shown in the lower figure, even though the source schematic is unchanged.

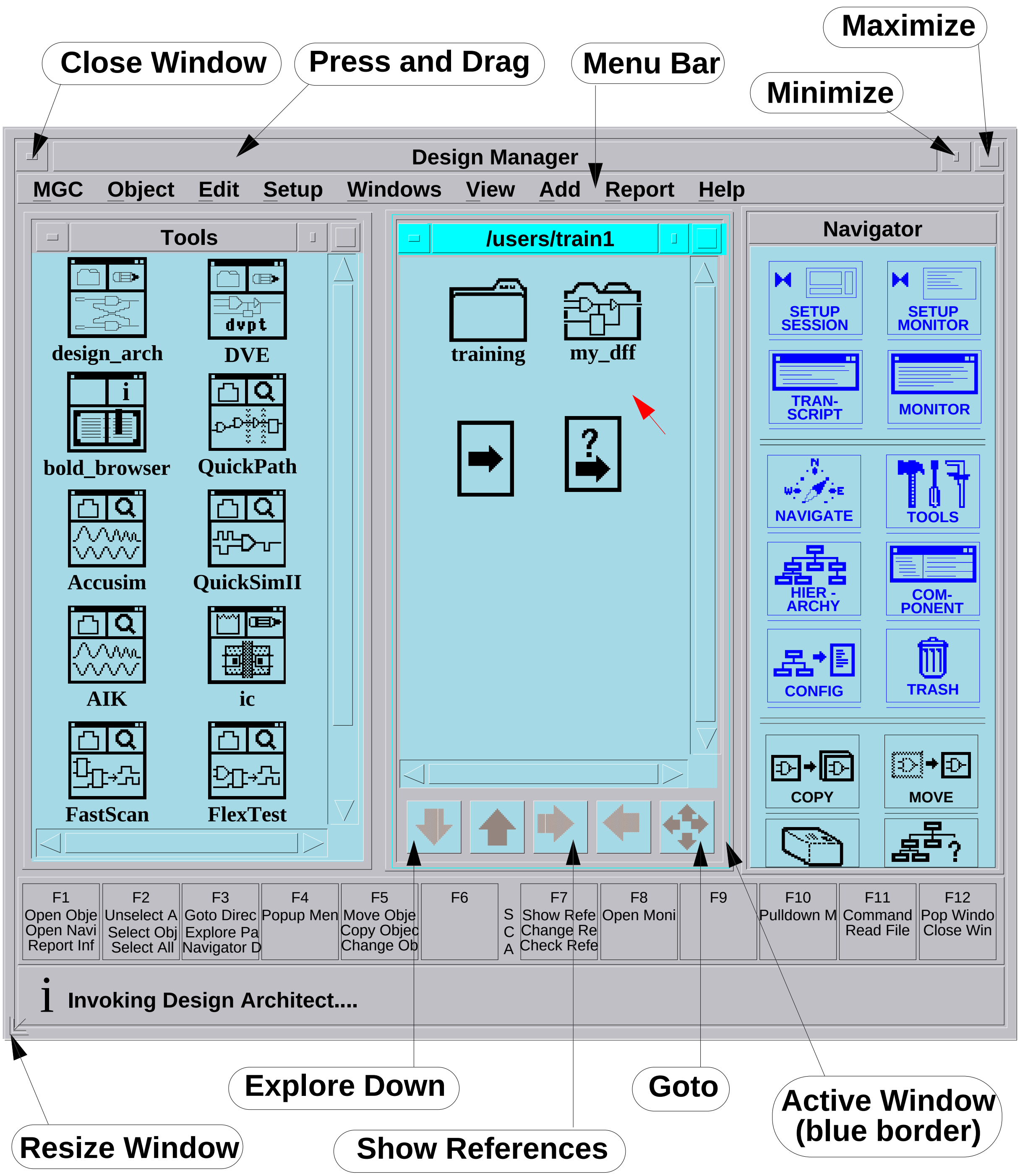
All downstream tools must view the source schematic through a viewpoint. Typically, if a schematic doesn't have a viewpoint, the downstream tool creates one automatically when the tool is invoked on the design.

Viewpoints can be created and modified with a tool called the Design Viewpoint Editor. Design Architect can also invoke on a design viewpoint (using the SET VIEWPOINT icon) as well as a source schematic (using the OPEN SHEET icon). When you invoke Design Architect on a design viewpoint, you may selectively merge back annotation information from the Back Annotation object onto the source schematic.

Lesson 2
Common Elements of the User Interface

Many applications in the Mentor Graphics environment share common elements. This lesson introduces you to these common elements.

Window Buttons and Navigator Controls



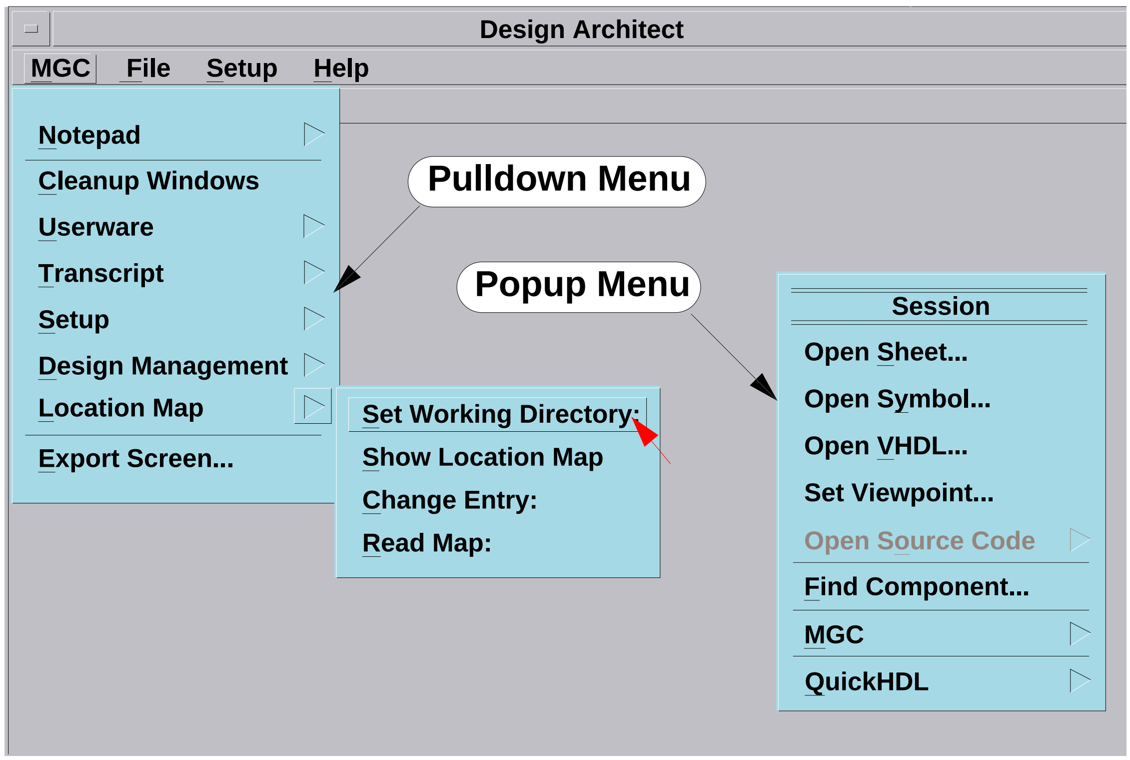
Window Buttons and Controls

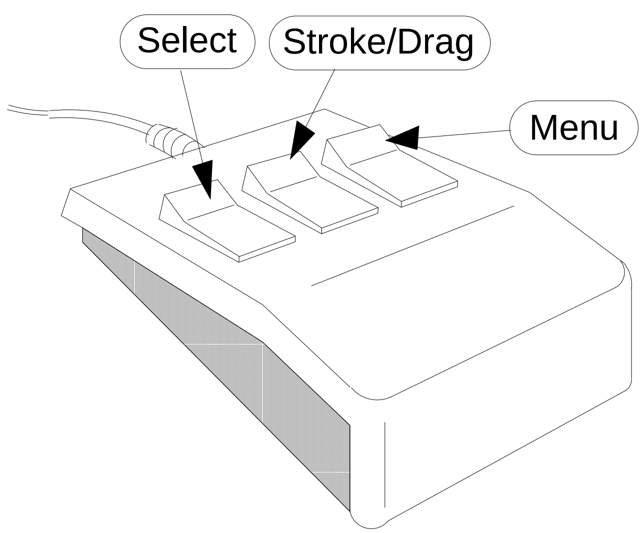
The illustration on the left show how the Design Manager appears when it is invoked from a shell. The title bar on the window has several important controls. When you click the Maximize button, the window expands to fill the whole screen. Click Maximize again and the window returns to its original size. When you click on the Minimize button, the window turns into an icon. When you double click on the Close Window button(on the far left), the window goes away. Finally, if you place the mouse pointer on the title bar and press the left mouse button, you can reposition the whole window to any place on the screen. When you move the pointer to a window corner, a Resize cursor appears, as shown in the lower-left corner, and you can press the left mouse button to resize the window.

You can navigate, or move around in, directories and files by using the Navigator window. The Navigator contains the following areas and controls:

- **Iconic Area.** Displays icons represented by icons.
- **Scroll bars.** Scroll iconic area vertically and horizontally.
- **Navigation buttons.** Navigate the file system. These buttons correspond to the following actions:
    - **Explore Down.** Navigates into the selected directory and displays the objects in that directory.
    - **Explore Parent.** Navigates up one directory or reference level.
    - **Show References.** Replaces the current display with the references of the selected design object.
    - **Explore Back to Parent.** Closes the report window of the references currently displayed.
    - **Go To.** Displays a dialog box into which you can enter a pathname and then moves directly to that destination. You may also type a Unix change directory command like “**cd /user/train1/training/da_n**” and the Navigator will move to that location.

Menus and Mouse Buttons





Menus/Mouse Buttons

Pulldown Menus Place the mouse pointer on the Menu Bar item and press the Left or Right mouse button.

Popup Menus Place the mouse pointer in the window area and press the Right mouse button.

Graphic Elements in Menus:

- **Menu items.** Represent a specific task or a category of tasks. The examples shown on the facing page are Notepad, Cleanup Windows, Userware, Transcript, Setup, Design Management, Location Map, and Print Screen menu items from the **MGC** pulldown menu.
- **Cascade arrow.** Indicates that a cascading menu is present.
- **Separator.** Divides a related group of menu items from the rest of the menu items.
- **Cascading menu.** Displays to the right of the parent menu item.
- **Prompt bar indicator.** A colon indicating that the menu item displays a prompt bar for additional command or function information.
- **Dialog box indicator.** An ellipsis indicating that the menu item displays a dialog box for additional information to the system.

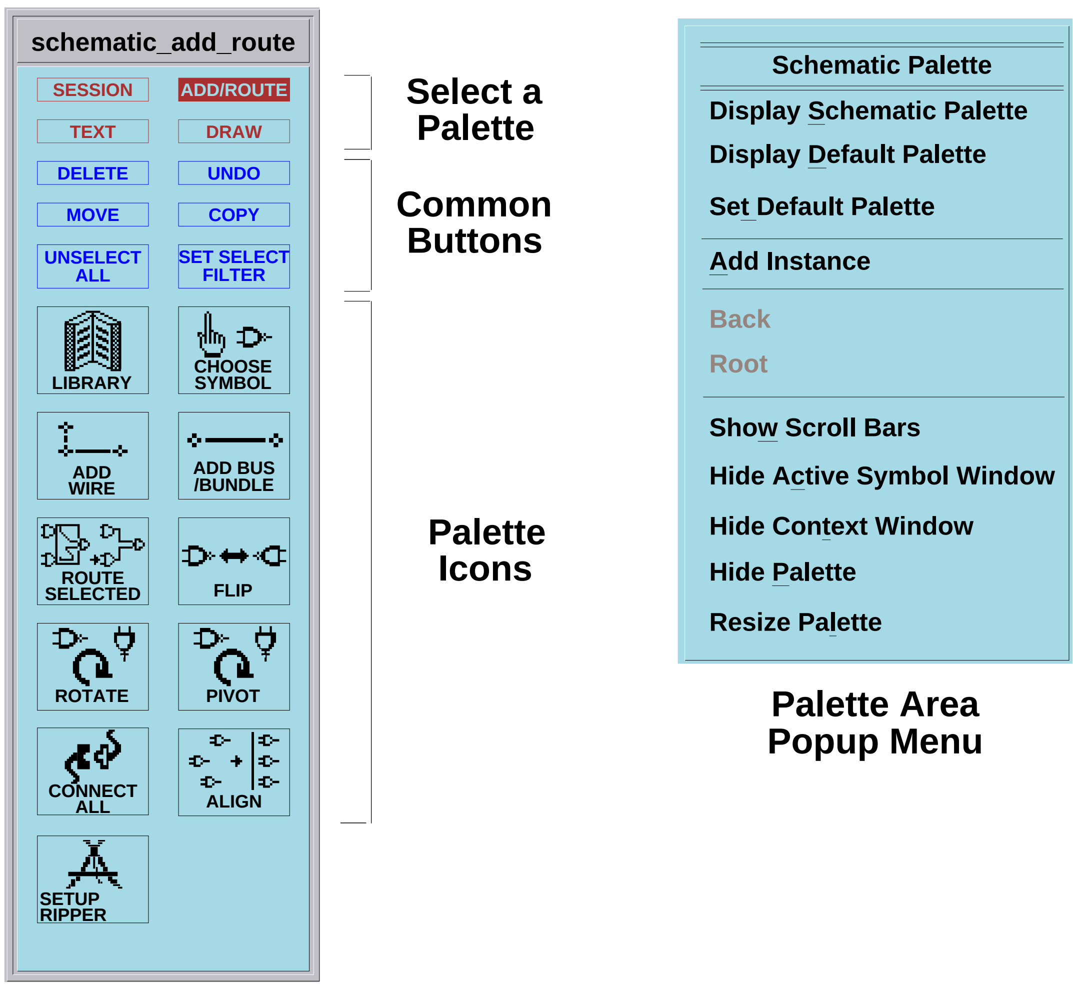
A cascading menu represents a group of tasks related to the parent menu item. On the facing page, the cascading Location Map menu item has four tasks:(1) Set the Working Directory, (2) Show the current location map, (3) Change an entry in the location map, and (4) Re-read the location map from disk.

To choose the menu item that you want, you slide the mouse pointer over the menu cascade arrow and on top of the desired menu item before releasing the mouse button.

Palettes

- **Default palette: displayed on invocation**
- **Name of palette depends on the application**

Three palette areas:



Palettes

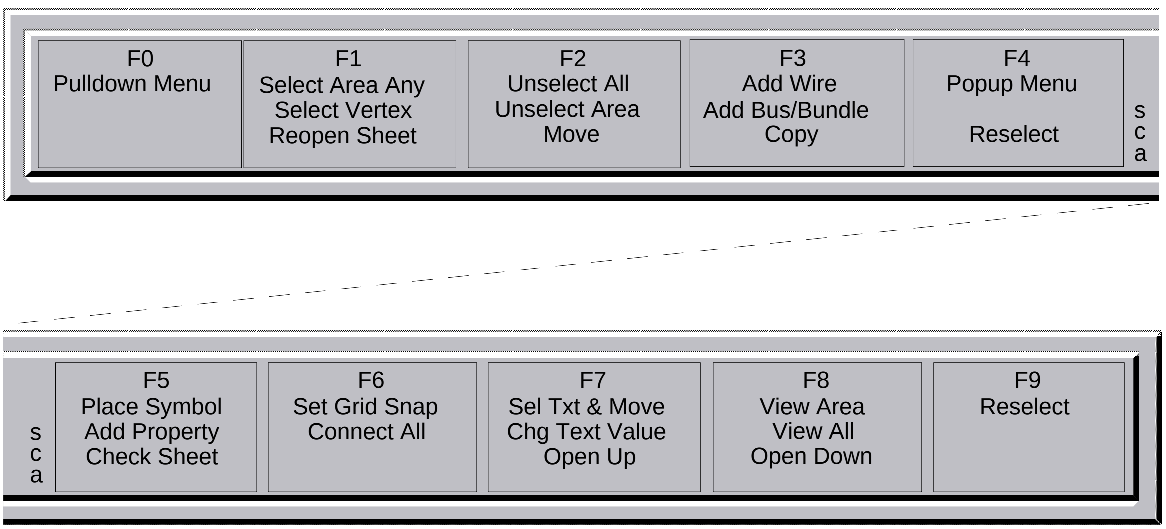
A palette is a convenient way to execute commands without having to traverse menu paths. You click the Select mouse button to execute a palette item. These items fall into three categories, depending on the area of the palette.

- **Palette Selection.** A palette is really a group of palettes all in one area. The top part of each application palette is identical and contains several choices, each representing a different palette. The button for the palette that is currently being view is highlighted in red, so if you click on it, nothing happens. When you click on any other button, you can access the corresponding palette. The new palette replaces the existing one.
- **Common Palette Buttons.** The middle area of each palette contains a common area of buttons that perform the same function on each palette. Many of these functions can also be found in the session popup menu.
- **Palette Icons.** These buttons form a subset of the total icons available and are grouped on a palette according to function. For example, the palette on the left contains icons that allow you to add instances to a schematic sheet and route nets between the instance pins.

Palette selections work much the same as menu choices; that is, you may be prompted for additional information in a dialog box or a prompt bar. In addition, palette buttons may be grayed to indicate that the choice cannot be made. Some buttons may require design objects to be selected before they become available, while other buttons require that a type of window be present.

Softkeys

- Indicate the operation of each function key
- They change from window to window
- They do not function as buttons (only visual)



Softkeys

Softkeys are not buttons like the ones you see in a palette. Instead, the softkey area is presented to give you a visual representation of the function keys. To execute a softkey action, you "look up" the action you want to perform in the softkey area, and then press the appropriate function key combination.

There are four key combinations for each function key.

- **Function key alone.** Several common user interface operations, such as "Pulldown menu" and "Popup menu," are reserved for this mode. Press the function key by itself.
- **Shifted function key.** The second row of operations, which are in line with "s" in the key, are executed by first pressing and holding the shift key while pressing the function key. The View All function (F8) is accessed this way.
- **Control + function key.** The third row of operations, which are in line with "c" in the key, are executed by pressing and holding the Control (Ctrl) key while pressing the function key. For example, this is how you access the Move command (F2).
- **Alt + function key.** The last row of operations, which are in line with "a" in the key, are executed holding the Alt key while pressing the function key.

To remove the softkey area, use the Hide Softkeys menu item in the Softkey menu, or the **Setup > Hide Softkeys** menu from the pulldown menu bar. To show the softkey area, you access the **Setup > Show Softkeys** menu item from the pulldown menu bar.

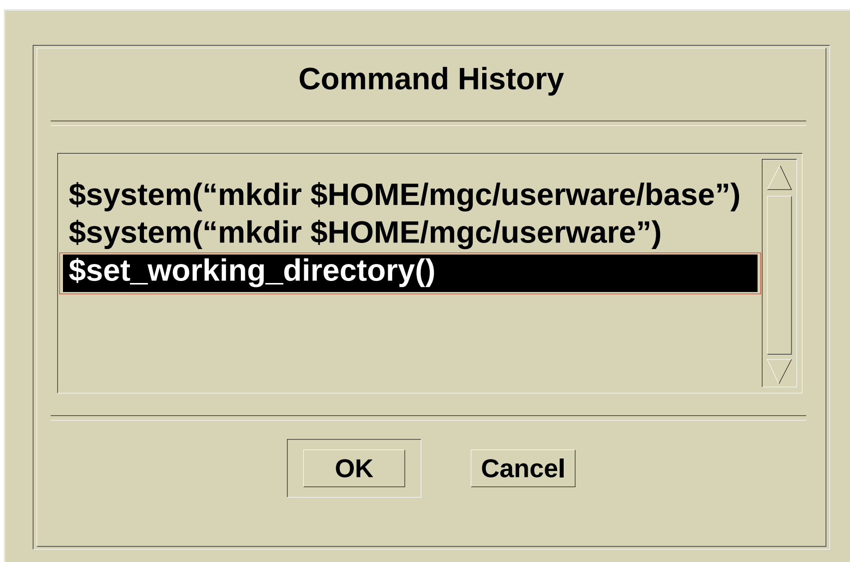
Command Window

Pops up when you start typing

CTRL P — Returns the previous entry to the window.

CTRL N — Moves forward in the History List and returns that entry.

CRTL H — Displays a complete History List. Select the entry you want.



ALT **Back Space** — Returns the previous entry to the window.

Pre-V8.4 Behavior

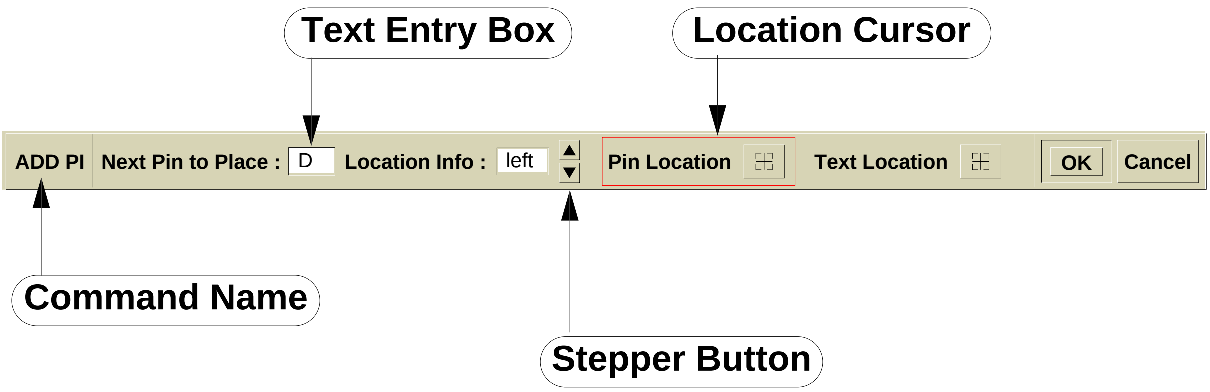
Command Window

When you start typing in a window, the command window automatically pops up with your entry. You may enter commands or functions in this fashion. The name of the active window is shown on the left side of the Command window and the command or function you type must be defined within a scope that is visible to this window.

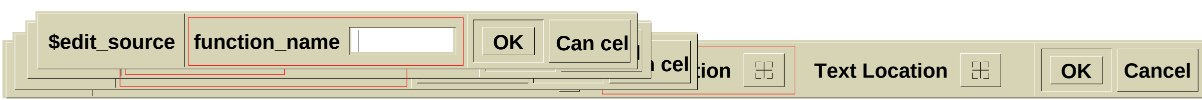
A history list is keep for this window. With the Command window displayed, type **Ctrl P** (for previous) to bring the last entry back into the window, type **Ctrl N** to move forward in the history list, and type **Ctrl H** to bring up a form with all the entries listed. Select one and click OK.

For software versions prior to V8.4, type **Alt BackSpace** to bring back the previous entry.

Prompt Bars



Mid-Command Freedom



Lets you execute an additional function before you complete the previous one

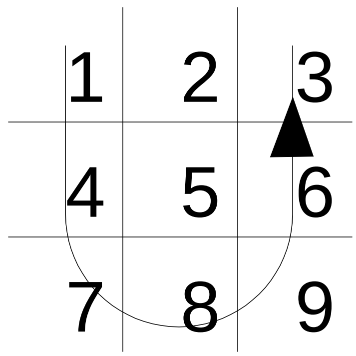
Prompt Bars

A *prompt bar* is a graphic aid that is displayed when you need to supply additional information to a command. When the prompt bar is displayed, you supply the additional information by typing in text or clicking a stepper button. Sometimes you must supply a coordinate location by clicking on a point in a window. When you have supplied all the necessary information, the prompt bar goes away. Sometimes, you execute a function like ADD WIRE in the Design Architect Schematic Editor and the prompt bar means that you are in the ADD WIRE mode. You can keep adding wires to the schematic until you exit the mode by clicking the Cancel button on the prompt bar.

Mid-command freedom is a feature that allows you to execute another command before you finish executing the current command. In the bottom illustration on the facing page, the stacked prompt bars indicate that three commands were suspended in mid-command while a fourth command $edit_source is being executed. Once the current command is finished executing, the system returns to the previous command. At any time, you can click the Cancel button on the prompt bar to clear it from the screen, but it must be the prompt bar on the top of the stack.

Strokes

- **You use the Middle mouse button**
- **A strokes pattern is converted to a number sequence from the grid**

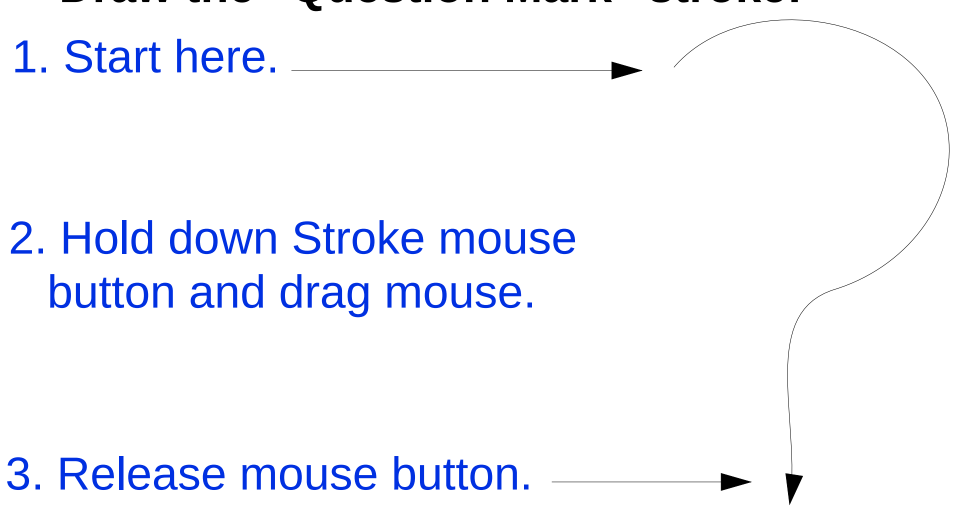


Unselect All

$stroke_1478963()

- **To get help on strokes:**

Draw the “Question Mark” stroke:



Strokes

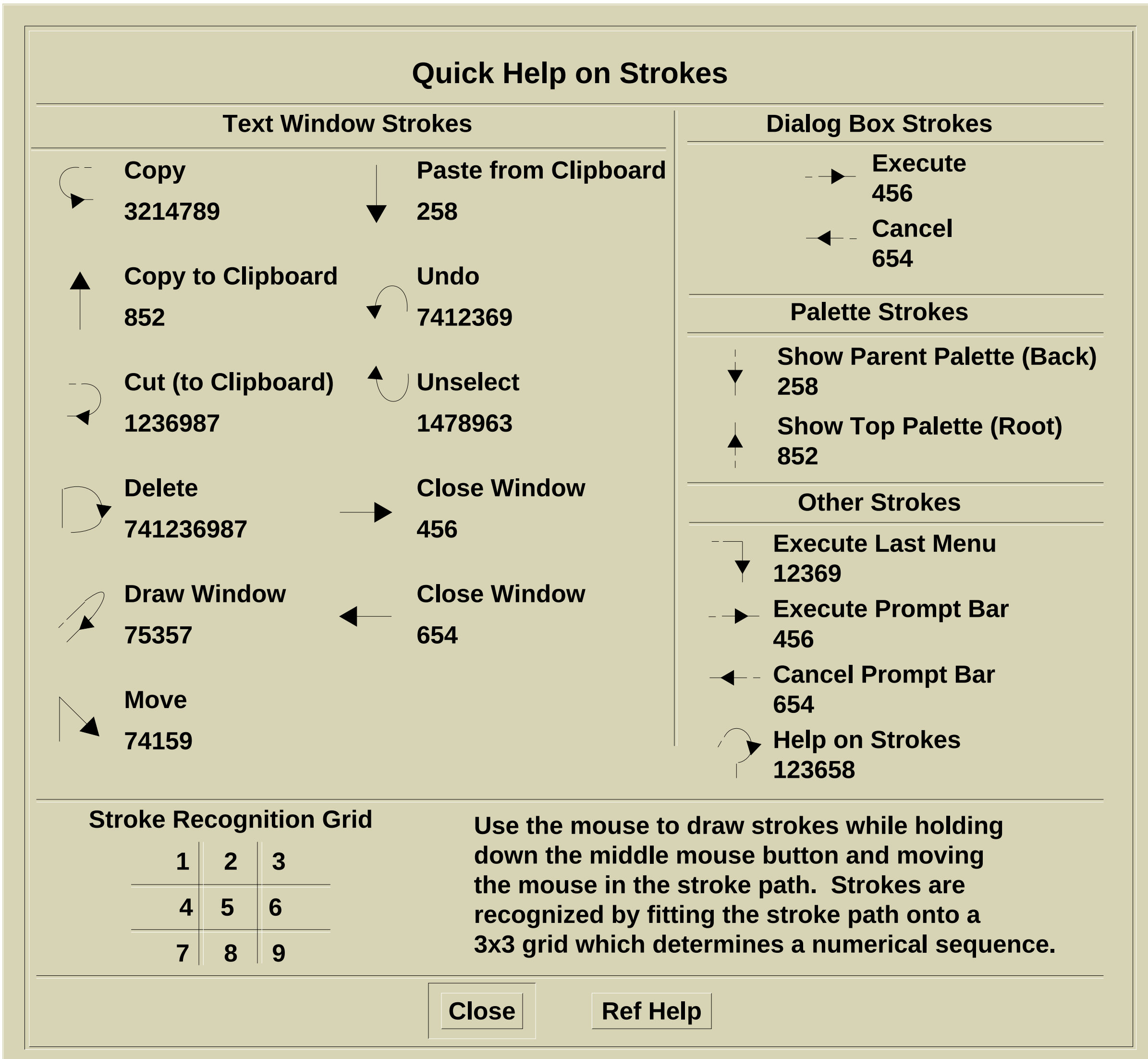
Strokes are another way of issuing commands within applications. You can activate stroke mode by pressing the Drag/Stroke mouse button (usually the middle button), and then you use the mouse to graphically “draw” the command.

For example, to unselect all objects in many applications, you press and hold the Drag/Stroke mouse button and then draw a “U” on the screen. The “U” will show graphically on the screen in the default stroke style and color, normally a narrow red line. When you have finished drawing the “U,” release the Drag/Stroke mouse button. The command executes immediately and all objects are unselected.

A stroke is defined by a sequence of grid coordinates, as shown in the figure on the left. This grid is called the *stroke recognition grid*. When you draw a stroke, the pattern is overlaid on the recognition grid, and a sequence of numbers is derived. If this sequence matches an existing defined sequence, the command for that sequence is executed. If the sequence is not defined, you get a “not defined” message.

In the example on the left, if you draw the “U” stroke, the pattern is interpreted as the number sequence 1478963. This is mapped to the function $stroke_1478963() which is defined as Unselect All. If you draw a “?” stroke, a quick help chart on the available strokes appears as shown on the following page.

Quick Help on Strokes

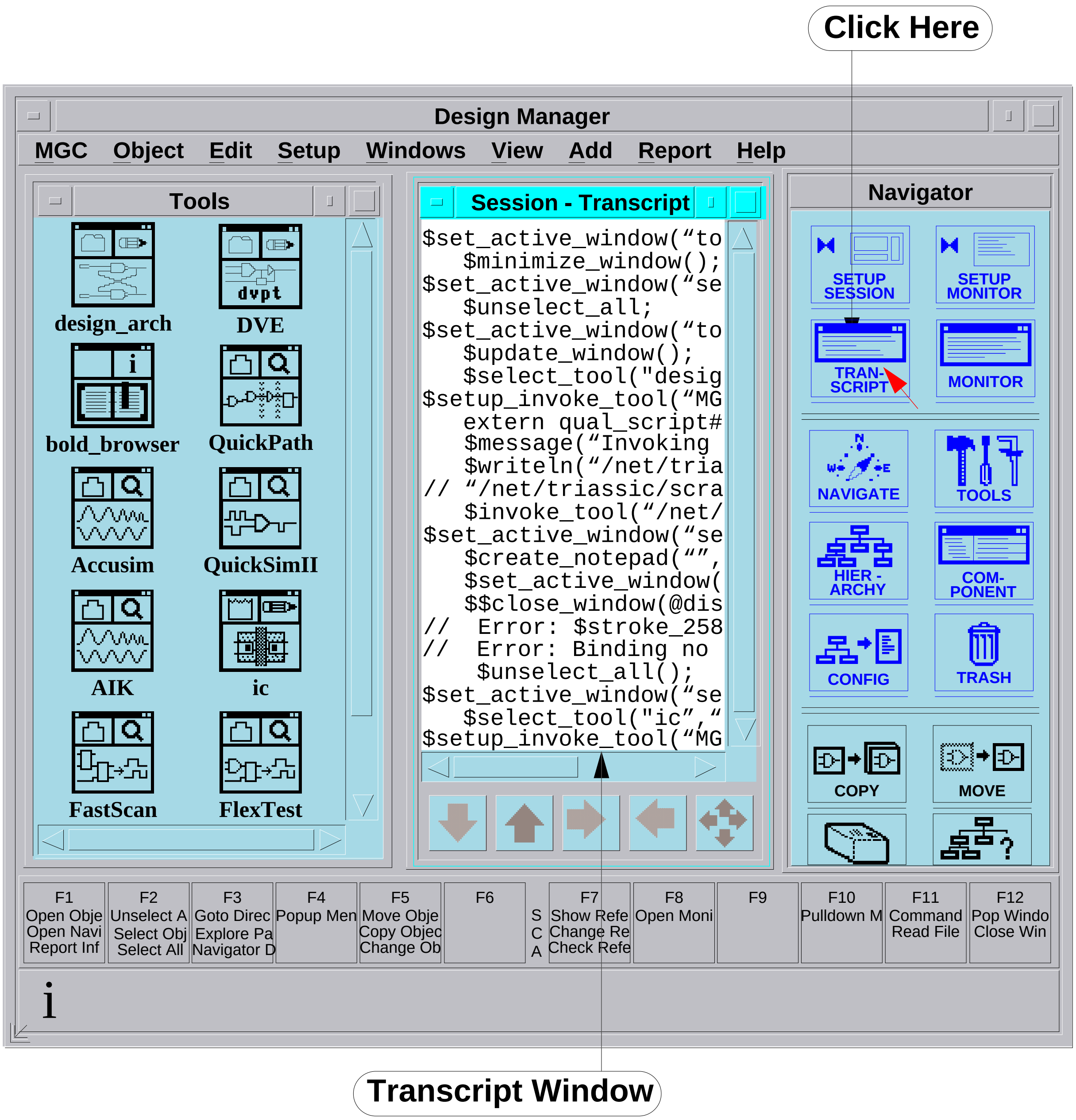


Quick Help on Strokes

If you draw a question mark stroke “?” in a Design Manager window, the Quick Help on Strokes chart appears as shown on the facing page. This chart defines the strokes that are available to you in that active window. Strokes are one of the most productive methods for executing commands, because all you have to do is wiggle the mouse in small patterns, instead of moving the pointer half way across the screen to click a palette icon or reach a pulldown menu..

It is often helpful to make a photocopy of this form, cut it up into strips and tape the strips on the edges of your display until you learn the strokes. After you use the strokes over time, you will remember them and they will come to you naturally, almost without thinking. Many of the strokes that you will learn from this chart will carry over to other applications, so they are well with the effort to learn.

Transcript Window

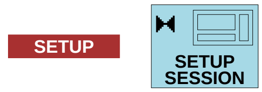


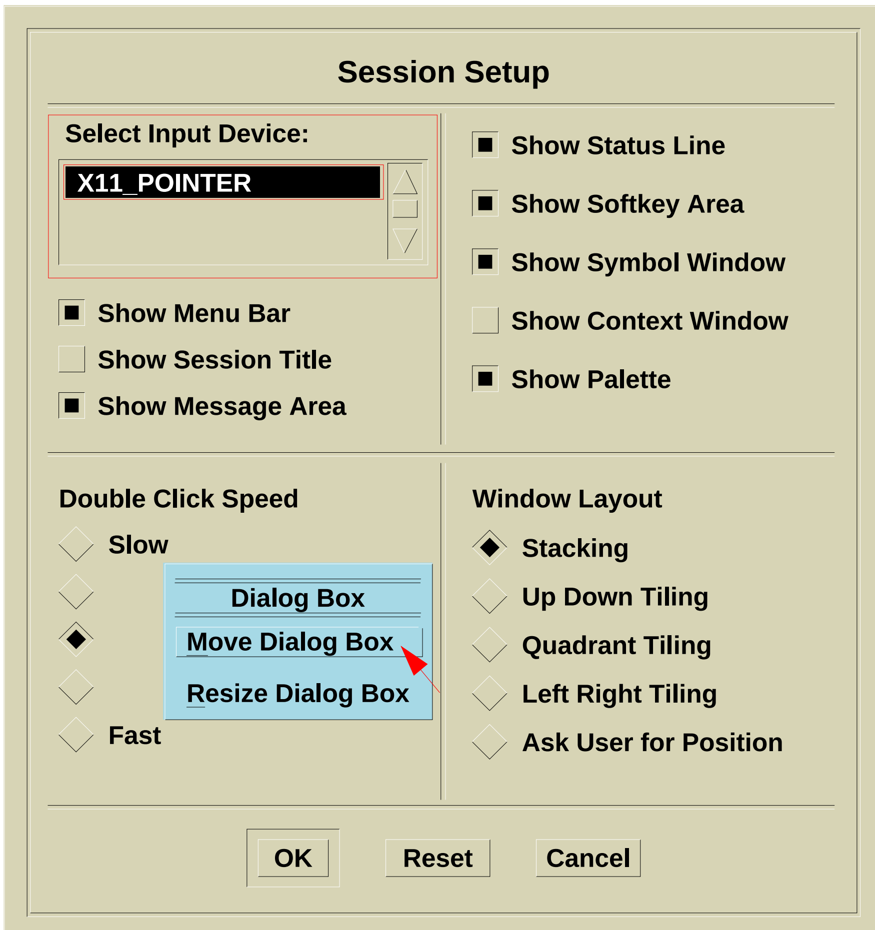
Transcript Window

A transcript is a record of the events that occur during a session. Although you can trigger events in a variety of ways such as clicking an icon or choosing a menu item, the actions taken are taken by executing a series of AMPLE functions. The transcript is a record of the AMPLE functions. Other information, such as a warning or error messages, are also recorded in the transcript.

It is possible to select text from a Transcript, then copy and paste the text to an ASCII file using the Notepad Editor. The ASCII text can then be saved as a macro file for later execution. This is a handy feature for creating application startup files and custom userware files.

Session Setup





Session Setup

When you click on a Session Setup icon or execute the pulldown menu **MGC > Setup > Session**, the form to the left appears. You can hide any number of out lying window areas, such as the palette and softkey area in order to gain more work space in the session area. In addition, you can set the speed of the mouse click and specify how you want the working windows in the session area to be displayed.

If a dialog box appears in an area that blocks your view of information, you can easily move the dialog box by taking the following steps:

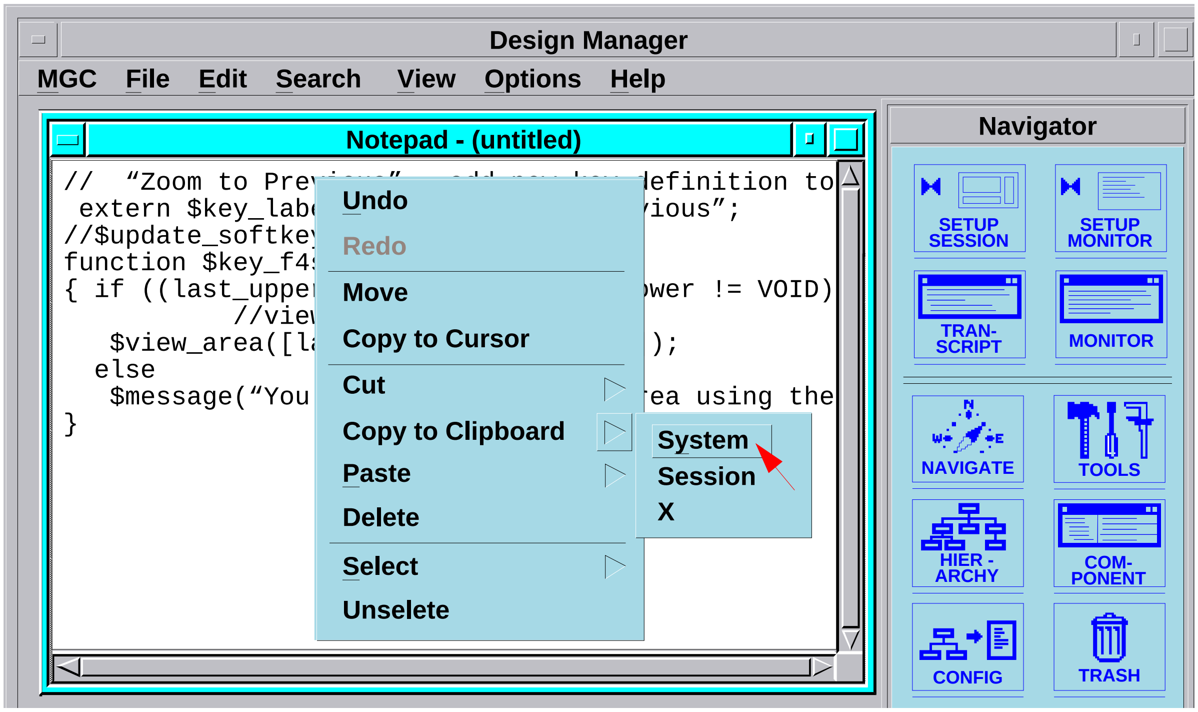
1. From any where in the Session area, press the right mouse button and click on **Move Dialog Box** as shown in the illustration on the left page:

2. Move the "shadow box" to a new location, then click the select mouse button.

Lesson 3
Using Notepad to Create and Modify ASCII Files

The Mentor Graphics environment provides a simple text editor that you can use to create and modify ASCII text files. In this course, you will use this editor to create and modify files that will customize the user interface.

Editing Files with Notepad



The following editing actions can be done:

- Move, duplicate, and delete selected text
- Copy selected text into a system clipboard
- Insert text from a system clipboard
- Undo or redo editing actions

Editing Files with Notepad

The Notepad editor enables you to accomplish several types of actions on text contained within a Notepad window. You can do any of the following from the Notepad popup menu, which is illustrated in the figure on the facing page.

- **Undo or redo editing actions.** If you want to reverse the effects of an action, you can use the **Undo** popup menu item to return the file to the way it was before the last action. If you decide you want that last action after all, you can use the **Redo** popup menu item.

- **Move, duplicate, and remove selected text.** You can use the **Move** or **Copy to Cursor** popup menu items to move text to a different location or to copy it to the cursor location. You can delete selected text into either a System paste buffer or into a Notepad Session paste buffer using the **Cut** popup menu items; the System buffer is the default.

- **Copy selected text into a clipboard.** If you have text that you want to transfer to another location, you can copy it into either the Session clipboard, the System clipboard, or the X window cut buffer using the **Copy to Clipboard** popup menu item. A clipboard essentially acts as a paste buffer within the Notepad session or between applications you have active on your workstation; the System clipboard is the default.

- **Insert text from a clipboard.** If you have text copied to a Session clipboard or a System clipboard, you can copy the text into the file you are editing using the **Paste** popup menu item; the System clipboard is the default.

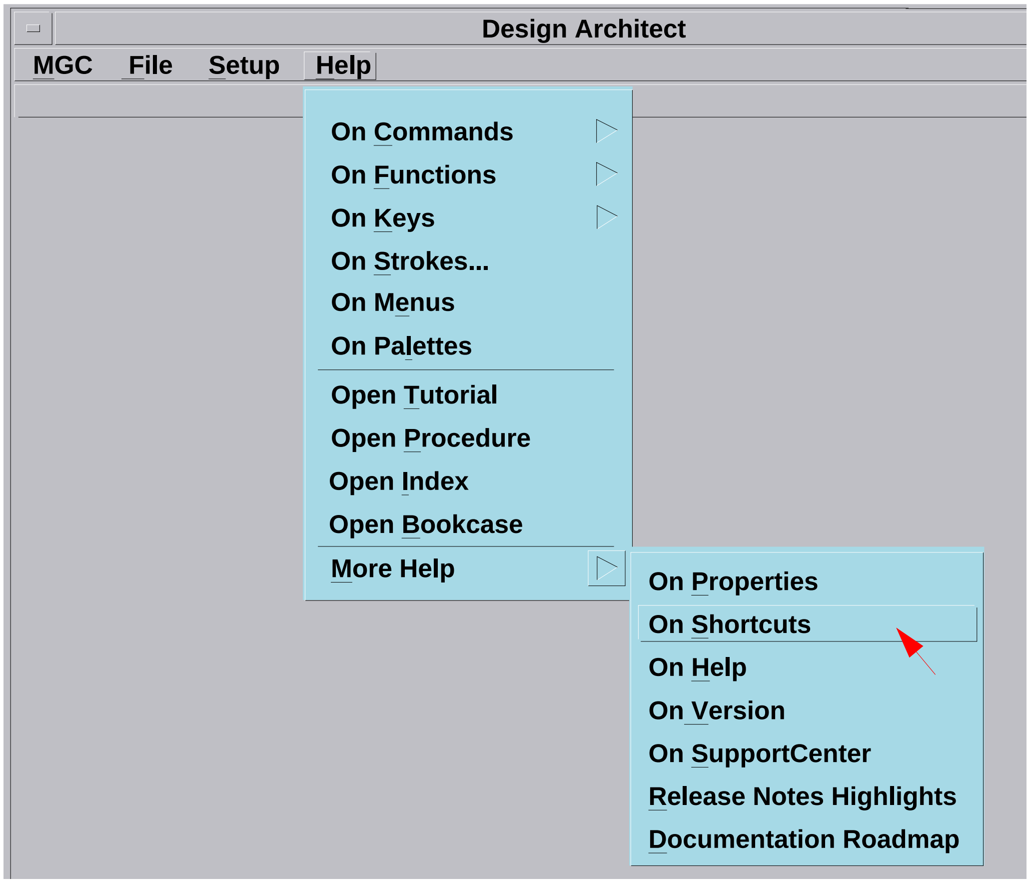
- **Select and unselect text.** You can use the **Select** popup menu item to select text on which you want to do other editing actions. You can also select text by pressing the Select mouse key, dragging the cursor to the end of the required text, and releasing the Select mouse button. If text is highlighted, you can remove the highlighting using the **Unselect** popup menu item.

Lesson 4
Viewing and Searching Online Documentation

A large resource of online documentation can be accessed through the BOLD Browser. This lesson shows you how to view the vast array of online documentation and perform a full text search on any given topic.

Online Help

- **Help menu options:**

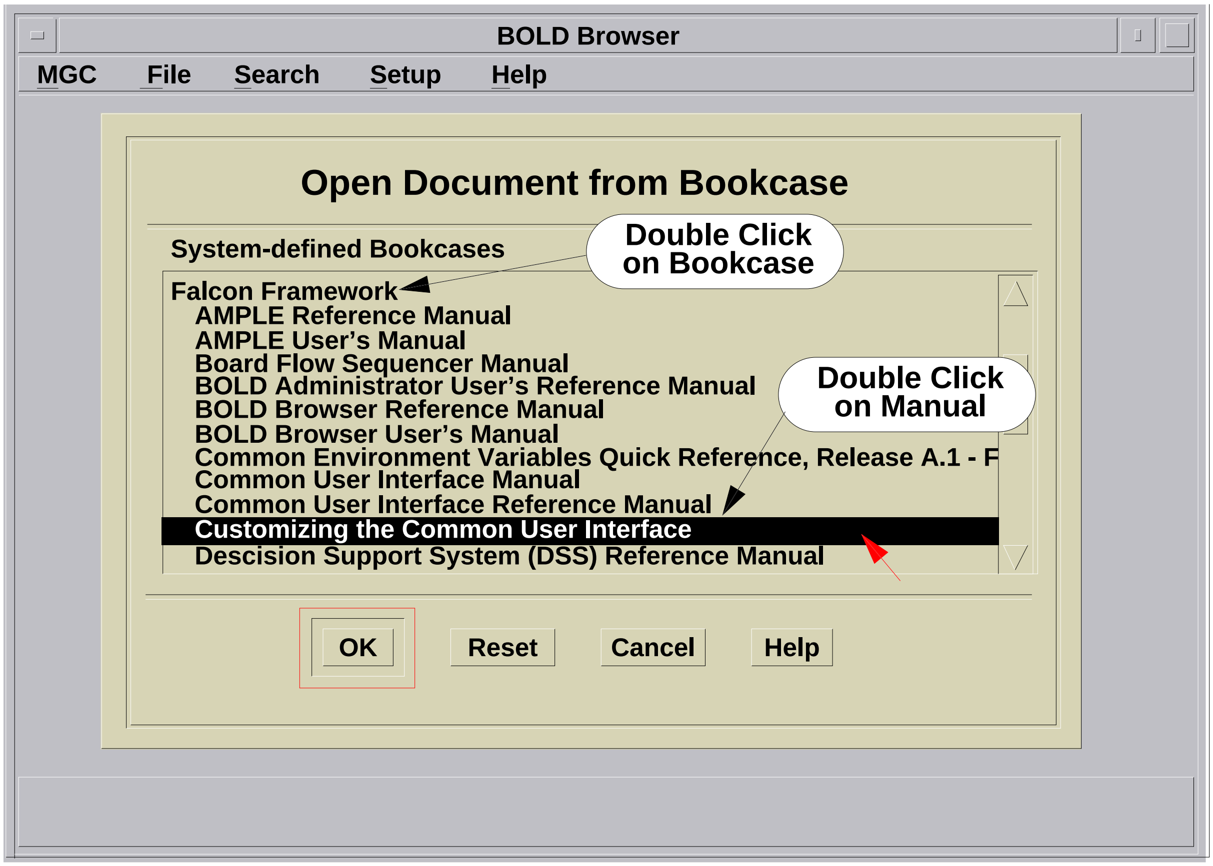


Online Help

You can access Design Architect online help through the Help menu in the menu bar. The following list describe what you can access from online help:

- **On Commands**. Provides Quick Help or Reference Help of Design Architect commands or a summary table of commands and functions. For a list of commands in the active window type ***** then **Ctrl-SHIFT ?**
- **On Functions**. Provides Quick Help or Reference Help of Design Architect functions or a summary table of commands and functions.
- **On Keys**. Provides help on predefined keys or a table of logical function key name mappings.
- **On Menus**. Provides help on the types of menus and the contents of each Design Architect menu.
- **On Palettes**. Provides a description of the Design Architect palette menus.
- **On Strokes**.   Provides a dialog box on strokes for the active window.
- **Open Tutorial**. Brings up a BOLD Browser window that displays the *Getting Started with Design Architect Training Workbook*.
- **Open Procedure**. Brings up a BOLD Browser window that displays the Operating Procedures section of the *Design Architect User's Manual*.
- **Open Index**. Brings up a BOLD Browser window that contains the index of the *Design Architect User's Manual*.
- **Open Bookcase**. Brings up a BOLD Browser window that contains the list of documents that compose the Design Creation documentation set.
- **More Help**. Information on properties, short cuts, customer support, release notes, documentation roadmap, on help, and current version.

Opening Online Documents



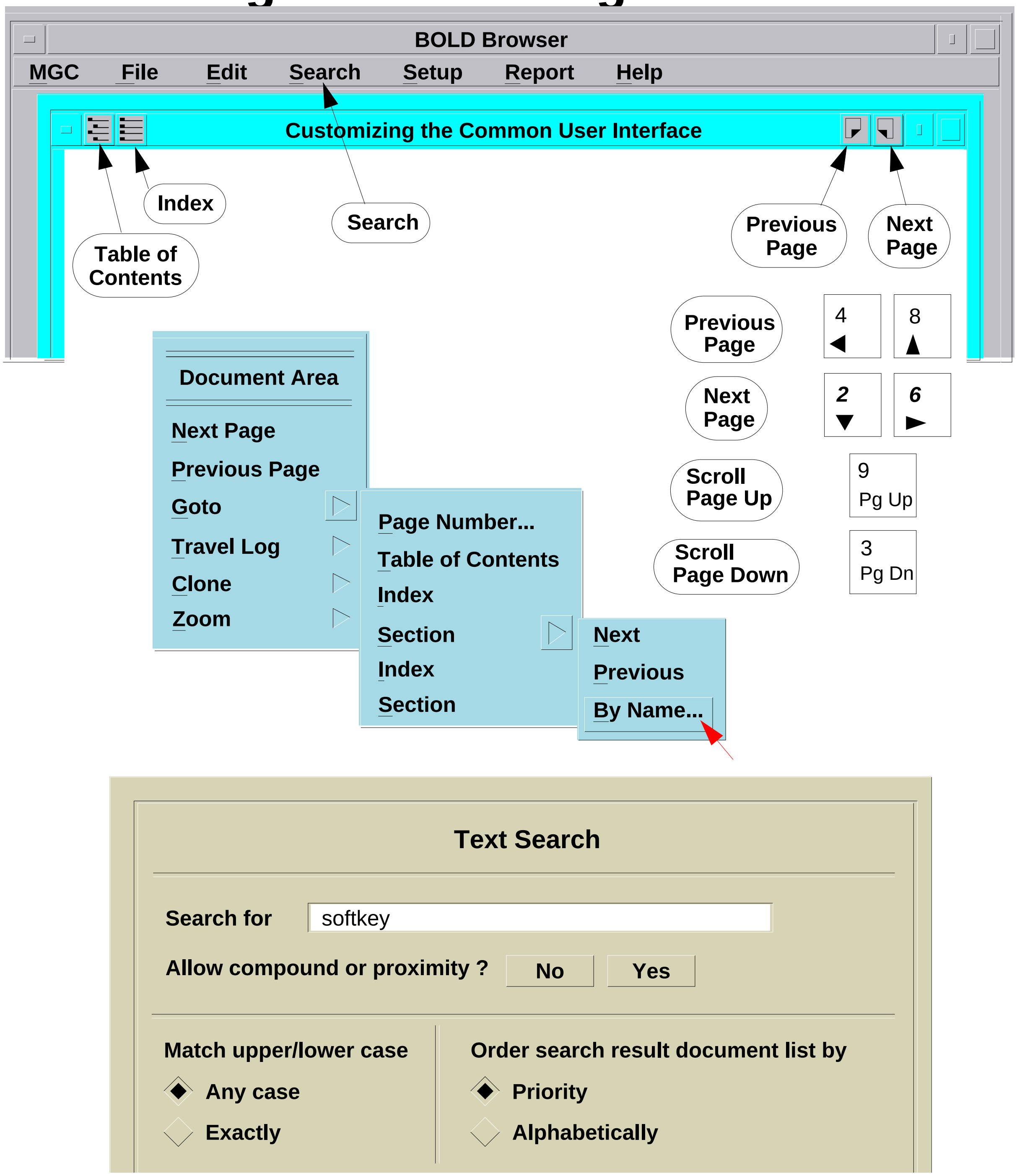
- **Double click on a bookcase to list the manuals within**
- **Select the manual and execute the form to open the document**

Opening Online Documents

Online documents are grouped into bookcases. When you choose **File > Open > Bookcase...**, the form illustrated on the left appears. When you want to open a bookcase to view the documents, just double click on the bookcase name. To open a document, double click on the document name.

All online training documents are contained in a bookcase named **Training.**

Searching and Traveling in Documents



Searching and Traveling in Documents

You can travel around in INFORM documents to continue with your current subject matter or to find additional information by one of the following methods:

- **Icons in the title area.** Clicking on these icons enables you to travel to the table of contents, index, a previous page, or the next page.
- **Popup menu or function keys.** Enables you to travel to the table of contents, index, a previous page, the next page, a particular page number or section, or to pages and documents that you previously viewed.
- **Hypertext links.** Enables you to travel to hyperlinked pages in the same document or in different documents. Hyperlinks are color highlighted on color displays and outlined on black-and-white displays. You click on the highlighted reference to move to that location.

The BOLD Browser incorporates a full text search index, which is a list of words and their locations in the online library. When you initiate a search for a word or word phrase, the BOLD Browser searches this index, rather than scanning through the text of all the documents at search time. This type of search is significantly faster than textual scans.

You can specify that BOLD search for specific needs, such as the following:

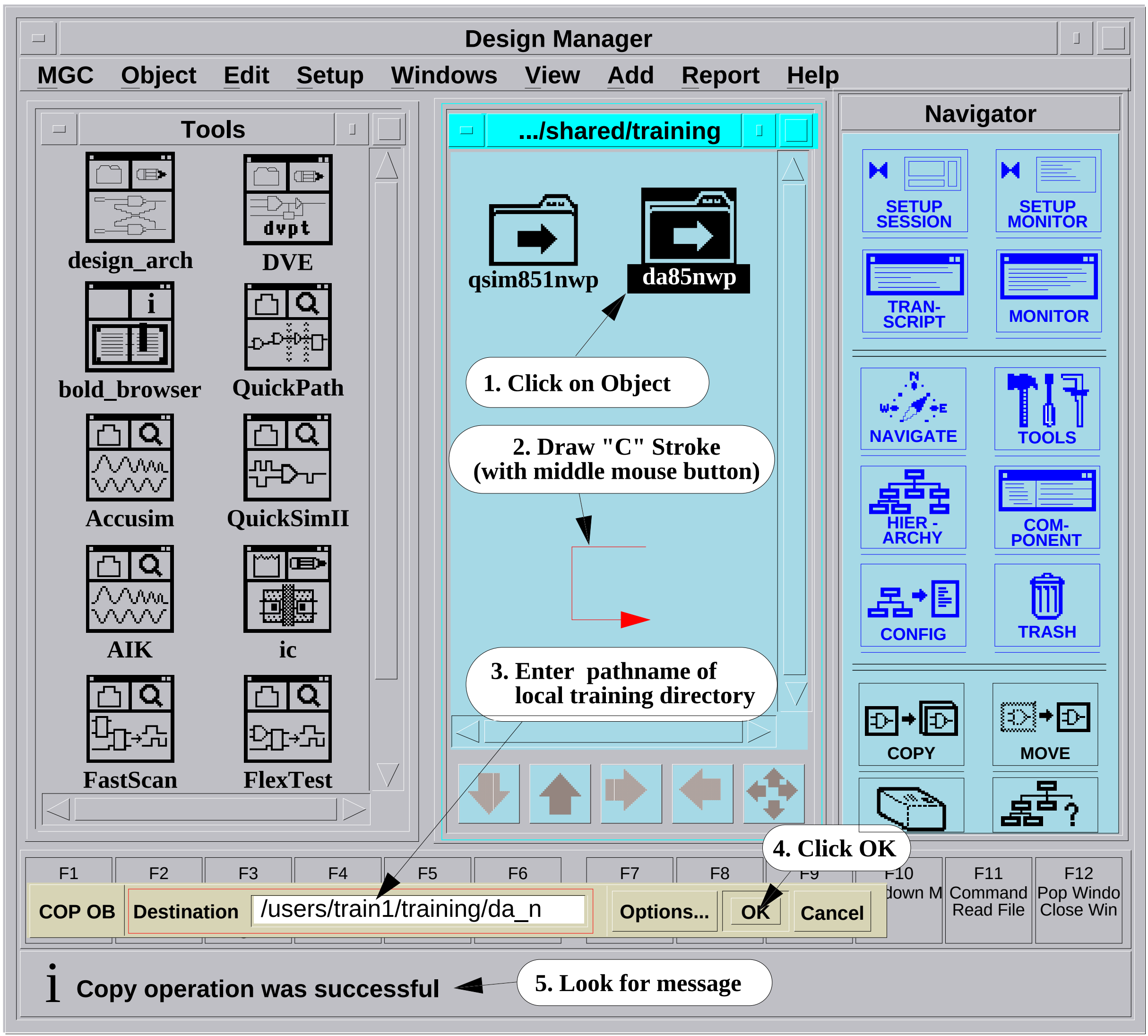
- Whether the word is part of a compound word.
- Determine letter case.
- Find variant forms of hyphenated words and plural forms.
- Restrict the search through certain document or a bookcase.

When the BOLD Browser locates the occurrences of the word(s) you requested, it displays a “Search Results” window where you can see a list of documents containing the term. You can click on the document titles to disclose lists of pages where the term occurs.

Lesson 5
Using Design Manager to Copy Objects

Changing the location of your design data in the file system should always be done with Design Manager. This lesson explains why you need to use Design Manager and how to successfully copy objects.

Copying Objects in the Navigator Window



Copy Objects in the Navigator Window

The illustration to the left shows the steps required to copy an object to another location in the directory structure. An alternative to using the “C” stroke is to press the right mouse button and execute **Edit > Copy:** from the popup menu.

Soft Prefixes and Location Maps

```
MGC_LOCATION_MAP_2 force

$CUSTOM_LIB -t LIBRARY
/usr2/dburnette/my_custom_parts

$PROJECT
/usr2/dburnette/project

INCLUDE /usr1/team_project/mgc_location_map
```

```
MGC_LOCATION_MAP_1

$MGC_GENLIB
/usr1/mgc_libs/gen_lib

#$MGC_LSLIB
#/usr1/mgc_libs/ls_lib

$PROJECT
/usr1/team_project
```

```
$MGC_HOME/etc/shared/mgc_location_map

$MGC_HOME/etc/mgc_location_map

./mgc_location_map

MGC_LOCATION_MAP=/users/dburnette/my_map
```

(environmental variable)

Search Path

Search Starts Here

Soft Prefixes and Location Maps

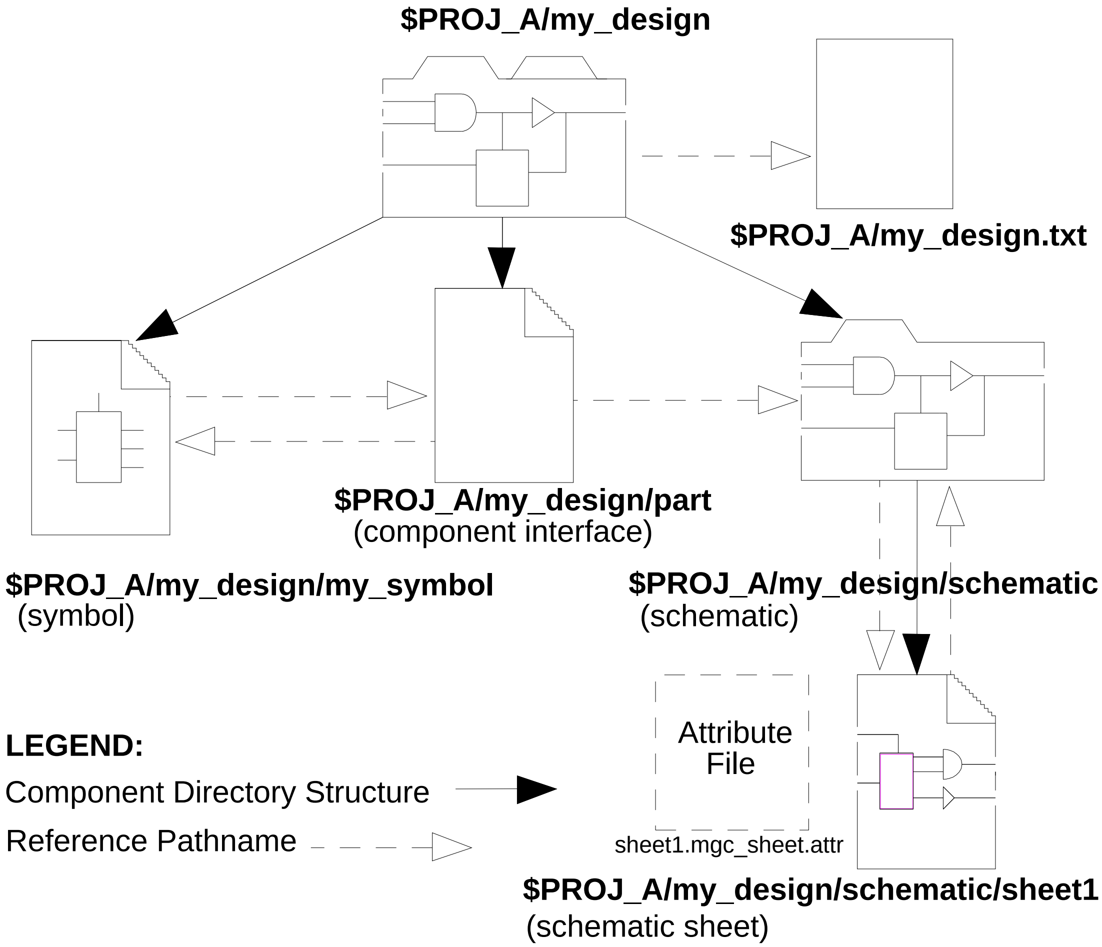
Soft prefixes are a Mentor Graphics mechanism for sharing resources, such as parts libraries and project data, among users on networks that might contain different types of workstations with different operating systems and configurations. Soft prefixes make it possible for users to access these shared resources in the same way from any workstation on the network, even though different workstations may require different hard pathnames to get to the same resource.

Your system administrator will set up and maintain the soft prefixes used at your site in a mapping file called a *location map*. To give you access to this location map, you may need to define the shell environment variable $MGC_LOCATION_MAP. You should contact your system administrator about defining this environment variable and about the set of soft prefixes that your site supports.

Once your system administrator has set up a location map and has defined the $MGC_LOCATION_MAP environment variable, you should be able to use the soft prefixes you will find in this training workbook. When you see an example that uses a soft prefix as part of an application command or as a response to an application prompt, you can type the soft prefix just as it appears in the manual or substitute the corresponding soft prefix in use at your site. If you experience any problems using soft prefixes, contact your system administrator.

The search path for a location map is shown graphically at the bottom of the facing page. The system always looks first for a pathname that is defined by the environmental variable $MGC_LOCATION_MAP. If not found, it looks for a file named **mgc_location_map** in the current working directory, then in the location $MGC_HOME/etc, then in the location $MGC_HOME/shared/etc. As soon as it finds a map in this search, the search stops and that map is used.

Design Object References



- **Reference - a pointer between two design objects**
- **References are keep in associated ".attr" file**
- **You can check and modify references using Design Manager functionality**

Design Object References

A *reference* is a pointer from one design object to another design object. Each reference consists of the pathname to the design object, its type, and a version specifier. References show relationships between design objects.

Design Architect creates references within the component. You can create and modify references using the Design Manager. You can modify DA-created references using the Design Manager or Design Architect. However, if you modify these references within the Design Manager, you run the risk of modifying them incorrectly.

You can display design object references by invoking the **Report > Show References** menu path in the Design Manager pulldown menu or you can click on the **Show References** (right arrow) icon in the Navigator window. The Design Manager also provides other navigation buttons to allow you to explore a design object's references, enabling you to traverse the design hierarchy.

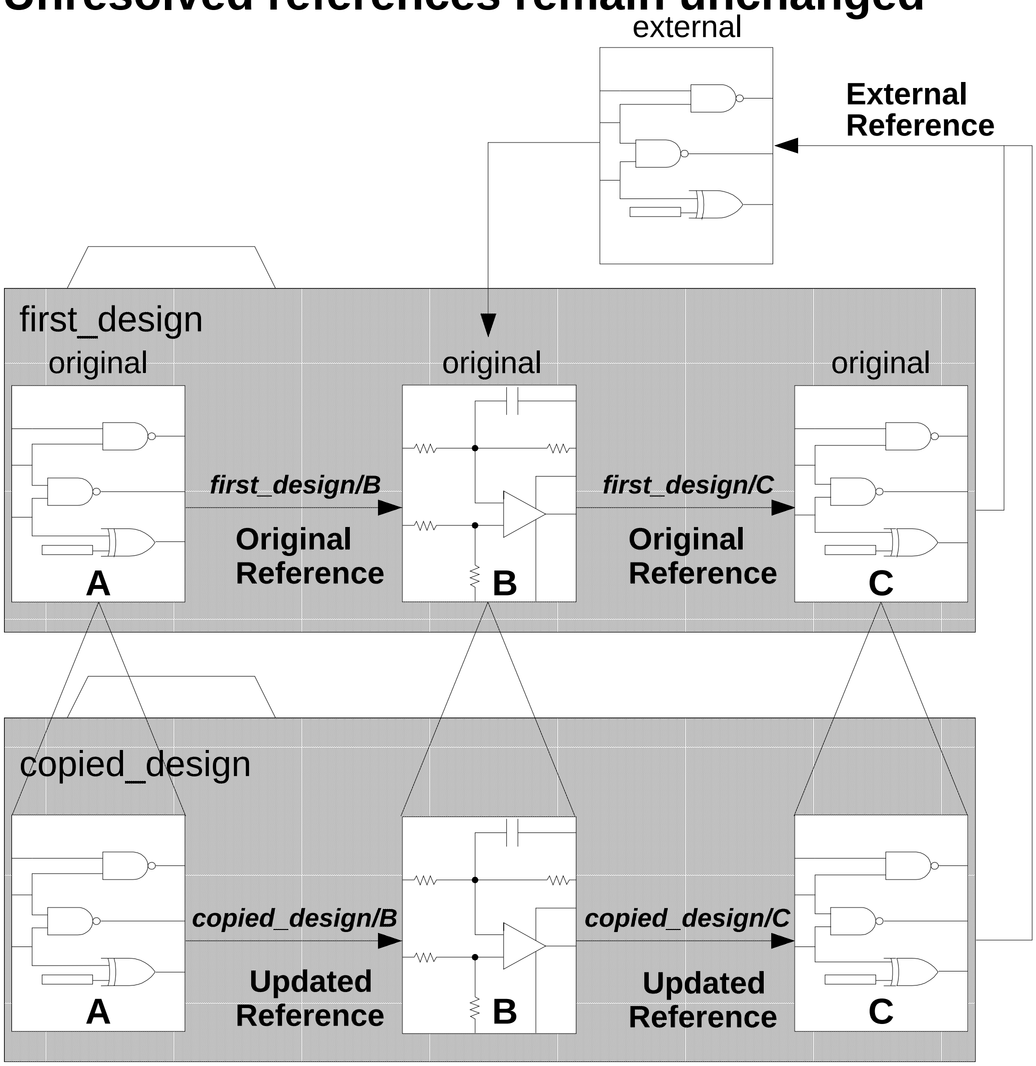
The illustration on the facing page shows two concepts: (1) the component hierarchy and the reference pathnames, and (2) references that are created by the Design Manager, and those that are created by Design Architect. The component hierarchy is defined by the solid arrows; the reference pathnames are defined by the dashed arrows.

When a symbol or schematic sheet is created, they are placed under the specified component. Reference pathnames are automatically defined by Design Architect. Thus, the symbol model *$PROJ_A/my_design/my_design* contains a reference to the component interface *$PROJ_A/my_design/part*. Notice that *$PROJ_A/my_design/part* also contains a reference to the symbol *$PROJ_A/my_design/my_design*.

The schematic model also contains a reference to *$PROJ_A/my_design/part*. In addition, the schematic model contains a reference to the sheet *$PROJ_A/my_design/schematic/sheet1* and *sheet1* contains a reference back it its parent *schematic*. The references for sheet1 are keep in an associated file called a ".attr" file.

Copying Design Objects

- **In Design Manager**
  - o **Copy by containment or by reference**
- **Resolved references are updated**
- **Unresolved references remain unchanged**



Copying Design Objects

You can use the Design Manager to copy your design or component. The Design Manager copies objects in two modes: by reference or by containment. The simple copy uses containment to determine which objects are copied. All objects contained in the component container are copied.

All resolved references are automatically updated. <u>Any references that were unresolved before the copy operation remain unchanged</u>. You can modify these unresolved references to point to existing objects using the Design Manager. The Design Manager allows all objects within a design object to be copied.

The figure on the facing page shows schematic reference updating during a simple copy operation. A set of schematics A, B, and C is copied from the *first_design* directory to the *copied_design* directory. In the directory, *copied_design*, the references of A, B, and C now point to local locations. The Design Manager automatically updates references. If you did not use the Design Manager to copy, the original references would have remained unchanged.

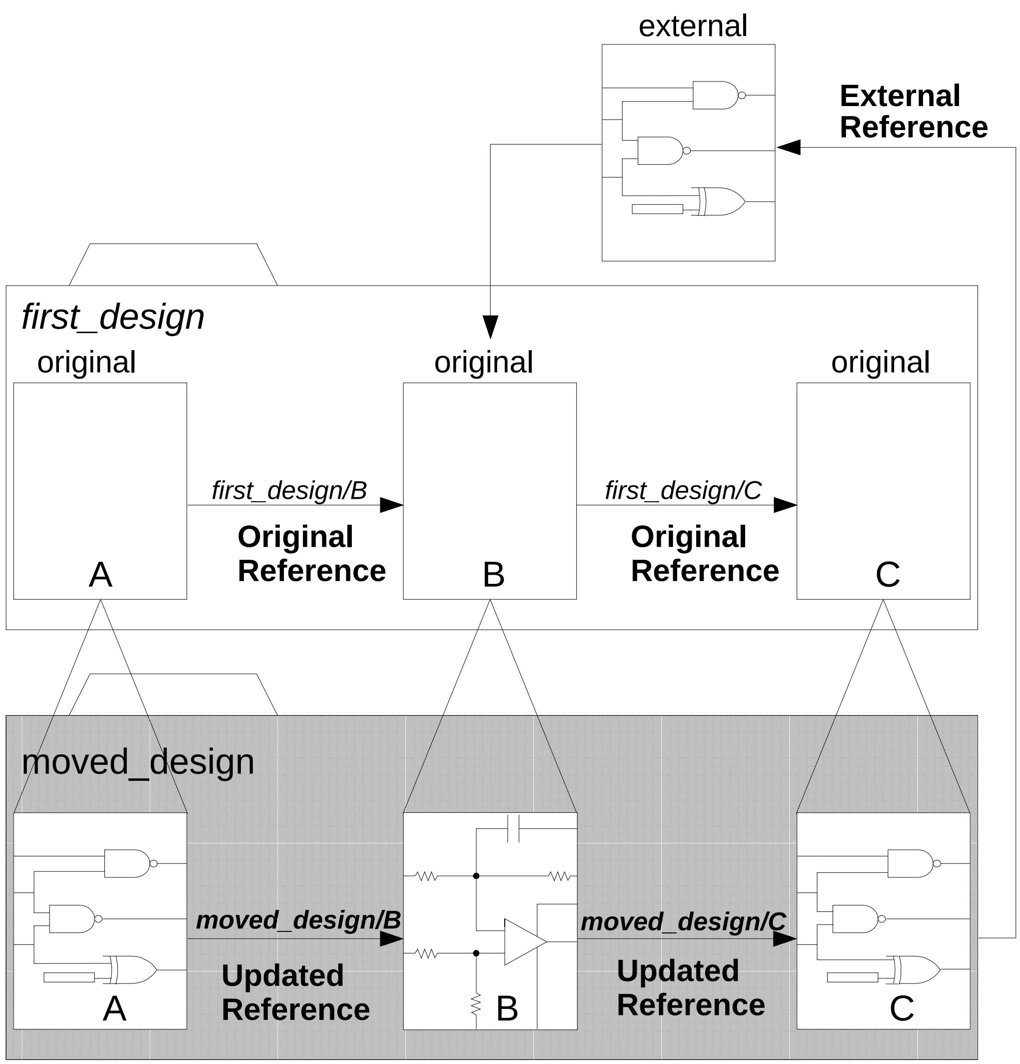
References to external objects are also copied. You can optionally choose that the Design Manager copy the referenced objects into the copied design configuration, instead of just copying the references. Thus, any design object referenced by the original design can optionally become part of the new design configuration. However, any references attached to outside objects that point to the original design remain the same; that is, they reference the original object and not the new copied object. You need to understand the design well enough to decide whether the copied design should reference other objects like the original design, or whether the copy should include the objects in the new configuration.

Note

If you copy a schematic or sheet model within a component, you must register the newly-copied model within Design Architect. You will learn more about model registration later in this training course.

Moving and Deleting Design Objects

- **Resolved references are updated**
- **Unresolved references remain unchanged**
- **References to moved/deleted object outside or at the same level as the component are not updated**



Moving and Deleting Design Objects

You can use either the Design Manager to move or delete a design or component. You can also use Integrated Design Management (iDM) functions, which enable you to manage design data that you create in Mentor Graphics applications by providing easy access to copy, move, delete, and change references.

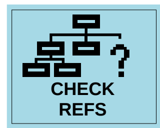
In essence, a moved design object is one that is copied and then deleted. When you move an object within a component or design, all objects in the containment hierarchy of the selected object are also moved. When you move design objects that refer to other design objects in the selected set, the Design Manager automatically updates those references to reflect the new location. Unresolved references or references to external objects remain unchanged.

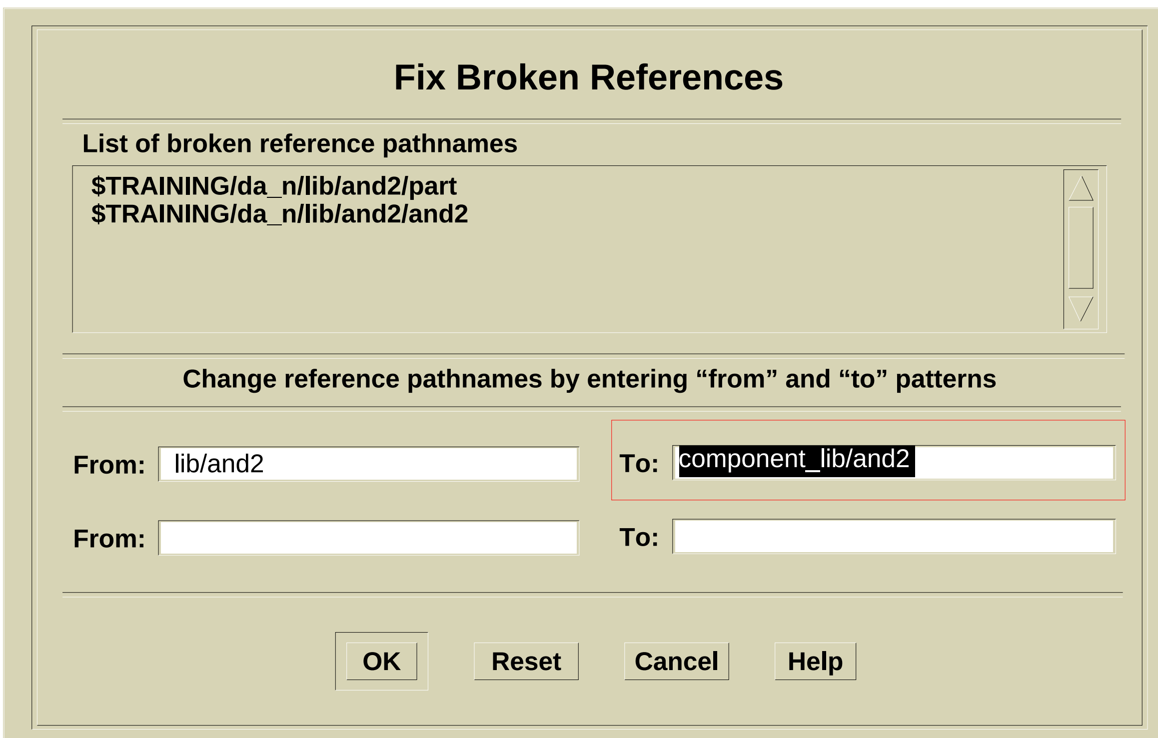
Any design objects that reference the moved object within the containment of a component are also automatically updated to reflect the new location. However, any external references that point to the moved object are not updated; that is, they continue to point to a design object that no longer exists. You need to understand the design well enough to decide whether moving the design will break external references.

The Design Manager allows all objects within the component container to be moved with the exception of symbols. Symbols are not allowed to be moved outside of their owner component container. It you attempt to move a symbol outside of its component container, Design Manager issues an error message and does not move the symbol.

When you delete an object, all objects and references in the containment hierarchy are also deleted. Any external references that point to the deleted object are not updated and are, therefore, invalid; that is, they simply point to an object that no longer exists.

Checking and Changing Design References





Checking and Changing Design References

It is a good practice to always check for broken references before you invoke an application on design data that has been relocated. You can check for broken references by selecting the icon in the Navigator window that represents the top (root) level of the design. When you click on the CHECK REFS icon in the Design Manager palette, a search for broken design references begins at that point and extends down the tree.

Broken references are reported in the Fix Broken References form as shown on the left. You may fix a broken reference by typing the complete “old” pathname in the left entry box, then the complete “new” pathname in the right entry box. To save time typing, you may just specify a pathname segment. For example, in the illustration on the left, the broken reference was created by someone changing the **component_lib** directory name to **lib**. You may fix this broken reference by replacing the pathname segment **lib/and2** with **component_lib/and2**. The important thing to keep in mind is to make sure that the pathname segment is unique, so you won’t be changing similar correct pathnames in other branches from right to wrong.

It is a good practice to always repeat the CHECK REFS operation until the message “**No broken references were found**” is reported.

Lab Exercises

Exercise 1: Copying the Training Data

In this training course, you will be creating a hierarchical design from training data supplied from a read-only master location. Since you will be modifying the data, you must first copy the data to a place in your local directory structure.

1. Log in to your workstation.

   Log in by entering your user name, password, and home directory pathname assigned by your Instructor.

   User Name: ____

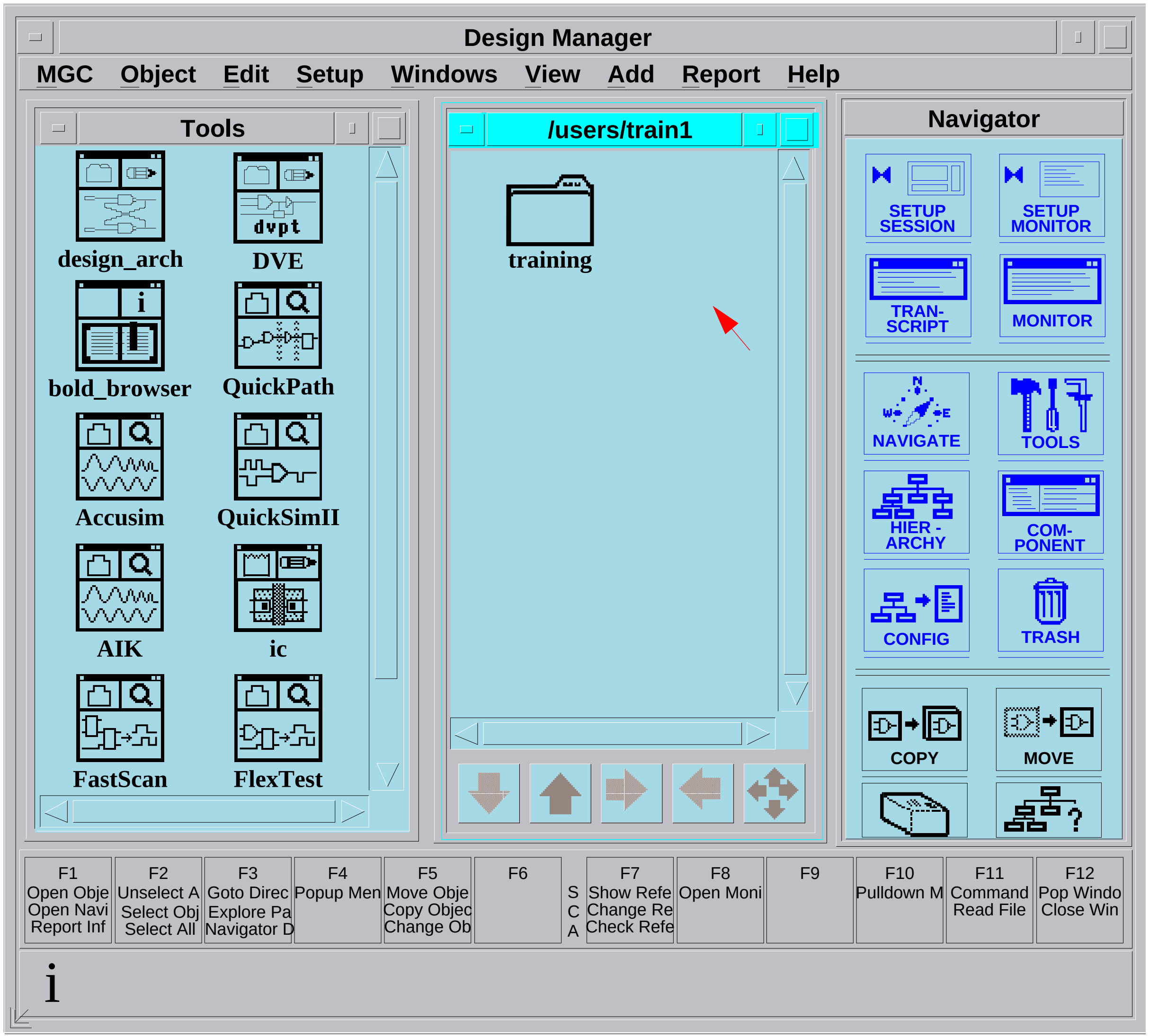
   Password:____

   Home Directory pathname:_____________________________

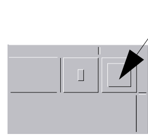
2. Bring up a shell, set the directory to your home directory and enter the following command:

   ```
   $ $MGC_HOME/bin/dmgr
   ```

After a few minutes, a Design Manager window should come up and appear similar to the following illustration:



3. Click the Maximize button to fill the screen with Design Manager (or use the “Full size" window menu option).



(Note: some window environments may have a menu choice that performs this function.)

Within the **Session** window, the **Tools** window is on the left and the **Navigator** window is on the right. The Navigator window displays the contents of the current working directory which is typically the shell directory from which Design Manager was invoked.

The Navigator window is the active window which is identified by the blue border.   All the pulldown menus in the top banner are associated with the Navigator window at this time. The icons you see in the Navigator window reflect the contents of you working directory.

The Tools window displays the tool icons representing the applications in your hotbox directory.

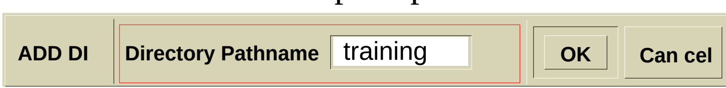
4. If the *training* directory doesn't already exist, create it now as follows:

   a. Select the following menu item from the Design Manager menu bar:

   Add > Directory:

   The Add Directory prompt bar is displayed.

   b. Fill in the ADD DIRECTORY prompt bar as shown below.



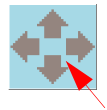
   c. Click **OK**. You will see the *training* directory added to the list of other objects displayed in the Navigator window.

5. View the contents of the *training* directory.

   a. Double Click on the *training* directory icon to expose its contents. There should be nothing in the training directory at this time.

6. Navigate to the *$MGC_HOME/shared/training* directory as follows:

a. Click the **Go To** button in this Navigator window as illustrated below:

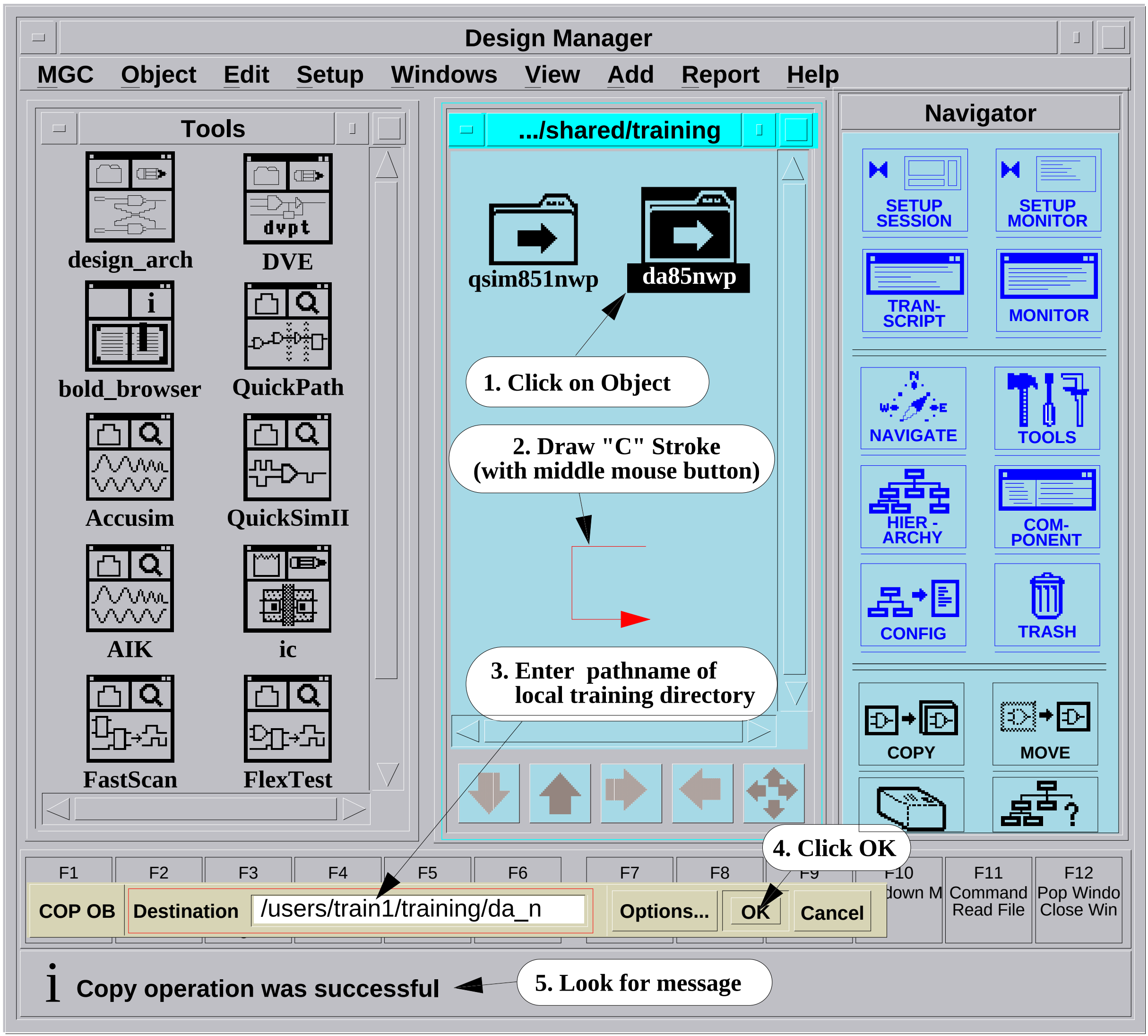


b. Enter the following in the text entry box: **$MGC_HOME/shared/training**

c. Click **OK**.

The Navigator window displays the **$MGC_HOME/shared/training** directory in the master tree.

7. Follow the directions in the illustration on the next page to copy the **da85nwp** training data to your local training directory.



The hourglass prompt should appear while the data is being copied. After the prompt disappears, and the Copy operation was successful message appears, navigate to the *$HOME/training* directory and the **da_n** icon should appear.

8. View the contents of the **da_n** directory, by double clicking on the **da_n** icon.

   The contents of this directory includes the following:

   - **ample_source --** contains information that you will use in customization exercises later in this course.
   - **card_reader --** an empty directory that will eventually be turned into a component container.
   - **com --** contains some files that you will use later in this course.
   - **component_lib --** contains some library components that you will use later in this course.

Exercise 2: Using Notepad to Create and Modify ASCII Files

During this training course you will have the opportunity perform several user interface customizing exercises. You customize the user interface by creating ASCII files containing AMPLE code. You then save these files in predefined locations. In the following exercise, you will be introduced to the Notepad editor which is a tool used in customizing. Because the use of strokes will increase your productivity, you will begin learning simple strokes that perform tasks like Copy, Move and Delete. These same strokes will be used throughout this course in applications like the Schematic Editor and the Simulator to perform similar tasks.

Creating a New File with Notepad

1. From the Design Manager Session Window, choose:
   MGC > Notepad > New

   A Notepad window appears on the screen, identified by the name **Notepad - (untitled)** in the title area.

2. Choose the **(Menu Bar) View** > **Wrap** menu item.

3. Click the "**Word**" button under "Mode" and the "**Yes**" button for "Wrap to Window Width."

   This controls the amount of characters on one line in the ASCII file.

4. **Click OK**.

5. Select the pulldown meu item **File > Import...**

6. Click the Navigator button, navigate to the file **...training/da_n/com/notepad_text.ascii**, select the filename, then click **OK**.

7. The following text is imported into the new NotePad session:

   This is a practice Notepad Editor session. Within this practice session, you will search for and replace text, copy text, move text, delete text, and undo the last action.

8. Save the Notepad information to a file as follows:

    a. Choose: **(Menu Bar) File > Save**

    b. Enter: *$HOME/training/da_n/com/**myfile.ascii***

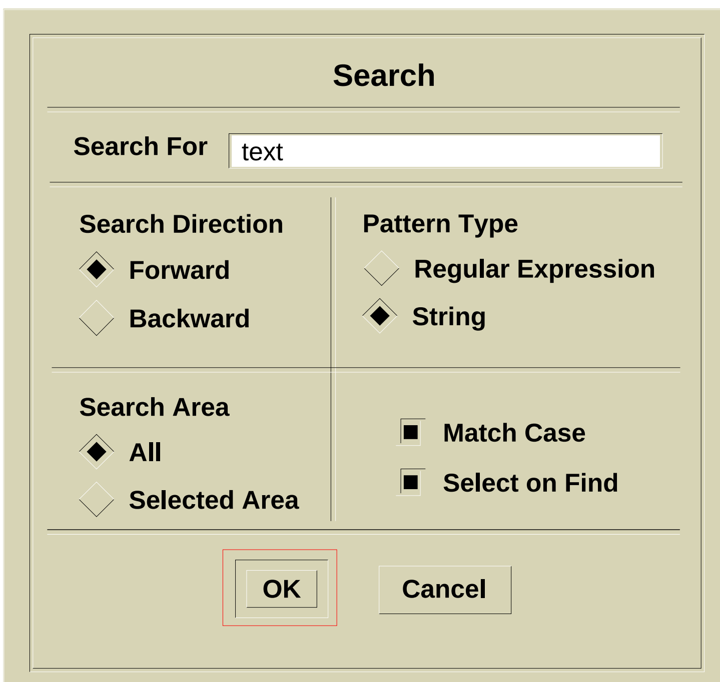
    c. Click **OK**.

Searching for a String

1. Move the cursor to the top of the file by pressing Ctrl-T. (You must position the cursor back at the top of the file each time you wish to search the entire file.)

2. Choose the **(Menu Bar) Search > Search...** menu path.

    Notepad displays the “Search” dialog box. Search for the string “**text**” by filling out the form as shown below:



3. Click **OK**.

4. Notepad highlights the word “**text**” after the word “replace.”.

   This is a practice Notepad Editor session. Within this practice session, you will search for and replace text, copy text, move text, delete text, and undo the last action.

5. Repeat the search by choosing:

   (Menu Bar) Search > Search Again > Forward

6. Notepad highlights the word “**text**” after to the word “copy.”.

   This is a practice Notepad Editor session. Within this practice sessio you will search for and replace text, copy text move text, delete text, and undo the last action.

7. Unselect the string “text” by drawing a “U” stroke .

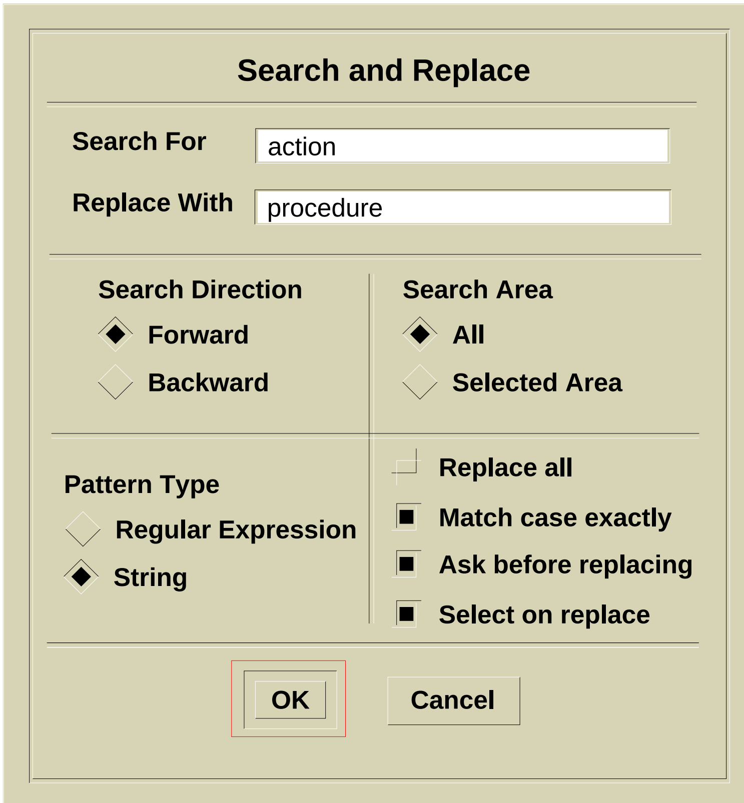
Searching for and Replacing a String

1. Go back to the top of the file by pressing Ctrl-T.

2. Display the “Search and Replace” dialog box by choosing:

   (Menu Bar) Search > Replace...

3. Search for the word “**action**” and replace it with “**procedure**”, by filling out the form as follows:



a. Click **OK**.

The word “**action**” is highlighted in the Notepad window. A “**Replace?**” dialog box asks you to confirm the replace operation.

b. Click **Yes**.

Notepad replaces the string “**action**” with the string “**procedure**”. The message window informs you that one occurrence of “**action**” was found and replaced.

4. Remove the highlighting by drawing a “U” stroke .

5. Move to the top of the file by pressing Ctrl-T.

Practice Copying, Moving and Deleting Text

1. Activate the Notepad window, if it is not already active.

2. Choose **(Menu Bar)View > Show Line Numbers**

3. Select the first sentence by placing the mouse cursor on the first character, then press the left mouse button and drag across the sentence.
   .

   This is a practice Notepad Editor session. Within this practice session, you will search for and replace text, copy text, move text, delete text, and undo the last action.

4. Click the Select mouse button at the end of line 3 to move the insertion cursor.

5. Move the sentence by drawing a “shark fin” stroke .

   The block is moved and remains highlighted.

6. Move the insertion point back to the beginning of line 1 (click there).

7. Copy the highlighted text back to the beginning of the file, by draw a “C” stroke . The block is copied, and the copy is highlighted.

8. Undo the copy action by drawing an up-side-down “U” stroke . The highlighted text is removed. Undo allows you to reverse up to ten (10) edit actions you have made during the session, beginning with the most recent edit action.

9. Select the sentence that you moved to the end of the paragraph.

Within this practice session, you will search for and replace text, copy text, move text, delete text, and undo the last action. his is a practice Notepad Editor session.

10. Delete the selected text by drawing a “D” stroke .

11. Select the word **practice** in the first sentence of the paragraph.

12. Type **temporary**. The word **practice** is replaced by the word **temporary**.

13. Highlight the string **tempory session**.

14. Send a copy of the string to the System Clipboard with a ↑ stroke.

15. Paste a copy back from the System Clipboard to the cursor position with a ↓ stroke. Do this several times.

16. Abandon the edits you made during this exercises by performing the following steps:

    a. Choose: **(Menu Bar) File > Revert to Saved**

    A dialog box appears asking you to confirm discarding the edits you made.

    b. Click **Yes**.

    The Notepad displays the unedited file.

17. **Close** the Notepad window by drawing a → stroke.

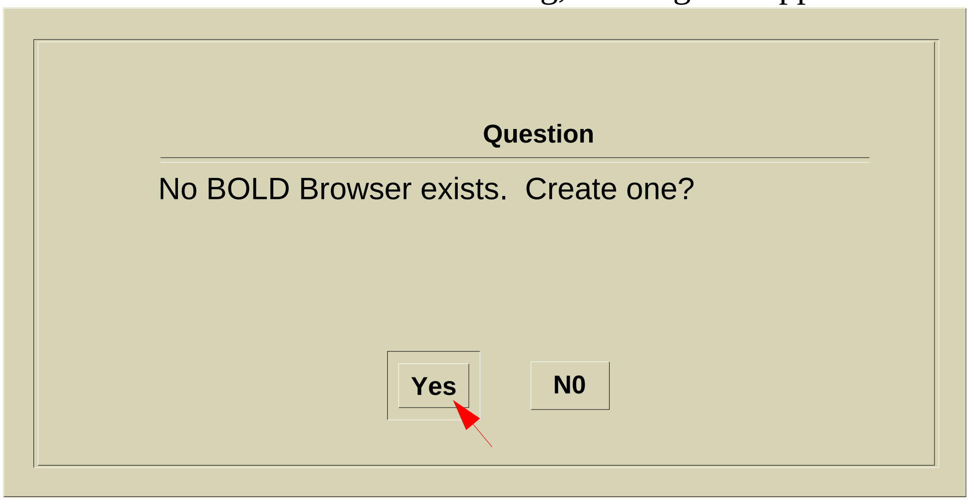
Exercise 3: Viewing and Searching Online Documents

Invoking the BOLD Browser with the Help Pulldown Menu

Start a BOLD Browser session from any application session window by performing these steps:

1. Choose: **(Menu Bar) Help > Open Bookcase**

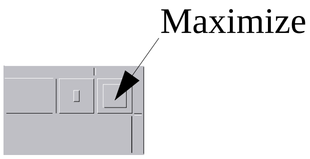
   If a BOLD Browser session is not running, a dialog box appears as follows:



2. Click **Yes**.

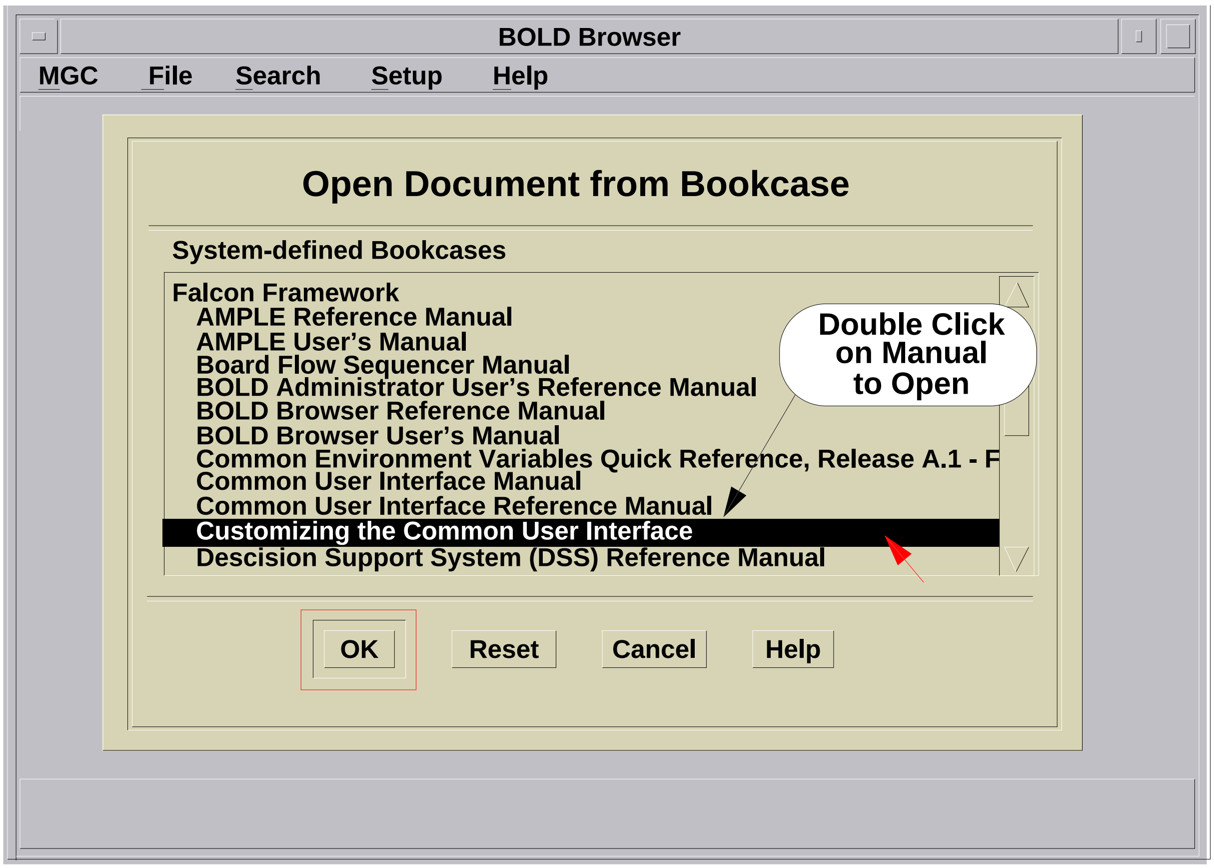
   A BOLD Browser session window appears, with an "Open Bookcase" dialog box displayed.

3. Maximize the window.



   (Note: some window environments may have a menu choice that performs this function.)

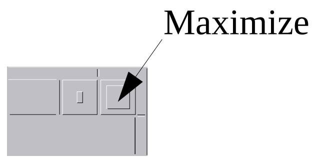
If the BOLD Browser does not display the “Open Bookcase” dialog box, choose the **(Menu Bar) File > Open > Bookcase** menu path and double click on the Falcon Framework bookcase.



4. Double click on **Falcon Framework** bookcase, if it is not already open.

5. Scroll down, then double click on **Customizing the Common User Interface**

6. Maximize the Document Window.

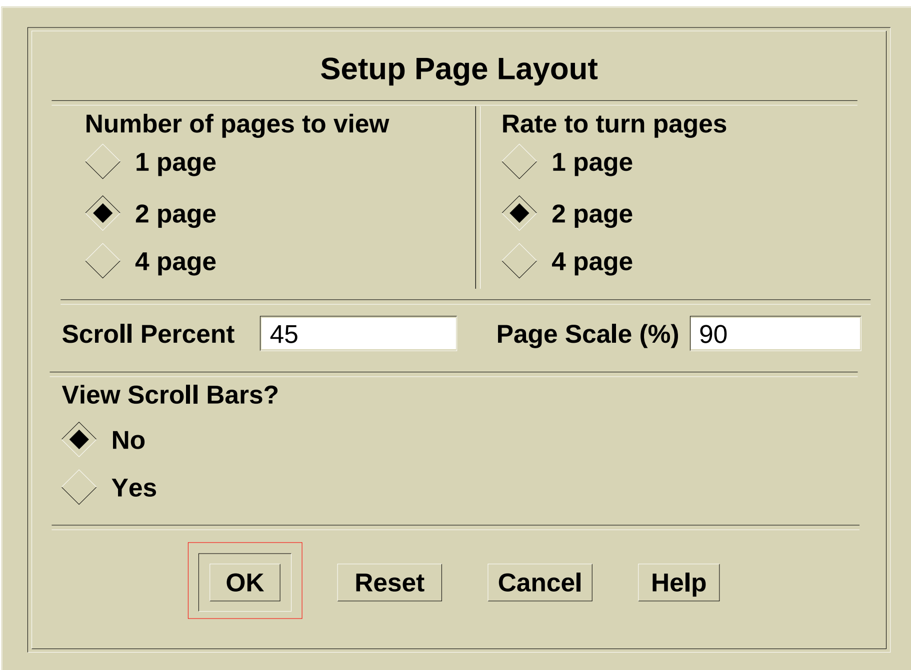


Setting Up the Page Layout.

You can setup the browser to display one, two, or four pages at a time. The default settings allow you to:

- Display one page of the document at a time.
- Turn pages one at a time.
- Scroll 35% of the page when you use the page up or page down keys.
- View the page at 115% of the original page size.

1. Change the Page Layout settings by choosing:
   (Menu Bar) Setup > Page Layout...



2. Change the settings to match the figure above.

3. Click **OK**.

   The document window now displays two smaller (90%) pages. When you use the page up or page down keys, you scroll 45% of the page.

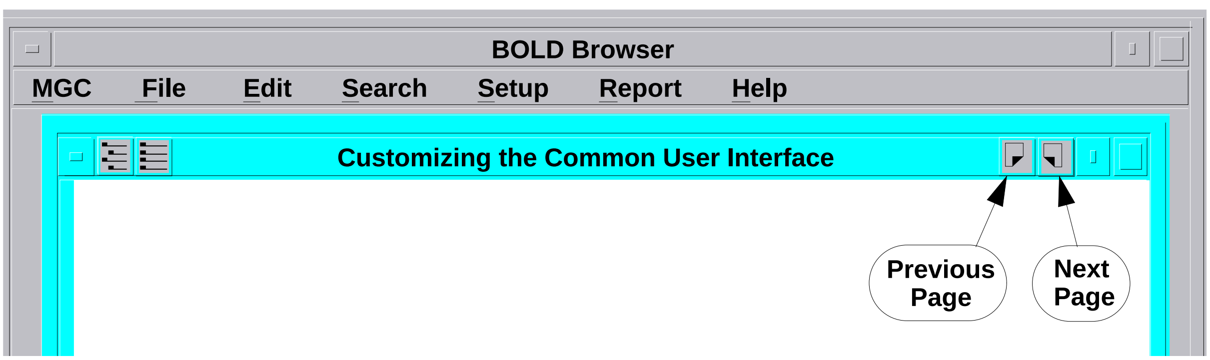
To quickly change the layout, you can press the F9 function key to view one page, SHIFT-F9 to view two pages, and Ctrl-F9 to view four pages. Try this, then return to the two-pages-up layout.

Using the Page Icons

The BOLD Browser provides icons and key definitions to allow you to navigate within an online document.

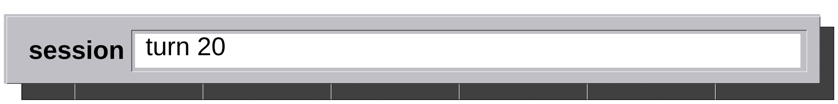
The figure below shows the location of the page icons. If you are located at the beginning of the document, the previous-page icon is not visible. Similarly, if you are located at the end of the document, the next-page icon is not visible.

Practice browsing the document by clicking the Page icons.



Turning Several Pages at a TIme

1. Type “**turn 20**”.



   A popup command line appears with “turn 20” in it.

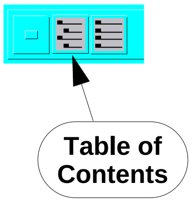
2. Press **Return**. BOLD Browser turns the document forward twenty pages.

3. Type “**turn -10**” and press the **Return**. BOLD Browser turns the document backward ten pages.

Following Hyperlinks within a Document

Hypertext links allow you to travel to a specific location in the document, or to a different document. Practice using hypertext links to travel within a document by performing the following steps:

1. Click on the Table of Contents icon.



2. Move the mouse pointer down the items listed in the table of contents.

   Notice how the mouse pointer changes to a hand with a pointing finger as the mouse pointer moves over the hypertext linked section names or page numbers. The hypertext links are enclosed in a box on black and white workstations. They are indicated by the default color on color workstations, usually blue, when BOLD Browser is invoked with the -Color option.

3. Place the pointer on the Section 2 title block and click.

4. The first two pages in Section 2 are displayed.

Following Hyperlinks to Another Document

Use a hypertext link to travel to another document by following these steps:

1. Click the Left mouse button on $time() in the middle of page 2-1. The BOLD Browser opens the AMPLE Reference Manual to the reference page for this function (page 3-316).

2. Click the Select mouse button on $date() at the bottom of page 3-317. The BOLD Browser moves to the reference page for this function (page 3-83).

Copying a Code Example to a Notepad Window

There are many AMPLE coding examples in the online documentation to help you customize the user interface. These coding examples may be copied directly to a Notepad window, then saved to the appropriate location before being loaded into an application scope.

There are two examples of AMPLE code on pages 3-83 and 3-84 in the *AMPLE Reference Manual*. Practice copying this code from the BOLD Browser to a Notepad window as follows:

1. With the Document window active, choose (menu bar) Edit > Copy Pages...

2. Click on the Notepad radio button and execute the form. A Notepad window comes up with the pages of text in it.

3. Choose (menu bar) View > Show Line Numbers

4. Select and Delete the following lines 1 to 32, 9 to 16, and 28 to 41. The remaining text is two samples of ample code.

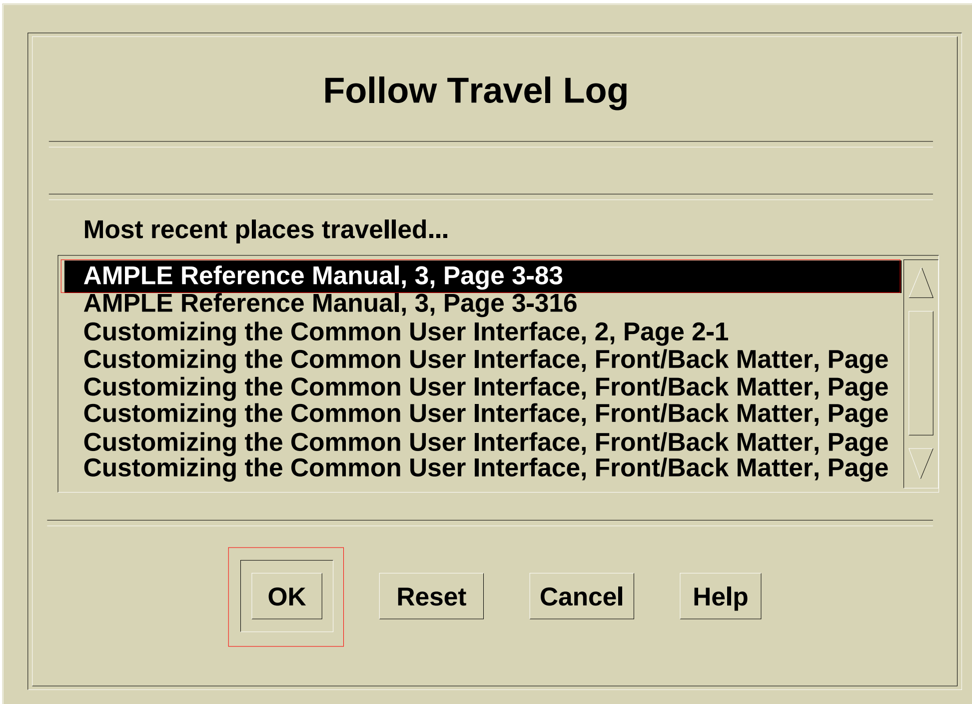
5. Using the **File > Save As...** menu item, save the code to a file named: *$HOME/training/da_n/com/**date_function_ample***

6. Close the NotePad with a horizontal stroke.

Travelling Back Using the Travel Log

The *travel log* is a list of previously viewed pages and manuals. Practice using the travel log by performing the following steps:

1. Choose: **(popup menu) > Travel Log > Summary**



The “Follow Travel Log” dialog box appears as shown above.

2. Select the “**Customizing the Common User Interface, 2, Page 2-1**”.

3. Click OK or execute a stroke.

   The BOLD Browser replaces the *AMPLE Reference Manual* with the document *Customizing the Common User Interface.*

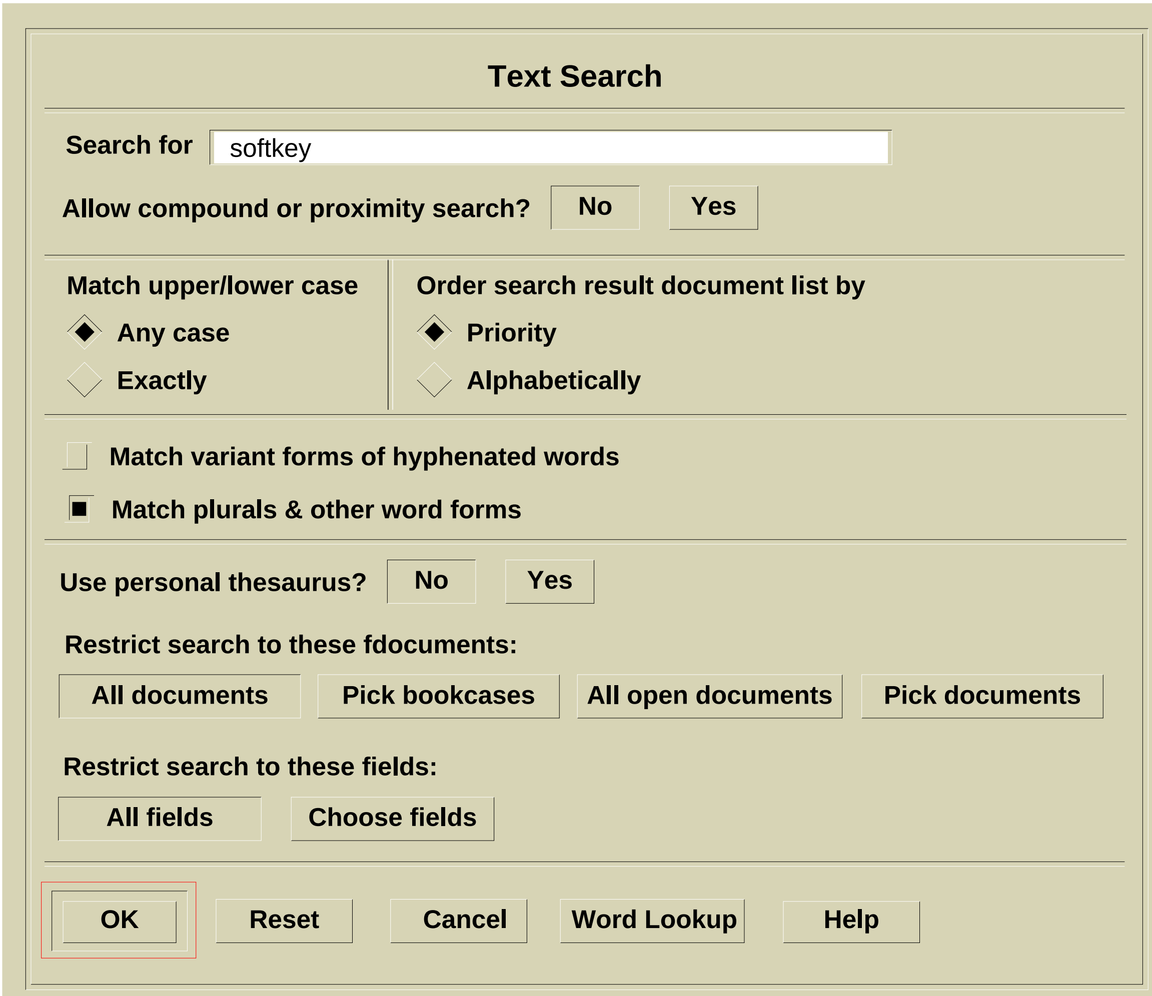
4. **Close** the document window when you are finished experimenting with the travel log.

Executing a Full Text Search

You can search for specific words or groups of words using the Search facility in BOLD Browser. You should have no windows open within the BOLD Browser except for the Session window. If you do, close them before proceeding.

1. Choose: **(Menu Bar) Search > Text Search (choose options)**

The BOLD Browser displays the “Text Search” dialog box as shown below.



2. Enter “**`softkey`**” in the “Search for” text entry box.

3. Click on “**Match plurals & other word forms**”, since there are probably instances for the plural form of “softkeys” also.

4. **Click OK**.

   The BOLD Browser indicates it is searching for the word and then displays a “Search Results” window displaying a list of documents in which the terms “softkey” and “softkeys” were found.

5. Select the “Design Manager Reference Manual” item for which there are seven “hits.” A hit is an occurrence of the search text found in a document.

6. Click on the **Visit** button. You can see a black arrow and line that indicates the exact location of the searched text.

7. **Close** the document window, and select and visit other “hits” in other documents.

8. You can end the application session at any time by double-clicking on the window menu button.

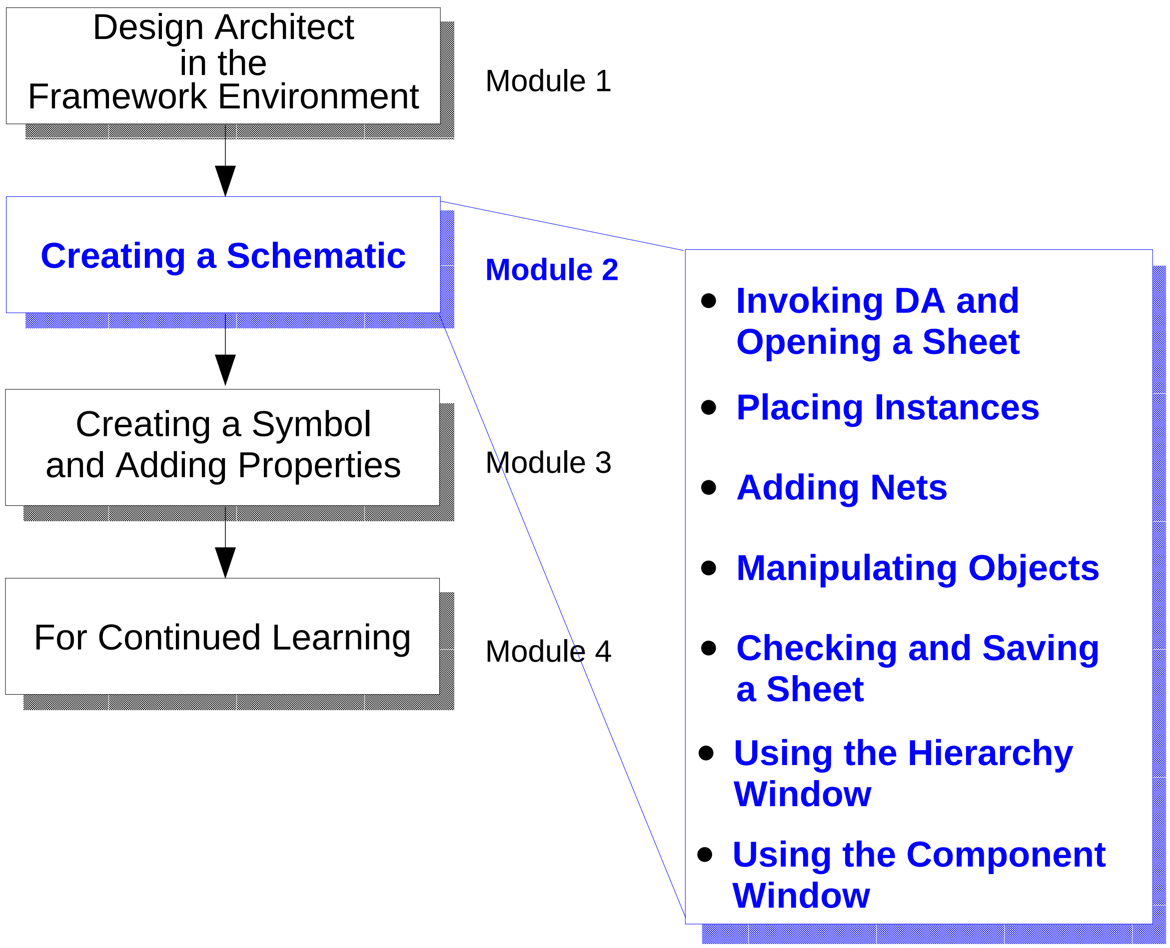
End of Lab Exercises

This concludes the lab exercises for this module.

Module 2
Creating a Schematic

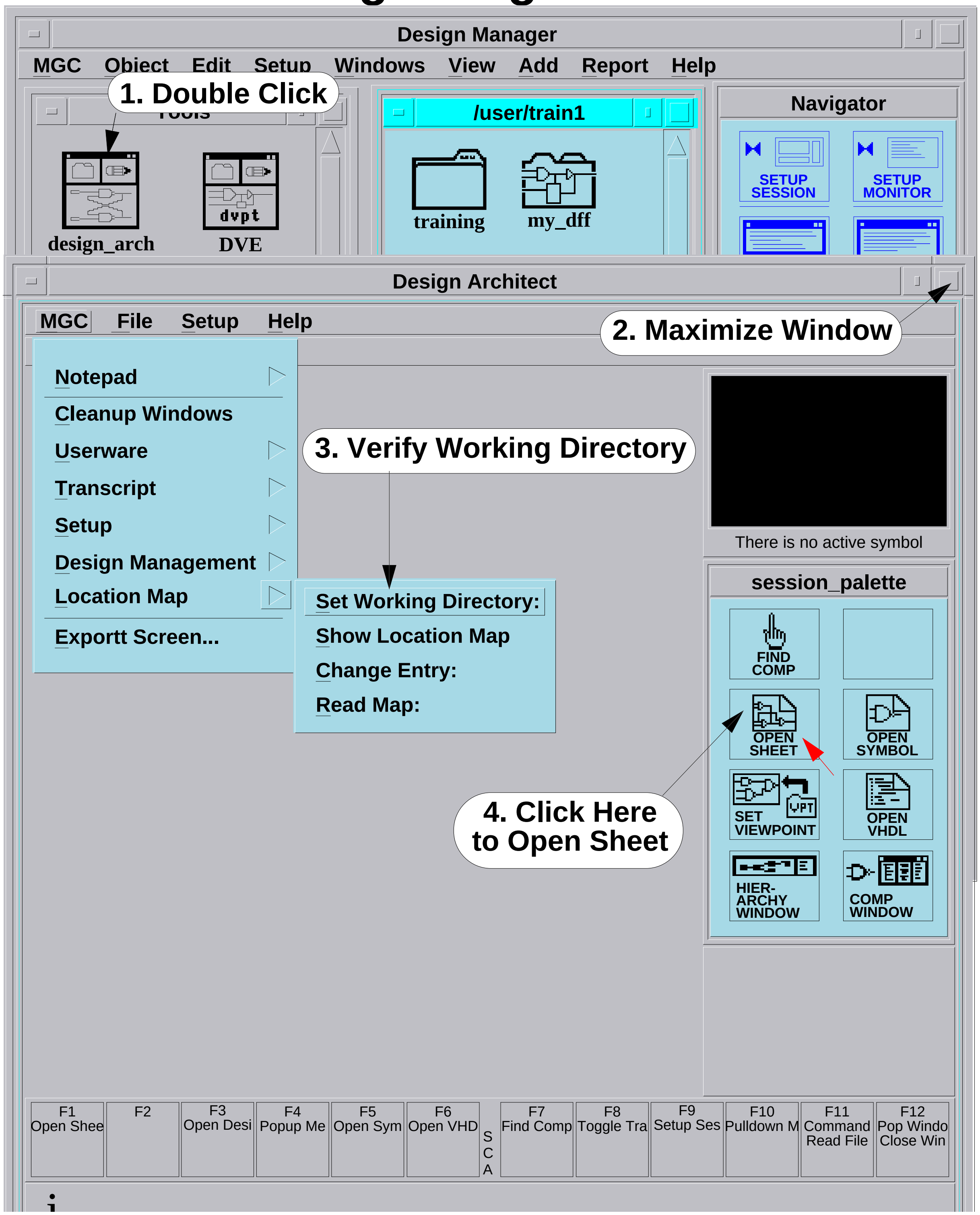
Lesson Creating a Schematic ____ 2-3
Lab Exercises ____ 2-59

Module 2 Overview



Lesson
Creating a Schematic

Invoking Design Architect



Invoking Design Architect

You invoke Design Architect from the Design Manager in one of two ways:

- Double Click on the **design_arch** icon in the Tools Window, or
- Select a component icon in the Navigator window, press the right mouse button, and choose **Open** > **design_arch**.

Design Architect can also be invoked directly from a shell by typing *$MGC_HOME/bin/da*.

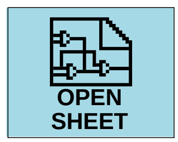
After Design Architect comes up, it is a good practice to Maximize the window, then verify the setting of the working directory. Normally, the working directory will be set to the pathname specified by the shell environmental variable $MGC_WD, if this variable is defined; otherwise the working directory is set to the same location as the shell from which DA is invoked.

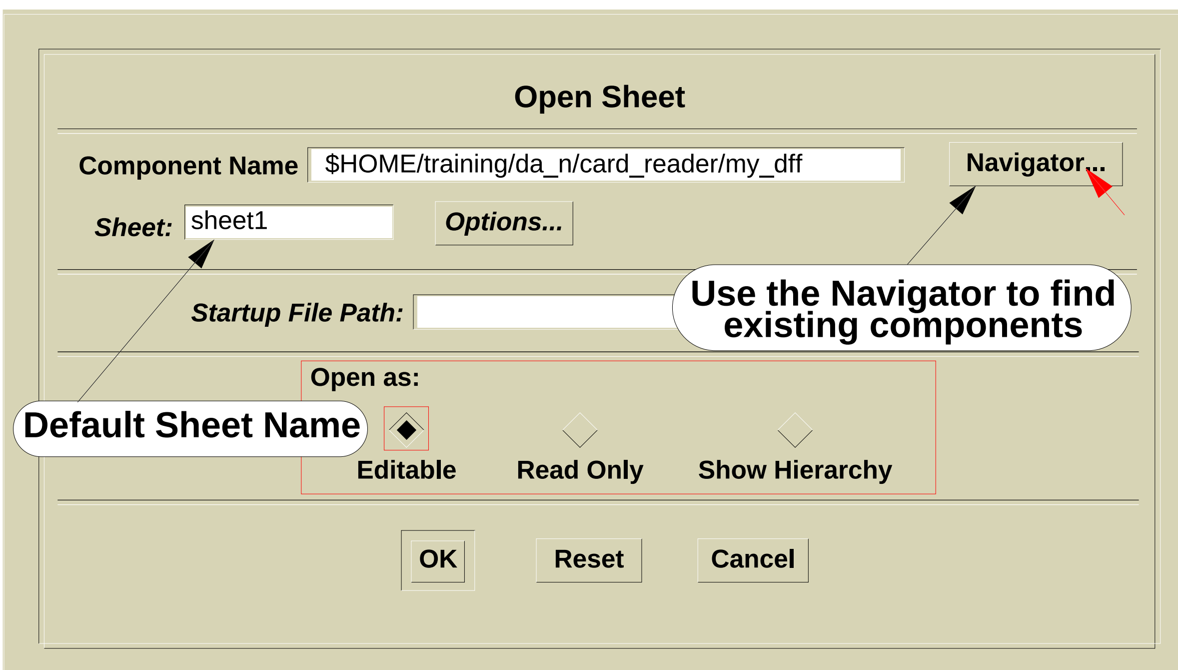
It is always best to verify that the working directory is set to the location where you want it, because all relative pathnames that you enter will be considered relative to the setting of the current working directory.

The tools within the Design Architect environment are represented by icons in the palette on the right. You click on the OPEN SHEET icon to bring up the Schematic Editor window.

Opening a Schematic Sheet

- **Click the OPEN SHEET icon**
- **Enter the new component pathname**
- **Use the Navigator button for existing component**





The Open Sheet Dialog Box

You click on the OPEN SHEET icon to open the Schematic Editor Window. If you specify a pathname for a component that doesn't exist, Design Architect will create a new component structure by that name at that location. Design Architect will then open a new (blank) schematic sheet for that component.

If the component exists, it is often easier to click the Navigator button, navigate to the component location, select the component icon and execute the form.

The sheet name is automatically set to "sheet1" by default. You can create additional sheets later by changing this name before you execute the form.

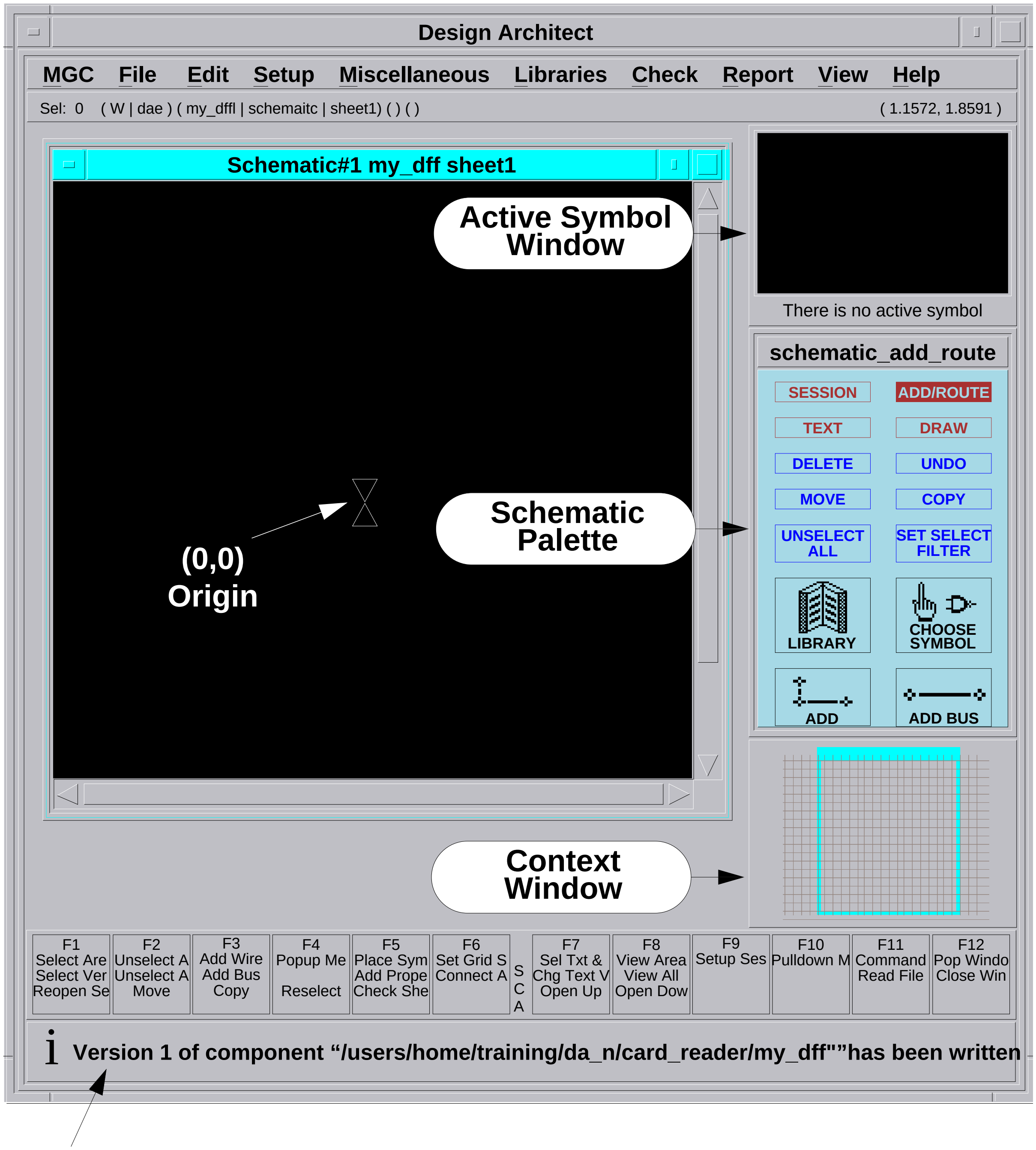
Clicking **Options** allows you to specify other tasks such as:

- Edit a sheet in another (non-default) schematic model.
- Create a new schematic model and/or a new sheet that contains a sheet border and title block information.

You can also open the sheet in edit mode, read only mode, or display the component hierarchy.

Note If you provide a relative component pathname that does not begin with the dollar sign ($) character, that relative pathname will be converted to an absolute pathname based on the value currently set for the working directory. If it is not set properly, the new component structure may be created in a location that you did not intend.

The Schematic Editor Window



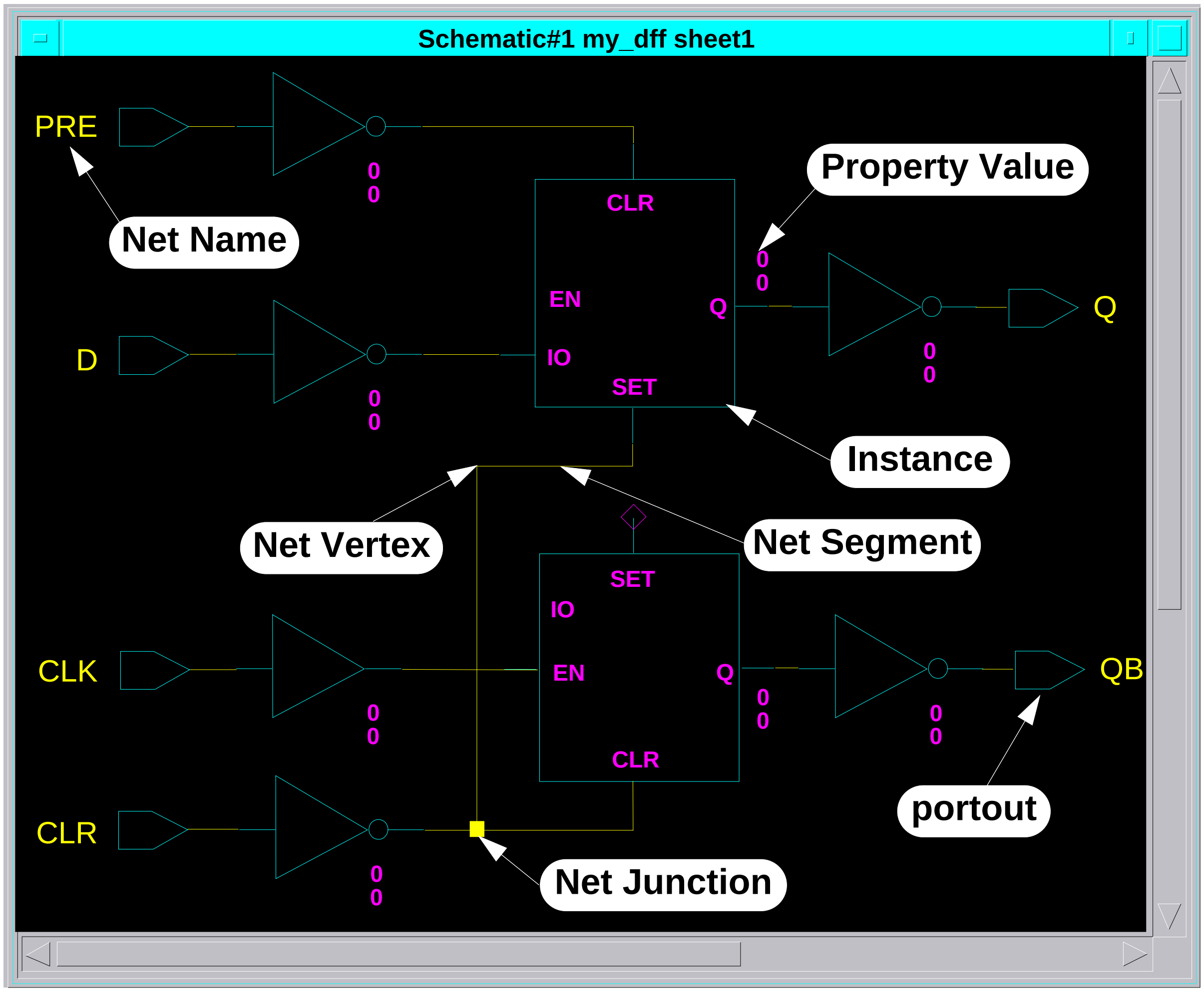
Message indicating the creation of a new component

The Schematic Editor Window

The Schematic window for a new schematic comes up blank with the origin (0,0) set in the center of the screen. A message in the Message window tells you that a new component structure has been created at the specified pathname location. The Design Architect session window changes as follows to accommodate schematic editing tasks:

- **Menu bar(at top)**. Contains the names of the Session pulldown menus; use the Select (left) or Menu (right) mouse button to access a pulldown menu.
- **Active Symbol window**. Located to the upper-right corner of the client area, displays the currently active symbol.
- **Palette menu**. Located to the right center of the client area, contains icons providing common functionality associated with the active schematic window.
- **Context window**. Located in the lower-right corner of the client area, displays where the edit window is located relative to the extent of the sheet being edited.
- **Softkeys**. Located above the message area, contains text describing the function keys for the active window.
- **Message area**. Displays notes, warnings, and errors.

Elements of a Schematic



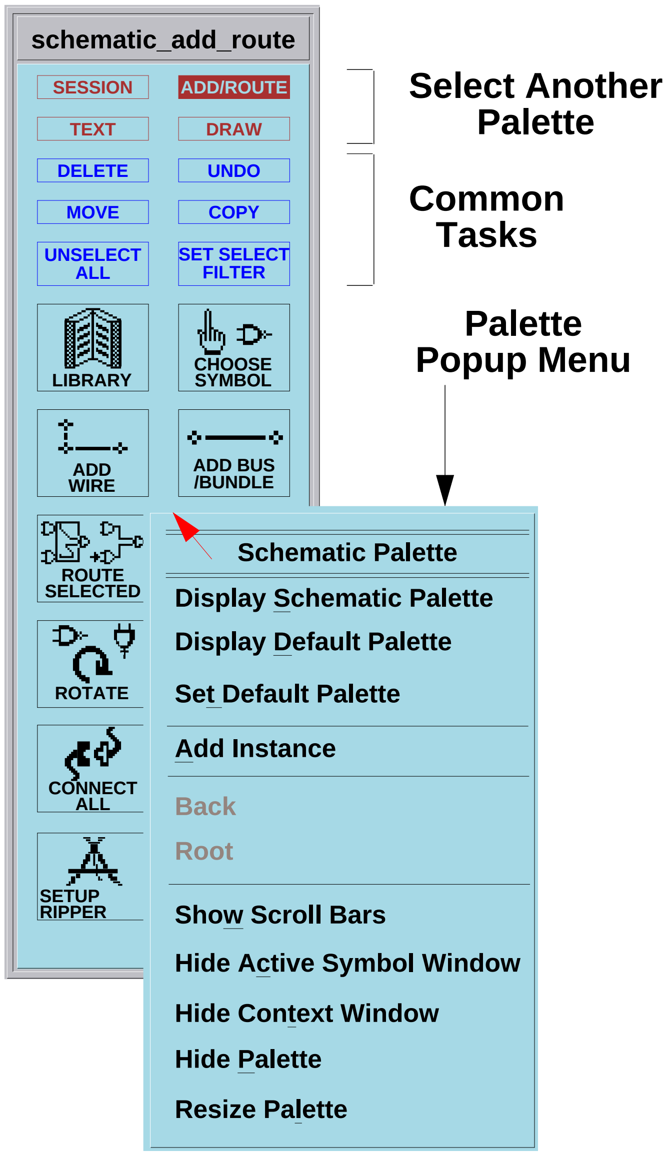
Elements of a Schematic

A schematic is a functional model of an electronic circuit that is made up of the following elements:

- **Instances of symbols.** An instance is like an active reflection of a symbol in a library. There may be more than one instance of the same symbol on a sheet. Instances have a body(colored blue) and pins(colored magenta). The pins on an instance are different than the pins on the symbol because they have a builtin net vertex, a place for attaching nets.
- **Nets.** The term "net" is short for network. A net is a wiring connection between two or more instance pins. The term net refers to either a single wire or a collection of wires called a bus. A bus is sometimes called a bundle or a wide net. A net may contain net vertices (corner points), net segments (the piece between two vertices) and net junctions.
- **Properties.** A property is a name/value pair (text) that represents a characteristic of the schematic that can't be represented graphically. Properties are typically attached to an "owner" object and take on the color of that object. Only the value of the property is visible on the schematic. For example, in the schematic on the left, the net in the upper-left corner has a NET property with a value of PRE. PRE is yellow, the color of the net (the owner). Attached to the output pin of each inverter are two properties with the value of 0 and 0. Since only the values are shown, you can't tell what the names of the properties are just by looking at them. By convention, the upper value is usually the RISE(time) property and the lower is FALL(time). Later you will be shown a way to quickly gather information on an unknown property value.
- **Comment text and graphics.** These objects are non-electrical display information (colored green). The sheet border and title block (not shown) are examples of comment text and graphics.

schematic_add_route Palette

- **Default palette: displayed on invocation**

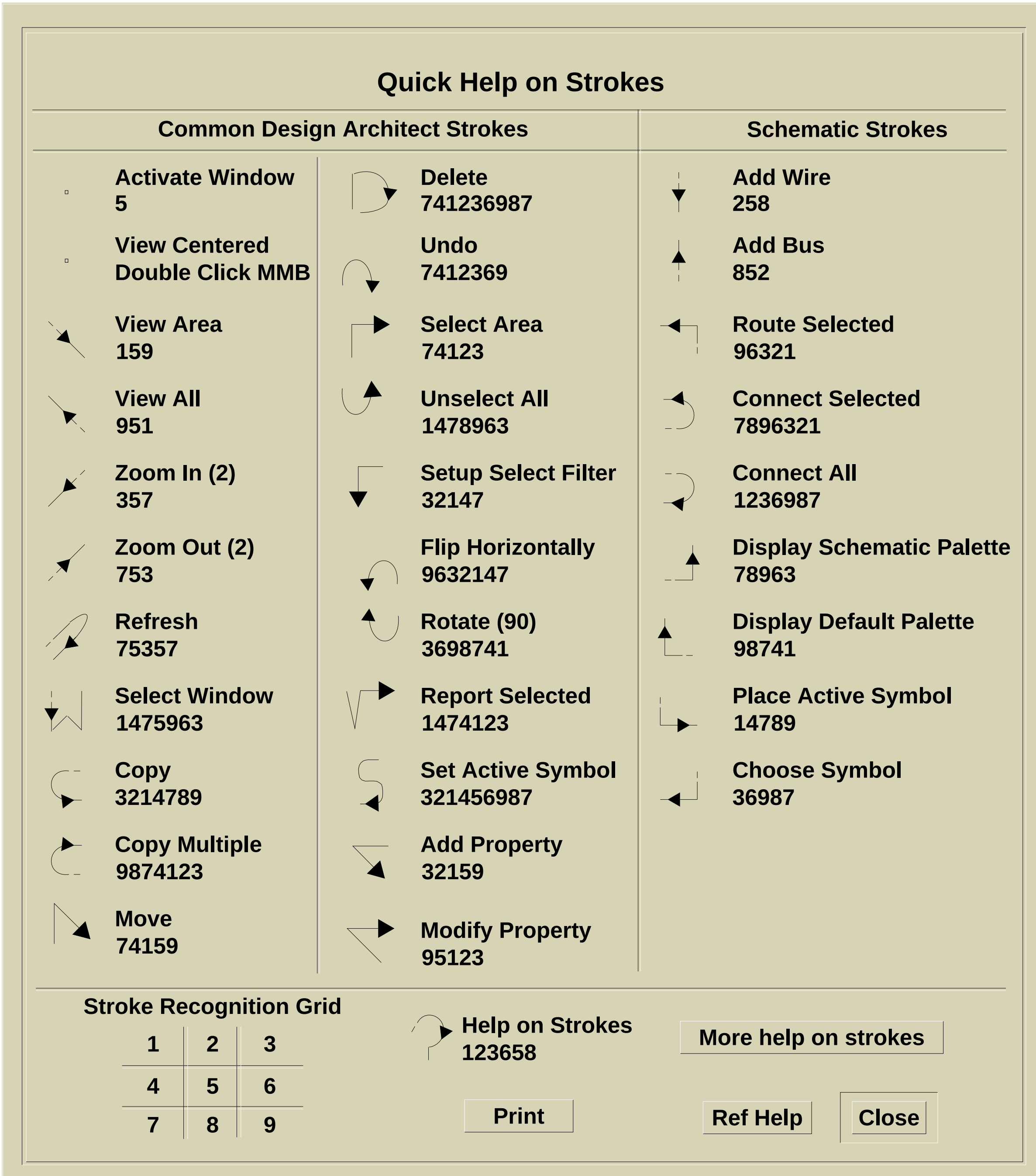


schematic_add_route Palette

The ADD/ROUTE palette is the main palette used for creating the graphic elements of a schematic. The most common tasks for manipulating graphic objects are represented by the blue buttons near the top. The icons in the lower portion also represent common tasks that will be discussed later in this module.

The palette area has a popup menu that allows you to reconfigure what you see. If the palette area is squeezed on the screen so that you can’t see all the icons, you can add scroll bars. Another option is to hide the context window or the active symbol window to make the palette area longer.

Design Architect Strokes



Design Architect Strokes

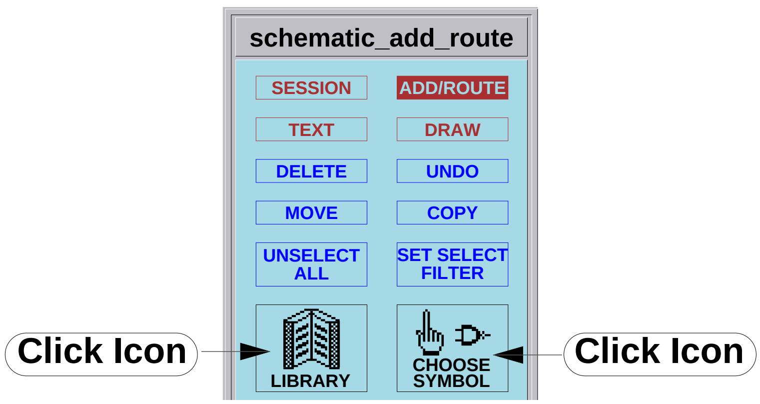
The concept of strokes was discussed in Module 1 and will be emphasized in this course as one of the most productive methods for schematic entry. The default strokes available in Design Architect are shown on the left. This form can be displayed by drawing a question mark “?” stroke.

It is often helpful to make a photocopy of this form, cut it up into strips and tape the strips on the edges of your display. After you use the strokes over time, you will remember them and they will come to you naturally, almost without thinking.

You have already learned some of the common strokes like “C” for copy, “U” for unselect, and “D” for delete. Many other strokes that you will learn in this module will also carry over to other applications.

Placing an Instance on a Sheet

- An instance is like an active reflection of a symbol in a library
- Placing an instance on a sheet is called "instantiating"
- Click LIBRARY or CHOOSE SYMBOL icon



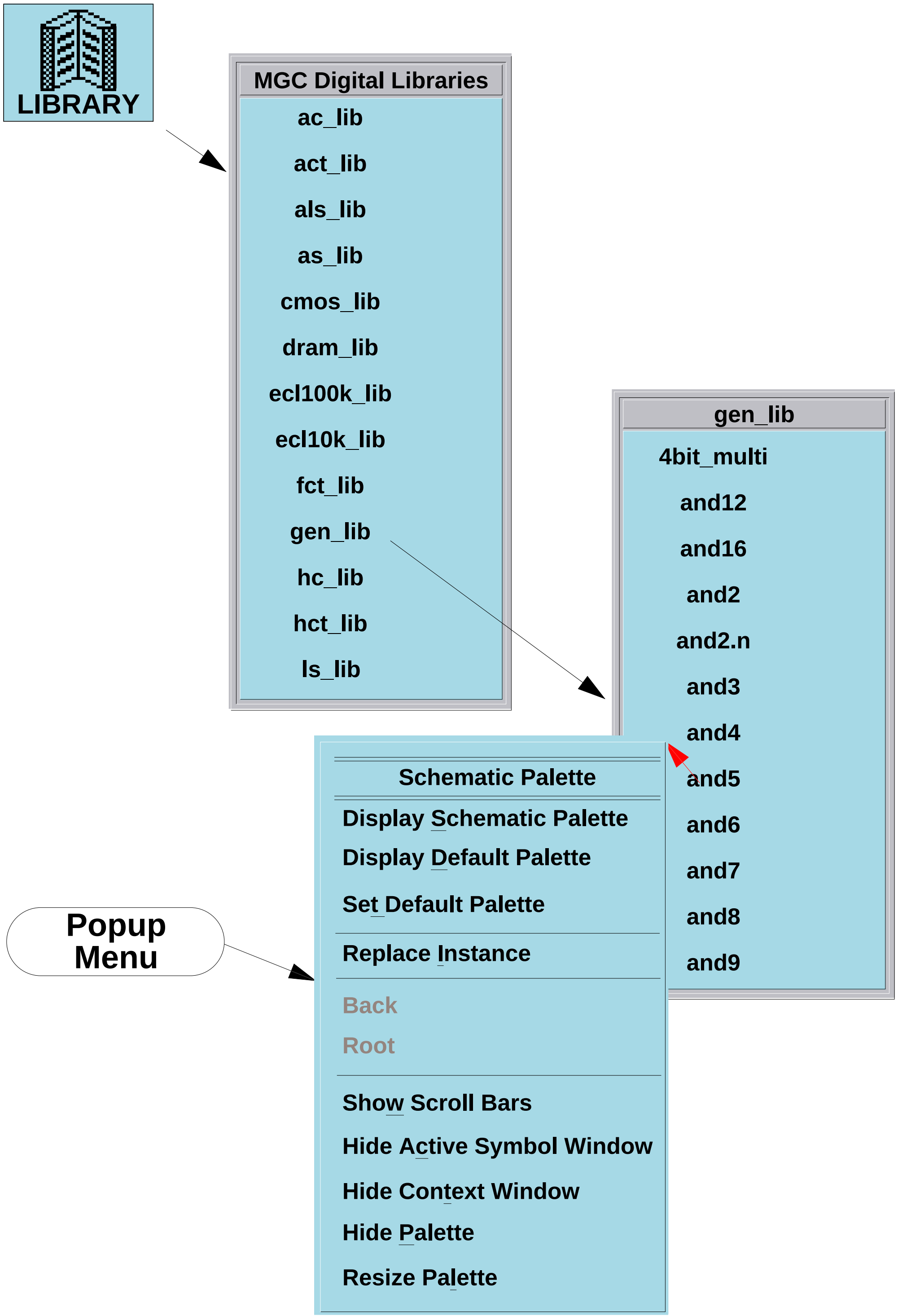
Placing an Instance on a Sheet

Component libraries contain a wide variety of software models that represent off-the-shelf electronic components available from IC vendors. You can use component libraries supplied by Mentor Graphics, third-party vendors, or you can create your own libraries.

When you place a symbol on a schematic sheet, you are really placing an *instance* of the symbol that represents that component. This is called *instantiating* a component symbol. This instance is not a copy of the component symbol; it is more like an active reflection because a direct reference to the symbol is established and maintained. Any changes made to the symbol are generally reflected by the instance on the schematic sheet, after an explicit update of the symbol is made, or the next time the sheet is opened.

There are several methods of placing component symbols on a schematic sheet. You can access component libraries through either the **schematic_add_route** palette, the **ADD** popup menu, or the **Libraries** pulldown menu. You will learn some of the more common methods to choose a component symbol in the next several pages.

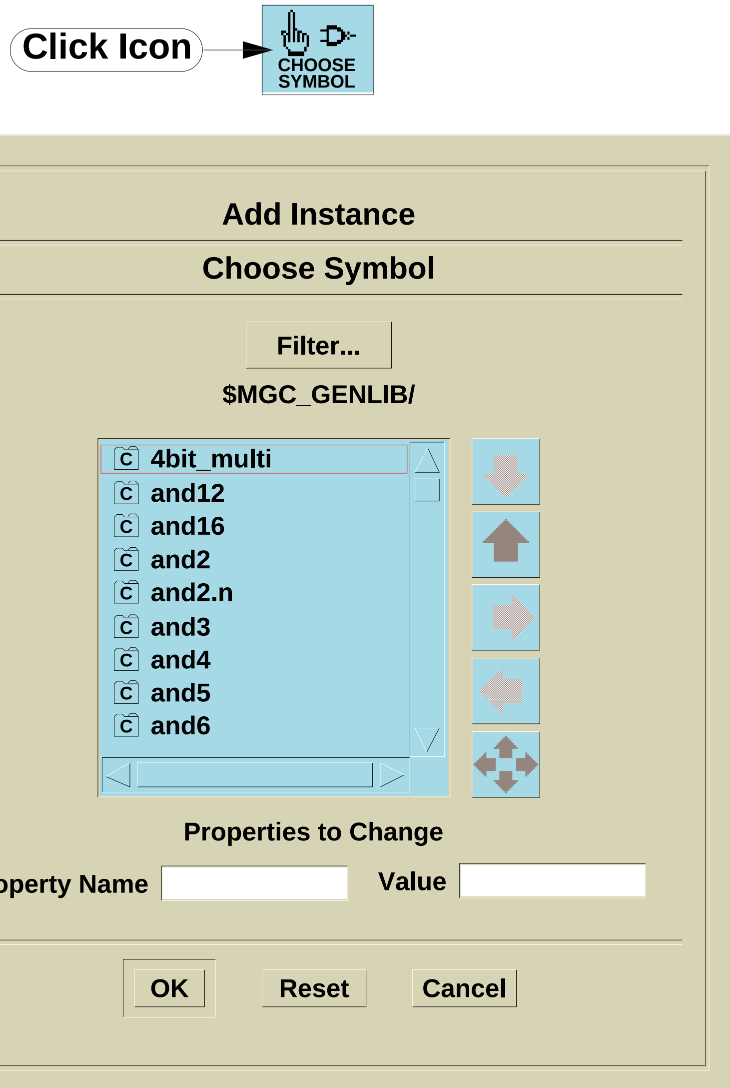
Mentor Graphics Libraries



Mentor Graphics Libraries

A variety of Mentor Graphics libraries are available for creating schematics. The most common is **gen_lib** which contains models of common logic elements without technology specific property values. Other libraries are available such as **ls_lib** which models the 7400-series ls family. Mentor Graphics libraries can be accessed through the LIBRARY icon as shown on the left. Once you are in a library, the list of components can be very long. You can use the popup menu to added scroll bars to the palette to help you scroll through the list.

Using the Choose Symbol Option



1. **Press Goto button to specify pathname**

2. **Double click the component icon to see possible symbol choices below.**

3. **Click on symbol icon, then click OK**

Using the Choose Symbol Option

When you click the **CHOOSE SYMBOL** icon from the **schematic_add_route** palette, the Choose Symbol dialog box is displayed as shown on the left.

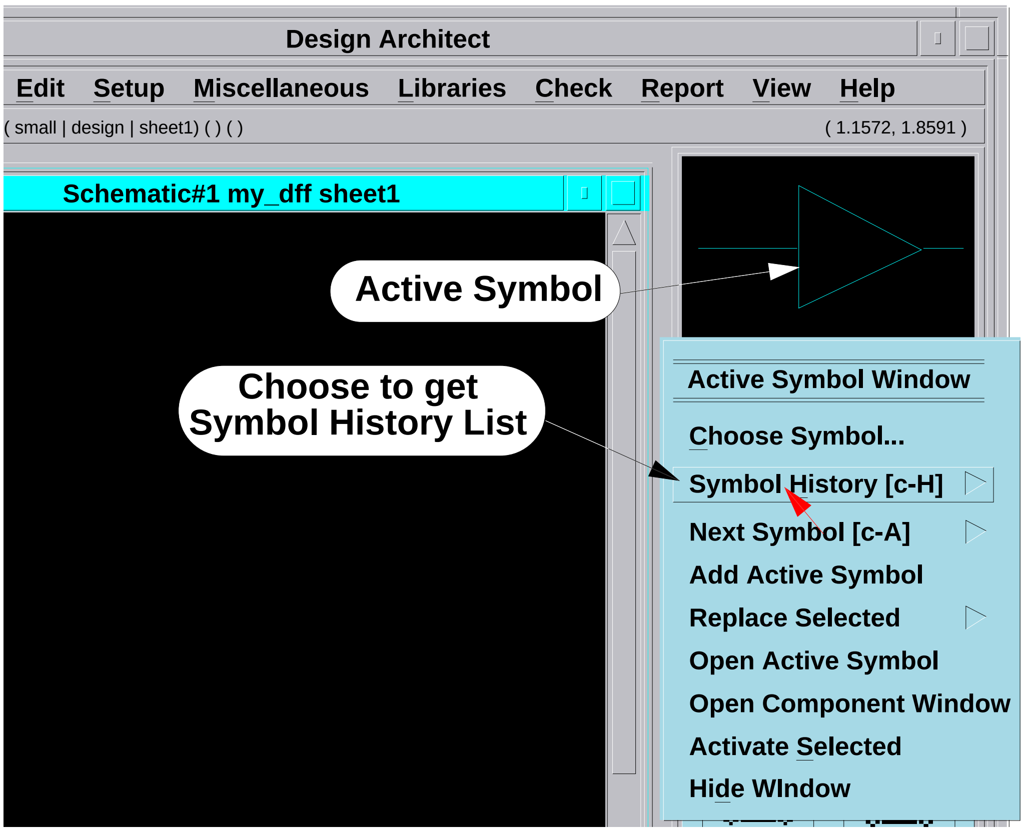
The scrolling list contains the contents of the current working directory. To move to another directory, you can click the **Goto** navigator button. This brings up another dialog box requesting a directory name. You can type in the name and click **OK**. The scrolling list now displays the contents of the specified directory.

You can select the component by clicking the Select mouse button on the component name (the icon that contains the “c”). You can also optionally select the symbol (if you want to select a symbol other than the default) by double clicking the “C” icon, then selecting the correct symbol icon and executing the form.

The **Filter** button lets you filter out certain types of objects from being displayed in the Dialog Navigator.

The Active Symbol

- **Symbol of the last instance added to sheet**
- **Displayed in the Active Symbol Window**



The Active Symbol

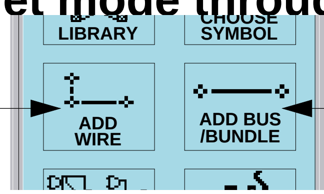
The *active symbol* is displayed in the Active Symbol window. It can be set implicitly when you place or replace an instance on a sheet or explicitly when you change the active symbol. For example, if the last instance you placed on the sheet was *$MGC_GENLIB/buf*, then **buf** is the active symbol. The Active Symbol window popup menu(shown on the left) is defined as follows:

- **Choose Symbol**. Changes the active symbol to another symbol.
- **Symbol History**. Shows an ordered history list of previously active symbols. You click on a symbol name in the list to make it active. This list also contains common elements such as portin, portout, vcc and ground to provide a quick way to get to these parts.
- **Next Symbol**. Used for components which have multiple symbol models, such as $MGC_GENLIB/rip. Clicking this item changes the view to the next symbol in the symbol list.
- **Add Active Symbol**. Adds another instance of the active symbol to the sheet. You can also click the Select mouse button in the Active Symbol window or draw an "L" stroke in the Schematic window.
- **Replace Selected**. Replaces a selected instance with the active symbol.
- **Open Component Window.** Brings up a Component Window on the Active Symbol. This window is discussed later in this module.
- **Open Active Symbol**. Brings up the Symbol Editor containing the symbol in read-only mode.
- **Activate Selected**. Makes the selected instance's symbol the active symbol.
- **Hide Window**. Removes the Active Symbol window from the Design Architect Session window. To show the Active Symbol window, use the popup menu **Palette** > **Show Active Window** or the pull down menu **MGC** > **Setup** > **Session** dialog box.

Adding Nets

- **Nets are used to connect instance pins together**
- **The term net refers to a wire or a bus**
- **A single net is called a wire**
- **A grouped set of wires is called a bus**
- **Enter the add net mode through the palette icons**

Add Wire



Add Bus

- **Or, use the Add Wire Stroke and the Add Bus Stroke**
- **You stay in the Add Net mode until you click “Cancel” on the prompt bar.**

Click to Exit

Adding Nets

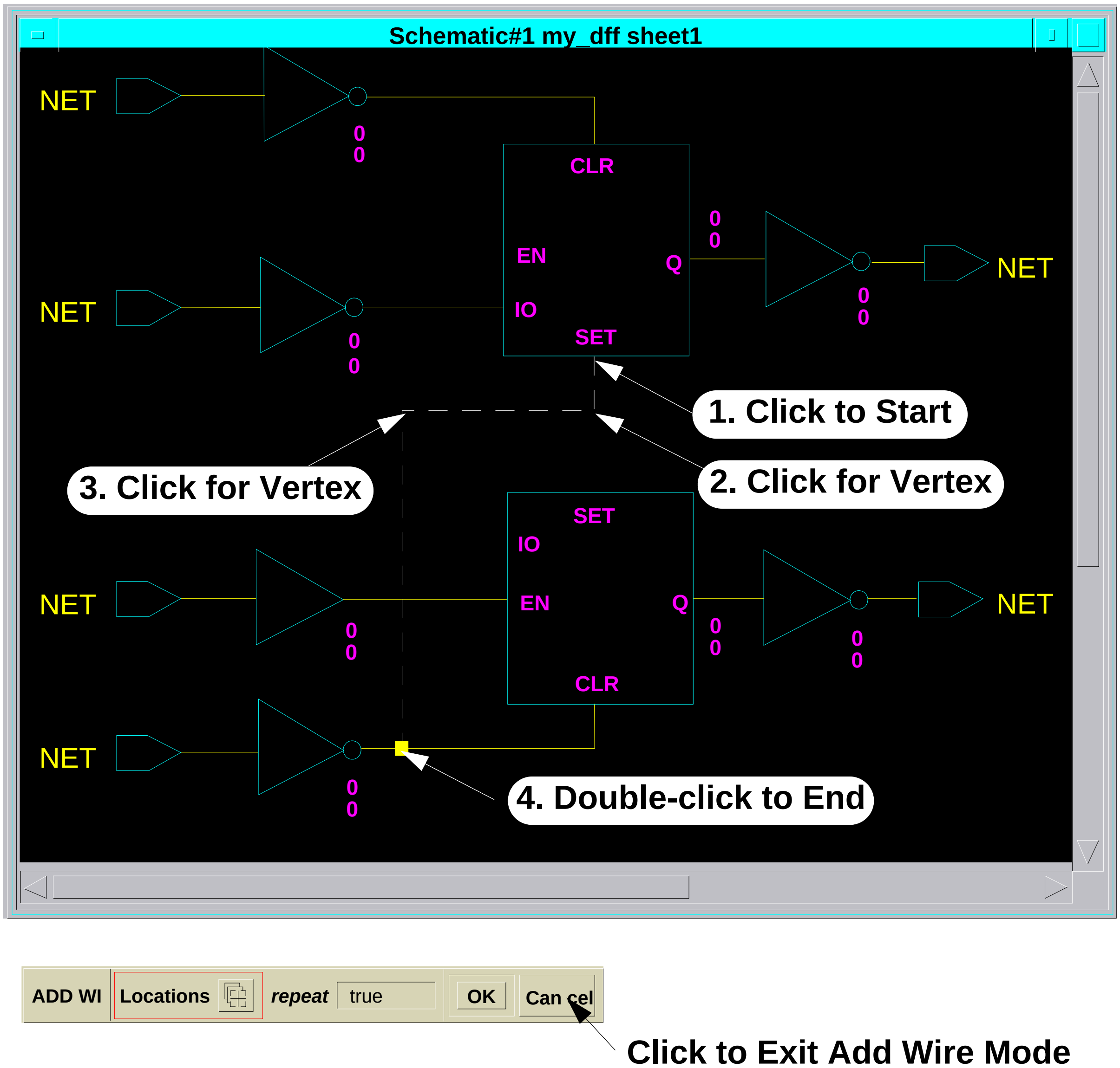
Short for network, the term “net” refers to a wiring connection between two or more instance pins. The term net refers to either a single wire or a collection of wires called a bus. A bus is sometimes called a bundle. A net may contain net vertices (corner points), net segments(the pieces between two vertices) and net junctions.

Although Design Architect assigns an internal system name to each net (a handle name like N$3) you can assign names that have more meaning. External nets, the nets connected to ports, must be named to match the pins on the associated symbol. You can also assign internal nets meaningful names to make it easier for you to identify them in downstream applications.

Two or more nets with identical names are treated as connected, even if they may not be graphically connected on the sheet. Two nets with identical names on different sheets in the same schematic are also treated as connected.

Graphically, wires are designated as single-pixel lines. Buses are designated as three-pixel lines by default, but they can also be designated as five or seven-pixel lines. The thickness of a wire or bus is only for visual appearance. The system knows whether a net is a wire or a bus by how it is named, not how it looks. Bus naming conventions will be discussed in a later module.

Net Creation Process



- **Press “Backspace” key to backup**

Nets Creation Process

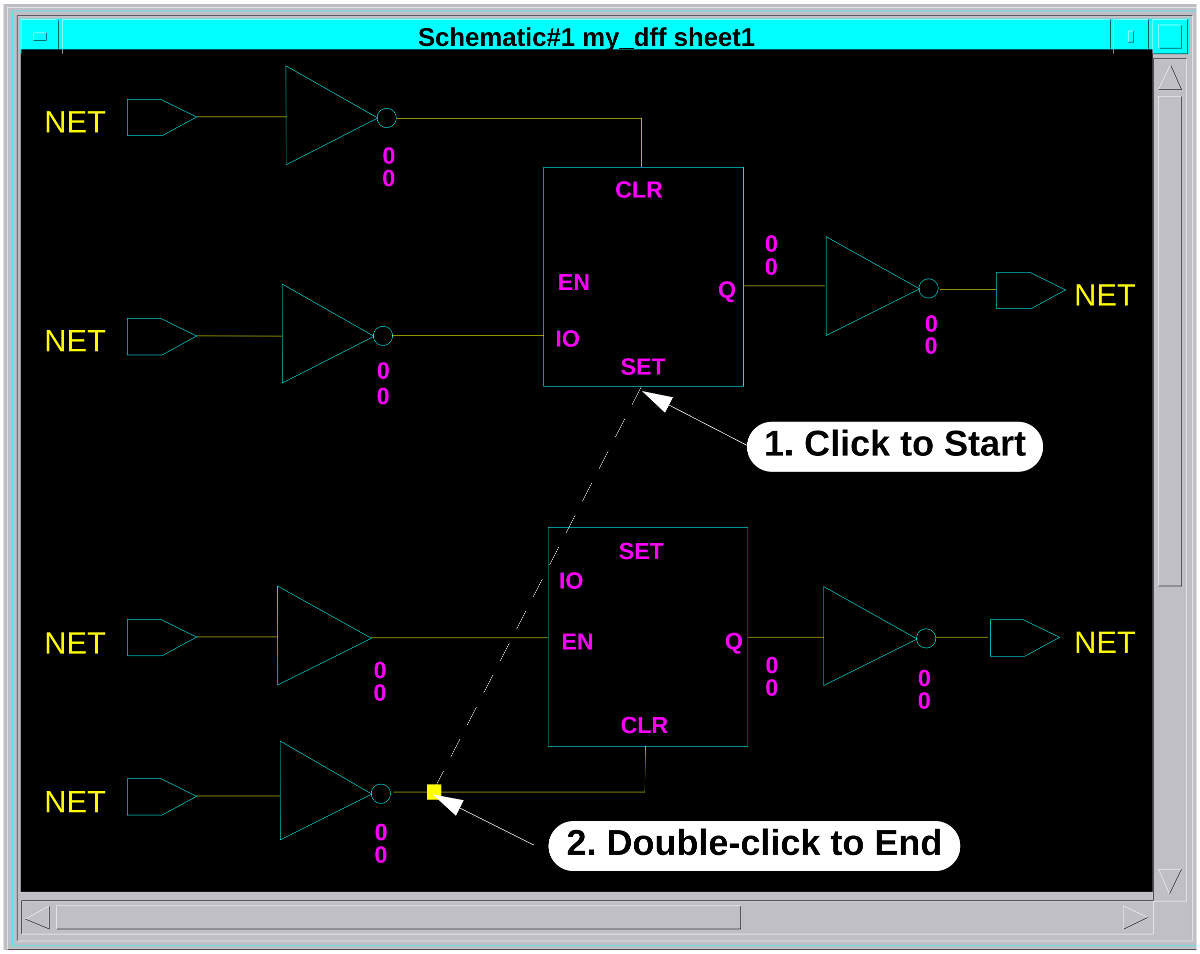
The example on the facing page shows that after the prompt bar is displayed, a net is started by clicking the Select mouse button on the starting point, usually an instance pin. Corner points (net vertices) are made by clicking once on a point and the end of the net is defined by a double click.

You can continue to add other nets by clicking the Select mouse button. You exit from “add wire” mode by clicking the **Cancel** button on the still-displayed **ADD WI**re prompt bar. The **ADD BUS** functionality works the same way.

If you are in the process of building a net, you can backtrack to a previous net vertex by pressing the BackSpace key.

When you are connecting a net to a pin, you must have the pointer within one-third of the pin spacing unit for the net to snap to the pin. If you don't, the net vertex snaps to another grid. When you check the sheet, the check software flags this as a warning (a dangling net and an unconnected pin). If you think that you missed the connection, you can zoom into the area with a  stroke to take a closer look.

Autorouting Nets



- Choose “Setup > Set Autoroute On” to autoroute nets as they are drawn

Autorouting Nets

You can automatically route nets as they are drawn, or you can manually select and route the existing nets later. The net router defines a pathway for a connected net that avoids instance extents, comment objects, and other nets, utilizing the pin snap grid as the routing grid. If net vertices are not on the grid, they are not routed.

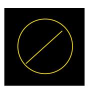
If you create a straight net between two pins, then click the ROUTE SELECTED icon, the autorouter will attempt to route the selected net. If you execute the **Setup > Set Autoroute On** pulldown menu item, the autorouter will route each net as you draw it.

Routing performance is faster if the pin snap grid is initially set to a value larger than one pin interval during the route operation, and then set back for component instantiation.

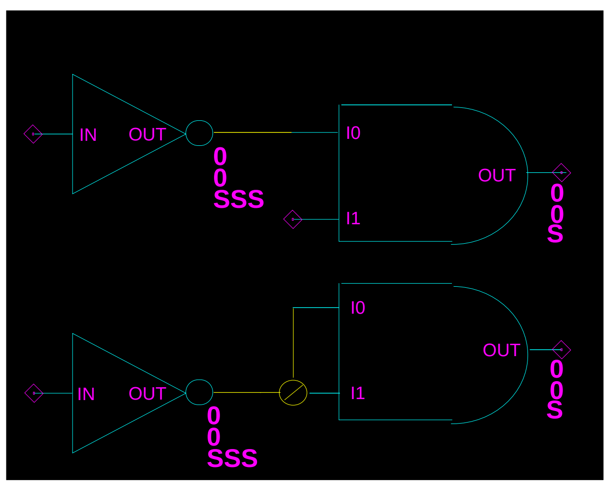
Net Connection Rules

- **Automatic connections are made when nets are created and instances are placed**

- **When Manipulating objects (such as move and copy)**
  - **Connectivity is preserved**
  - **No new connections are made at pin / vertex intersections**

- **Not-dots:** 
  - **Are non-electrical, graphical indicators showing that no connection was made**
  - **Are displayed when a net crosses a pin or vertex during an edit operation**

- **Example:**
  - **Before move:**
  - **After move:**



Net Connection Rules

During net creation, connections are created at input points with pins, net vertices, or segments. However, edit commands such as Move, Copy, Flip, Delete, Rotate, and Pivot, do not automatically make new connections. Instead, they preserve the connectivity that was established before the edit operation. If a net passes over any existing pins or vertices, no connections are made at those vertices.

This can result in cases where nets may look connected, but are not. To enable you to see where a net connection has not been made, these locations are marked with a non-electrical graphical object called a "**not-dot**." A not-dot on a vertex indicates that not all segments passing through the vertex are connected. It can be plotted, but cannot be removed without changing the graphics or connecting the overlapping vertices using one of the **Connect** commands (discussed on the next page).

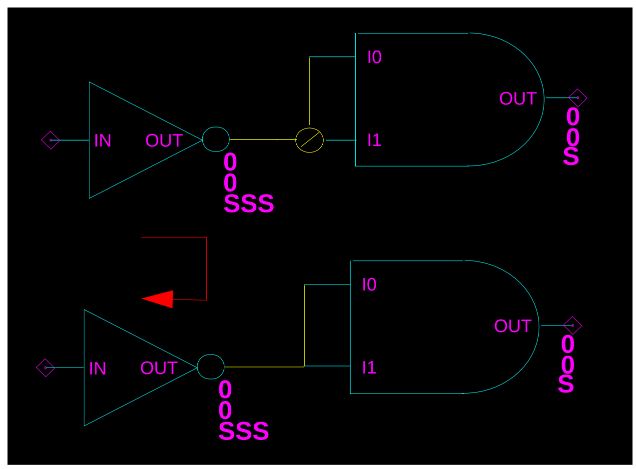
In the example on the facing page, an instance of an **inv** symbol and an instance of an **and2** symbol are connected by a net. The **inv** instance is selected and moved so that the attached net touches the second pin vertex of the **and2** instance. A not-dot appears to indicate that there is no electrical connection made at that point.

Connecting and Disconnecting Net Vertices

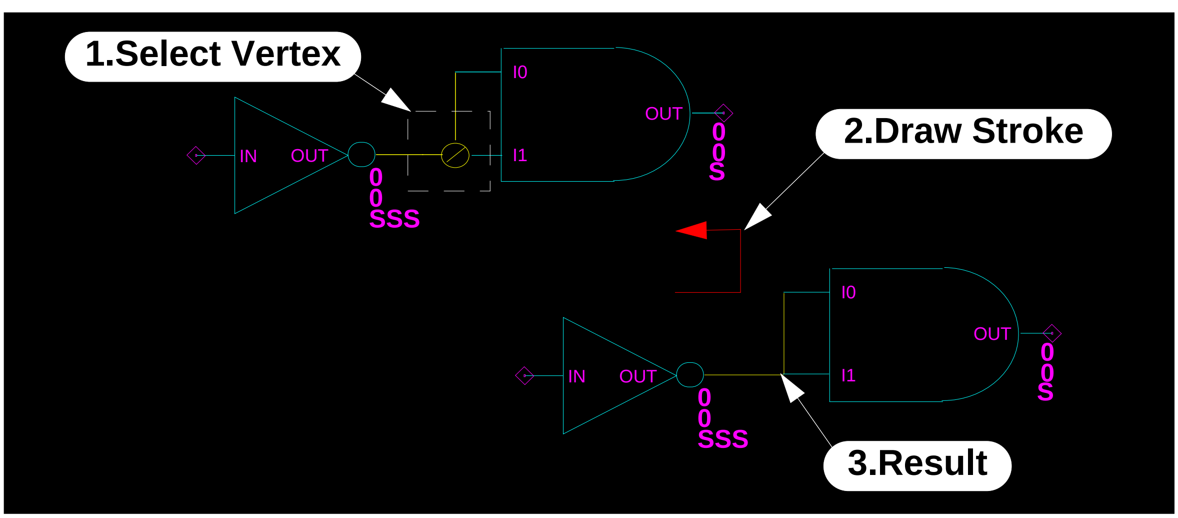
- **Forces connect or disconnect of net segments and instance pins**
- **Occurs at specified junctions and intersections**
- **Connect All:**

Before:

After:



- **Connect Selected:**



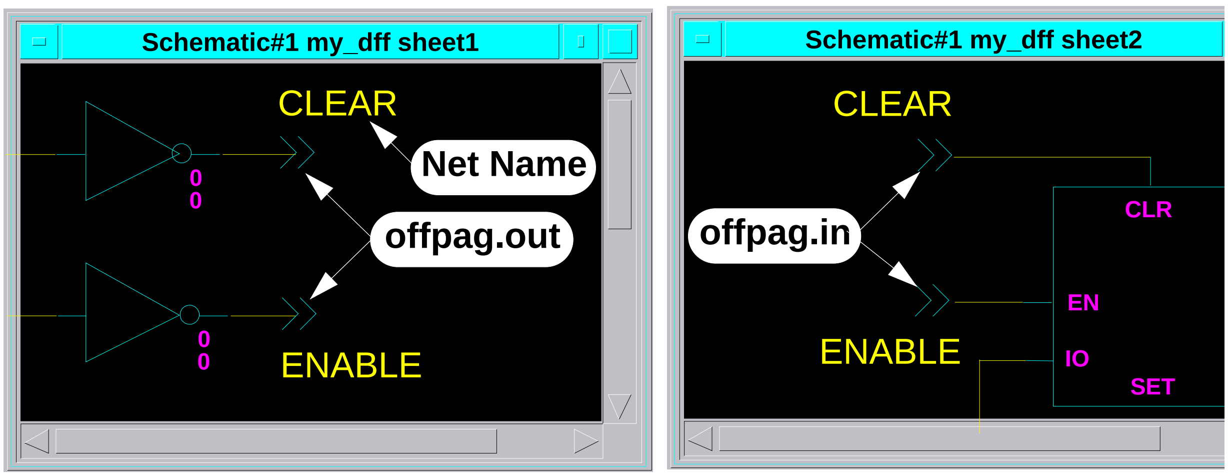
Connecting and Disconnecting Net Vertices

Clicking the CONNECT ALL icon or drawing a backward “C” stroke(top to bottom) forces all net vertices marked by not-dots to be connected. If you select one of several not-dot vertices and draw a backward “C” stroke(bottom to top), only the selected connection is forced.

Sometimes you need to disconnect the end of a net from a pin and move the net to a different location. The procedure is to execute the popup menu item **ADD > Connections** > **Disconnect Area,** draw the bounding box around the pin (with the Select mouse button, select the net vertex at the end of the net and move the vertex with a move command.

Naming Nets

- **Net Names give each net a unique identity**
- **Net Names establish connectivity across sheet boundaries**



Adding Net Names

1. **Select the nets you wish to name**
2. **Choose (popup menu) Name Nets:**

Changing Net Names

1. **Place the mouse cursor over the net name**
2. **Press Shift-F7 (Change Text Value) function key**

Naming Nets

You can easily name a net that currently has no name and you can rename a net that does have a name. Remember all net names must be unique and all buses must have a net name. In the illustration to the left, the offpage connectors are added only to provide a visual indication that the net goes off the page. The net name is the object that actually establishes the connection across the sheet borders.

You can use the **NET > Name Nets:** menu item to add new net names and to change existing net names. This menu item works with selected nets, selected vertices, and selected net names, or a combination of all three.

To quickly change an existing single net name, you can use the Shift-F7 function key. Select the name, press **Shift-F7**, fill out the form and click OK.

Selection Concepts

- **Types of Selection**
  - **Point Selection (click on object)**
  - **Area selection (press and drag bounding box)**
- **Selection Modes**
  (da_session) Setup > Set > Individual Selection Model
  - **Additive - selected object is added to the set**
  - **Individual - selecting an object deselects others (Ctrl-LMB overrides and adds to the set)**
- **Selection Sets**
  - **Objects are added to a selection set until an operation that uses the selection set is performed**
  - **One selection set per sheet is maintained**
- **Selection filter**
  - **Used with mouse pointer**
  - **Function keys do not use selection filter**

Selection Concepts

Types of Selection. You select objects so that you can manipulate them with commands such as Move and Copy. You select a single object by clicking the Select mouse button on an unselected object. This is called *point selection*. When you click the Select mouse button on an already-selected object, it becomes unselected.

You can select many objects by pressing the Select mouse button, moving the cursor to the opposite corner of the rectangular area with the mouse button, and releasing it. This is called *area selection*.

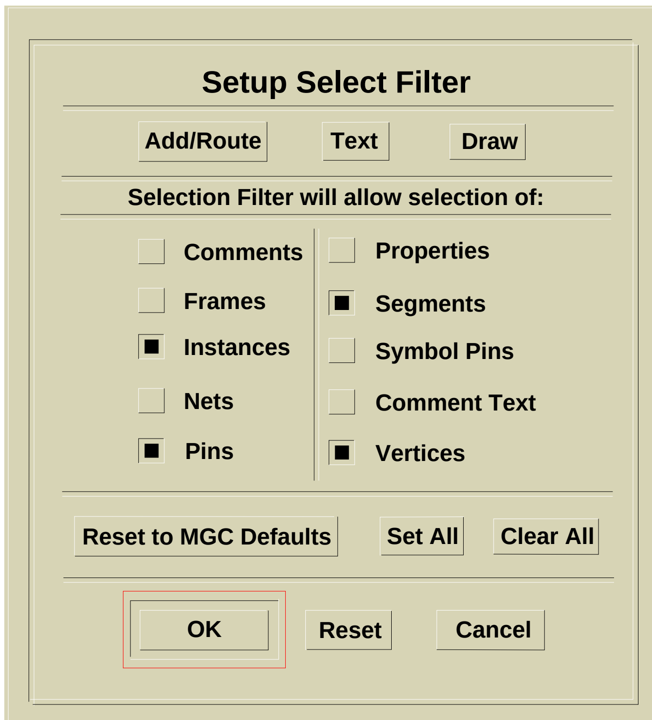
Selection Modes. The default selection mode is *additive*. Each object you select is added to the object already selected. You may switch to the *individual* selection mode by choosing **Setup > Set > Individual Selection Model** from the DA Session window. In this mode, when you click on an object, all others are unselected (unless you hold down the Ctrl Key).

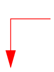
Selection Sets. As you select objects, they are added to the *selection set* (when in the additive selection mode). Operations that use the selection set are any of the following: Move, Copy, Flip, Rotate, Delete, Pivot, Change, and Report. After one of these operations, objects are still selected, but the selection set is closed and still defined. The next selection will discard that selection set to open a new selection set. Each sheet or symbol opened has its own open selection set. If you select objects in one sheet and then activate a different window on another sheet and select new objects, objects in the previous sheet remain selected.

Selection Filter. Whether or not an object is selectable is defined by the *selection filter*, which is used in point and area selection. This filter is ignored when you use the menus or the F1 (Select Area Any) or F2 (Unselect All) function keys. For example, property text is not selectable by default. If you move the cursor on a property text and click the Select mouse button, the property text remains unselected. However, if you press the F1 function key, the property text becomes selected.

Select Filter

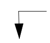
- **Specifies object types for selection**
- **Used only with mouse point selection**



- **Use**  **stroke to bring up form**
- **Or, click the** SET SELECT FILTER **button on the palette**

Select Filter

The select filter controls what objects are currently selectable from the mouse pointer. It is not use with the F1 (Select Area Any) function key.When you change the type of objects that are selectable through the **Setup Select Filter** dialog box(as shown on the left), the result is in effect for every existing schematic window and any subsequent schematic windows that are opened in that editor.

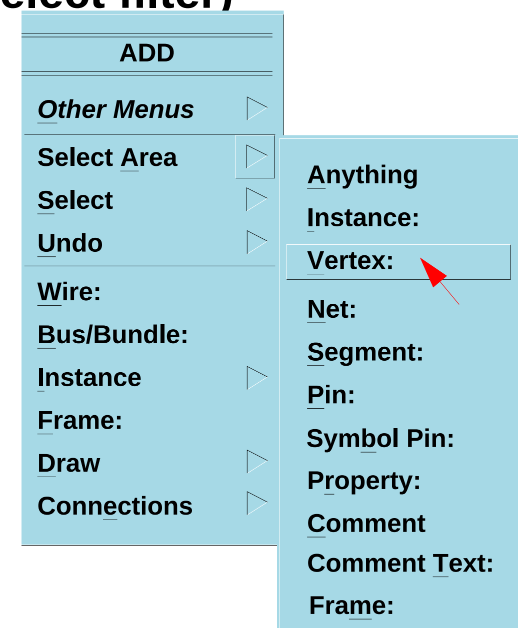
When you choose either the **Set Select Filter** button in the palette, or draw the  stroke, a dialog box is displayed as shown on the left. The three buttons that are displayed at the top, "**Add/Route**", "**Text**", and "**Draw**" correspond with the three Schematic palette buttons. When you are adding instances and creating nets, choose the "Add/Route" button to set the appropriate selection filter. When you are adding and manipulating property or comment text, choose the "Text" button. And when you are creating symbol graphics, choose the "Draw" button. Or, you can click on the individual object types that you want to be selectable. The "Reset to MGC Defaults" button resets the options to the Mentor Graphics-specified defaults.

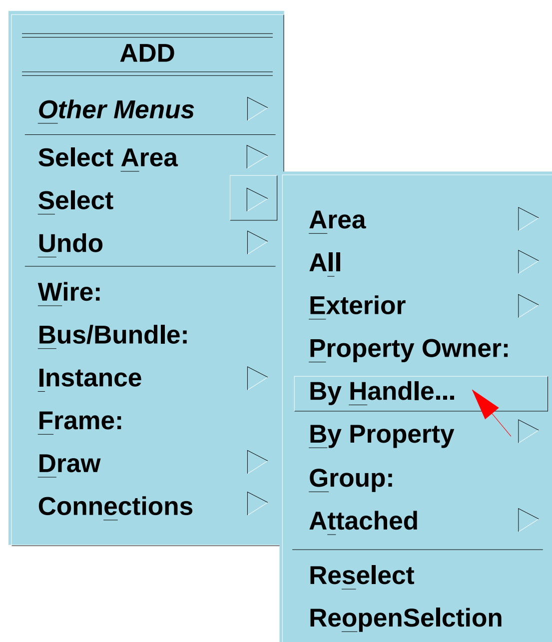
You can override the select filter by choosing menus from the **Select > Area**, **Select > All**, or **Select > Exterior** menus. However, you only override the filter for that one time.

The filter remains in effect for subsequent select actions for the current Design Architect session unless you override them for a single operation or change the defaults in the Setup Filter dialog box. You can add the appropriate select filter function to a startup file that can be executed any time a sheet or symbol window is opened, or you can include this function in a startup file that is executed when Design Architect is invoked.

Using the Select Popup Menus

- **Good for one selection operation (then returns to select filter)**





Using the Select Popup Menus

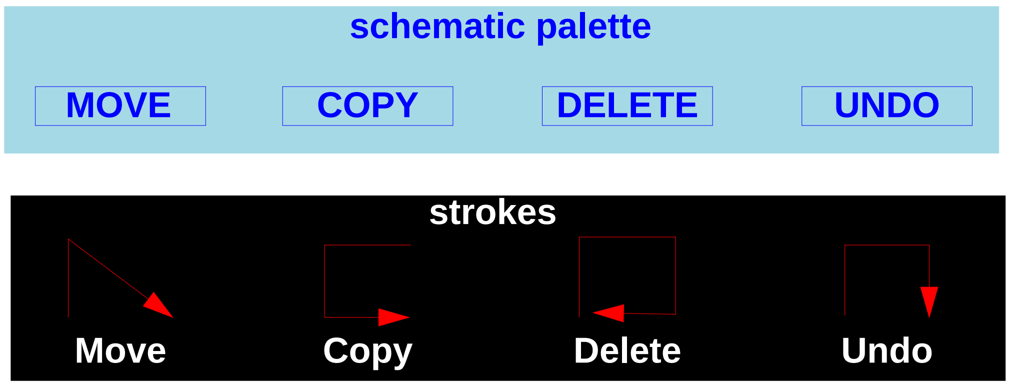
Objects can be selected in a variety of ways using the **popup menus.**

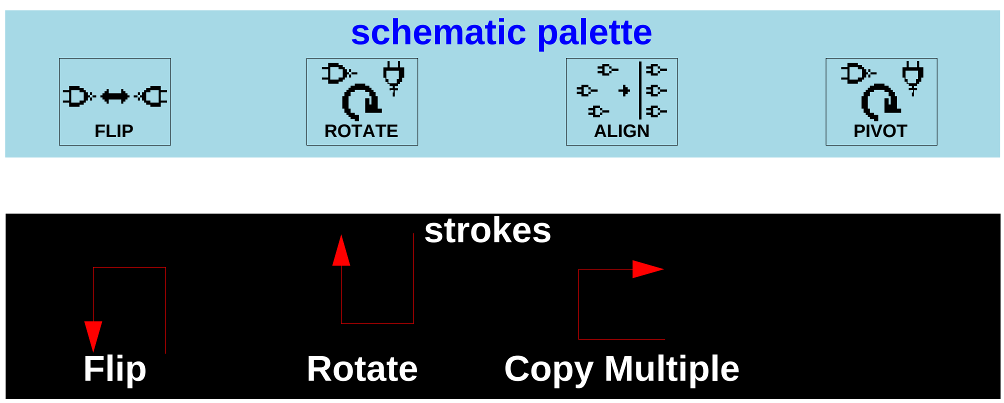
Select > All, **Select > Area**, and **Select > Exterior** cascading menus are found under all popup menus. If you choose anything other than Filtered or Anything, the selection of that object overrides the selection filter for that one time only.

For example, assume that the selection filter prevents property text from being selected, and that nothing is currently selected in the window. If you choose the **ADD > Select > All > Filtered** menu item, none of the property text is selected. If you choose the **ADD > Select > All > Anything** menu item, all of the property text is selected, in addition to everything else that is in the window. This may not be what you want. However, if you choose **ADD > Select > All > Property**, all of the property text is selected in the window, and all other objects remain unselected.

The **Select > Exterior** menu selects objects outside a specified rectangular region.

Manipulating Objects





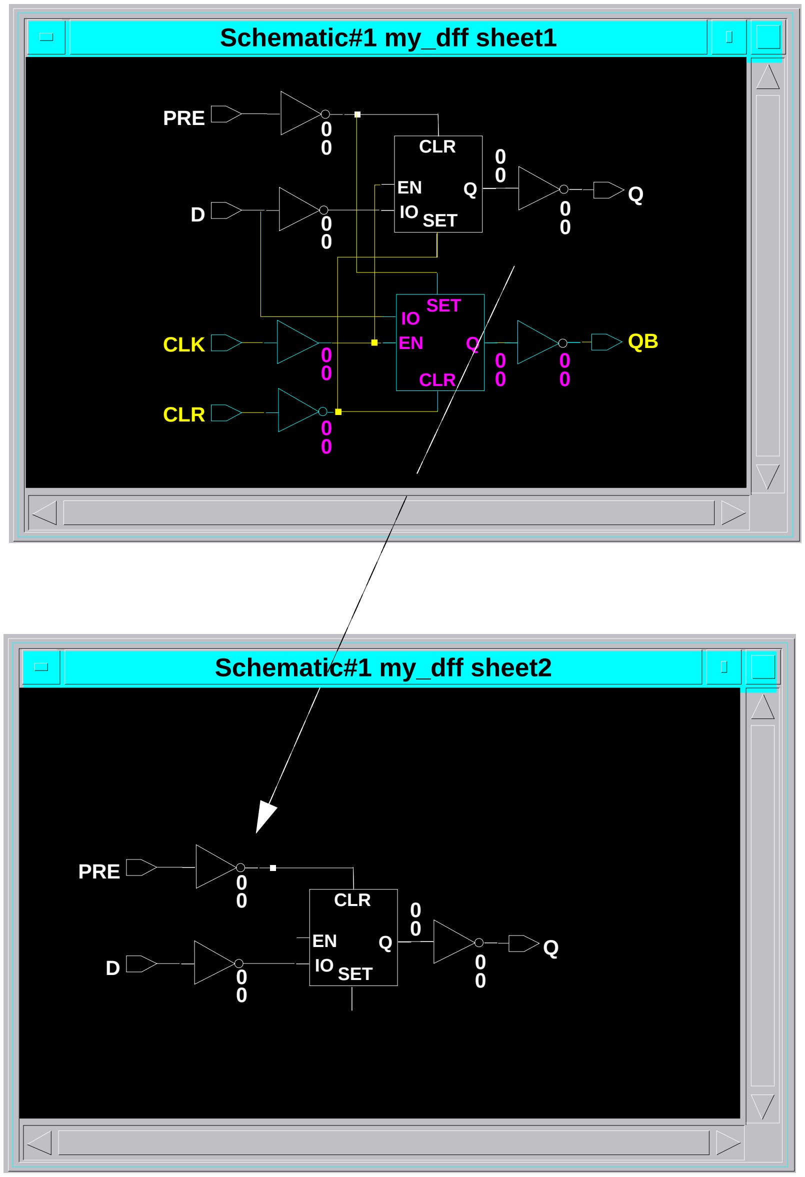
- **Select objects, then click icon or draw stroke**
- **Net connectivity is preserved**

Manipulating Objects

After selecting an object(s), you can manipulate it (them) by clicking the COPY, MOVE, DELETE, FLIP, ROTATE, ALIGN and PIVOT palette items as shown on the left. You can manipulate the objects a little faster by using the equivalent stroke as shown on the left.

These actions preserve the net connectivity that existed before the edit. For example, if a net vertex is placed over any existing pin or vertex, no connections are made at those vertices. However, not-dots appear at those vertices as a warning that what might appear to be a connection, is not.

Interwindow Copy and Move



1. Select Objects

2. Execute Copy or Move

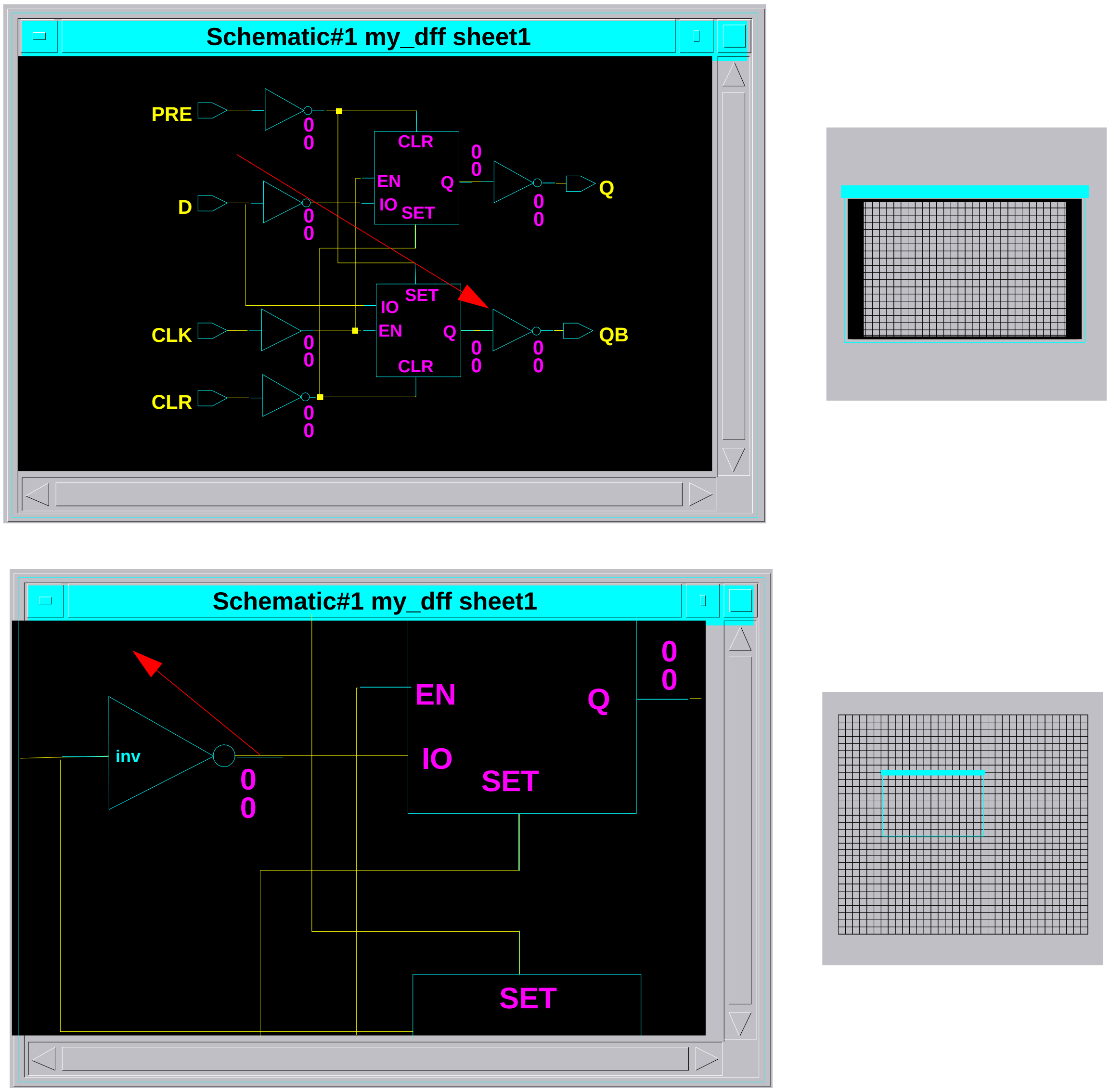
3. Move cursor to second window

4. Click the Select Button

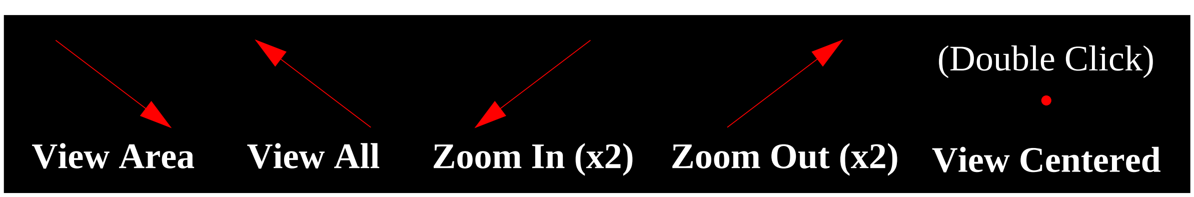
Interwindow Copy and Move

Moving and Copying schematic objects from one sheet to another or from one schematic to another is quick and easy. The procedure is to open a new window on a second sheet or second schematic. You select the objects you want to move or copy, execute the MOVE or COPY command, then move the mouse cursor to the second window and click.

The Context Window



Viewing Strokes



The Context Window

The Context window is displayed in the lower-right corner of the DA Session window. The display of the Context window consists of two rectangles. The solid patterned rectangle represents the contents of the sheet in the active window, and the window-shaped hollow rectangle represents the viewing area in the active window. The two rectangles in the Context window are moved around to remain in proportion with the active window's border and the extent of the schematic in the window.

View functionality is available through the strokes shown on the opposite page. The top illustration shows the results after a View All action is performed. The Schematic Editor windows displays the entire contents of the sheet, and the Context window shows that the viewing area is the entire sheet.

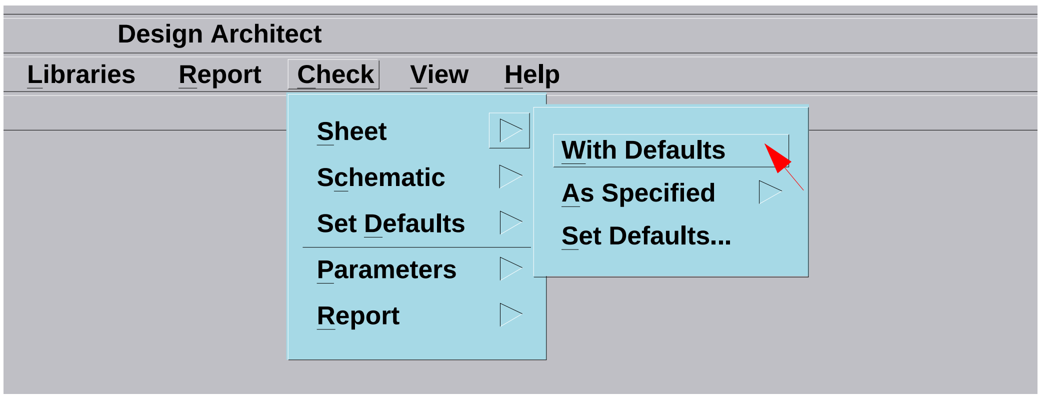
The second illustration shows the results after either a View Area or Zoom In action. The Schematic Editor window displays the partial view of the sheet and the Context window shows the relationship of the viewing area with respect to the entire sheet.

The viewing strokes can be done in either the Schematic Window or the Context Window itself.

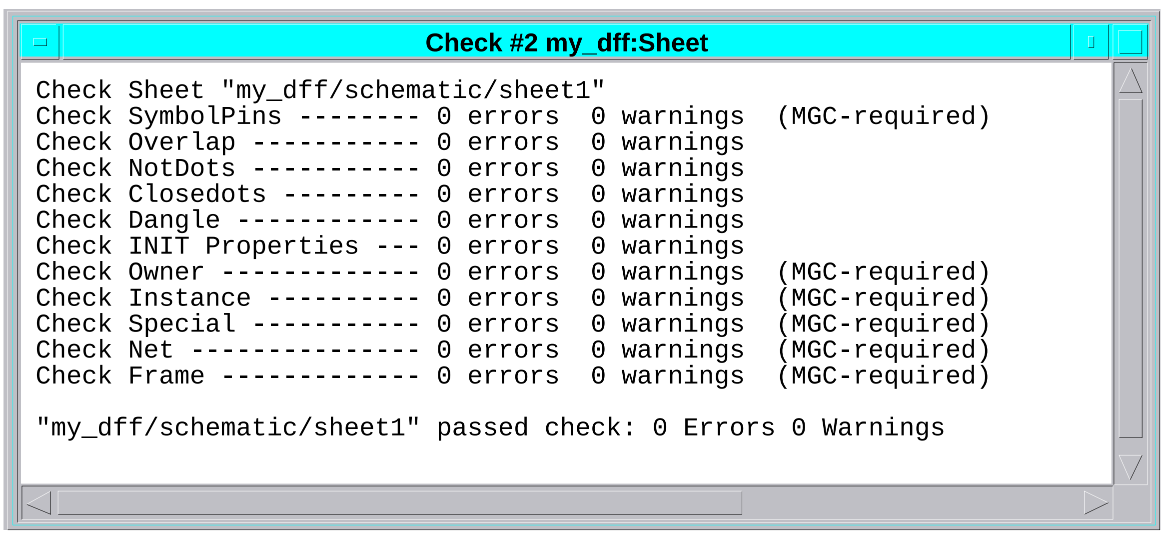
If you use the Context window to identify the viewing area and are concerned about “screen real estate”, you can hide the scroll bars associated with the active window by typing the **$hide_scrolls()** function in the popup command line.

Checking the Sheet

- **Sheets must pass required set of checks:**



- **Displays check status window:**



Checking the Sheet

To ensure that you produce a valid, workable circuit, a full set of checks must be passed. Many downstream Mentor Graphics applications require schematic sheets to pass a set of checks. In addition to the required checks, Design Architect includes a set of optional checks that can be incorporated into your design check and validation process.

Designs are not complete until they have passed the minimum required checks. If a sheet does not pass the required checks before a downstream application is invoked, the downstream application typically issues a warning message.

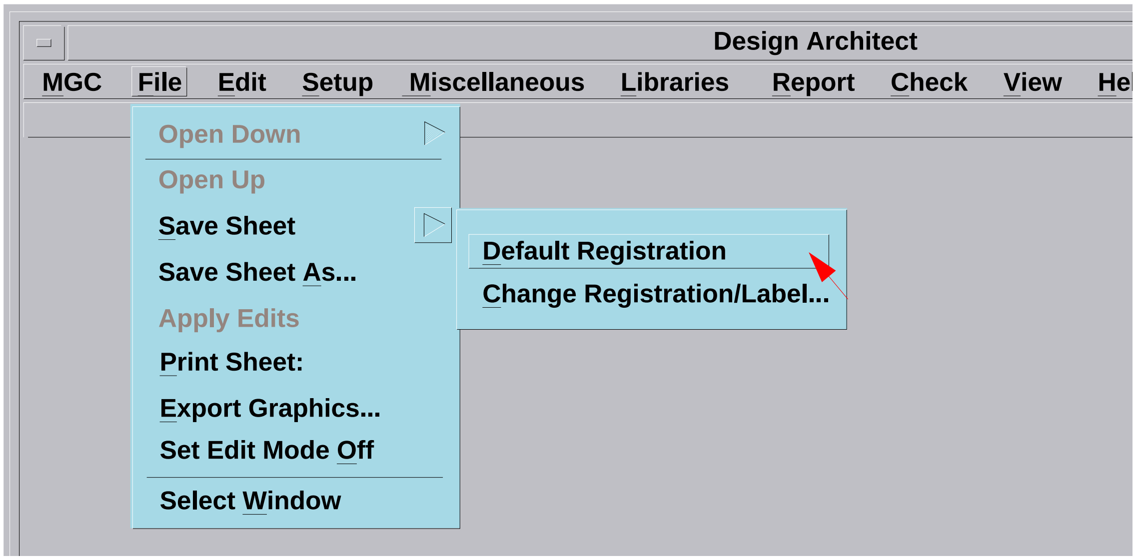
There are three levels of checking: (1) report all errors and warnings, (2) report only errors, and (3) do not check. By default, all required checks are set to the errors and warnings check level. Optional checks are set to a nocheck level.

You can specify sheet checks and schematic checks. Sheet checks validate only the contents of the sheet in the active window. Schematic checks actually validate all of the sheets of the schematic. You can check your sheet using the default check levels by selecting **Check > Sheet > With Defaults** from the pulldown menu bar. Generally, the report generated from this check provides information on instances, frames, nets, any symbol pins left on the sheet, or special instances (such as: bus rippers, net connectors, globals, and offpage connectors). An example is displayed on the facing page.

By default, check messages are displayed in a check status window that is displayed at the end of the check, and in the Transcript window where they are written during the check process.

Saving the Sheet

- **Sheets must be saved to disk:**

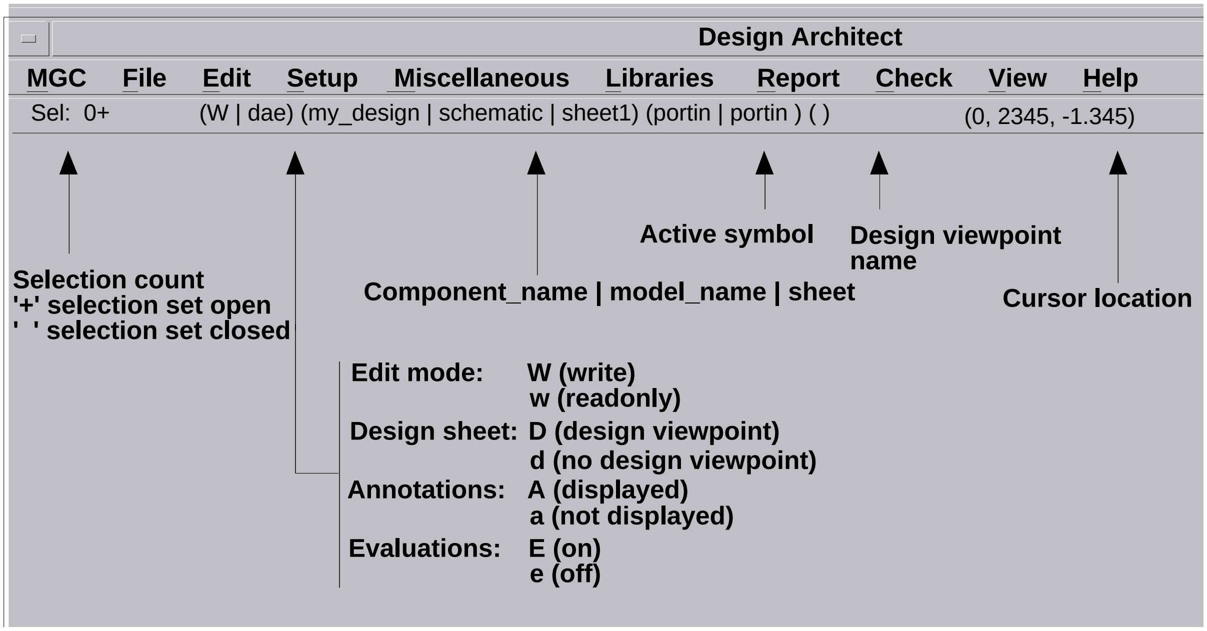


Saving a Sheet

The results of your schematic work are only maintained in dynamic memory until you save to disk with a **File > Save Sheet** command. The data is saved to disk and the schematic is registered to the component interface. The registration process includes running a validation check to see if all the ports on the schematic match the pins on the symbol. If the ports match the pins, the schematic is marked "Valid". If there is not a one-to-one match, the schematic is marked "Not-Valid". Some downstream tools care if a schematic is marked Not Valid, and some tools don't. The action the tool takes when it finds a "Not Valid" schematic depends on the tool.

Regardless of the results of the validation check, the schematic is entered into the Model List in the Component Interface Table. This is called "registration".

Schematic Editor Window Status Line

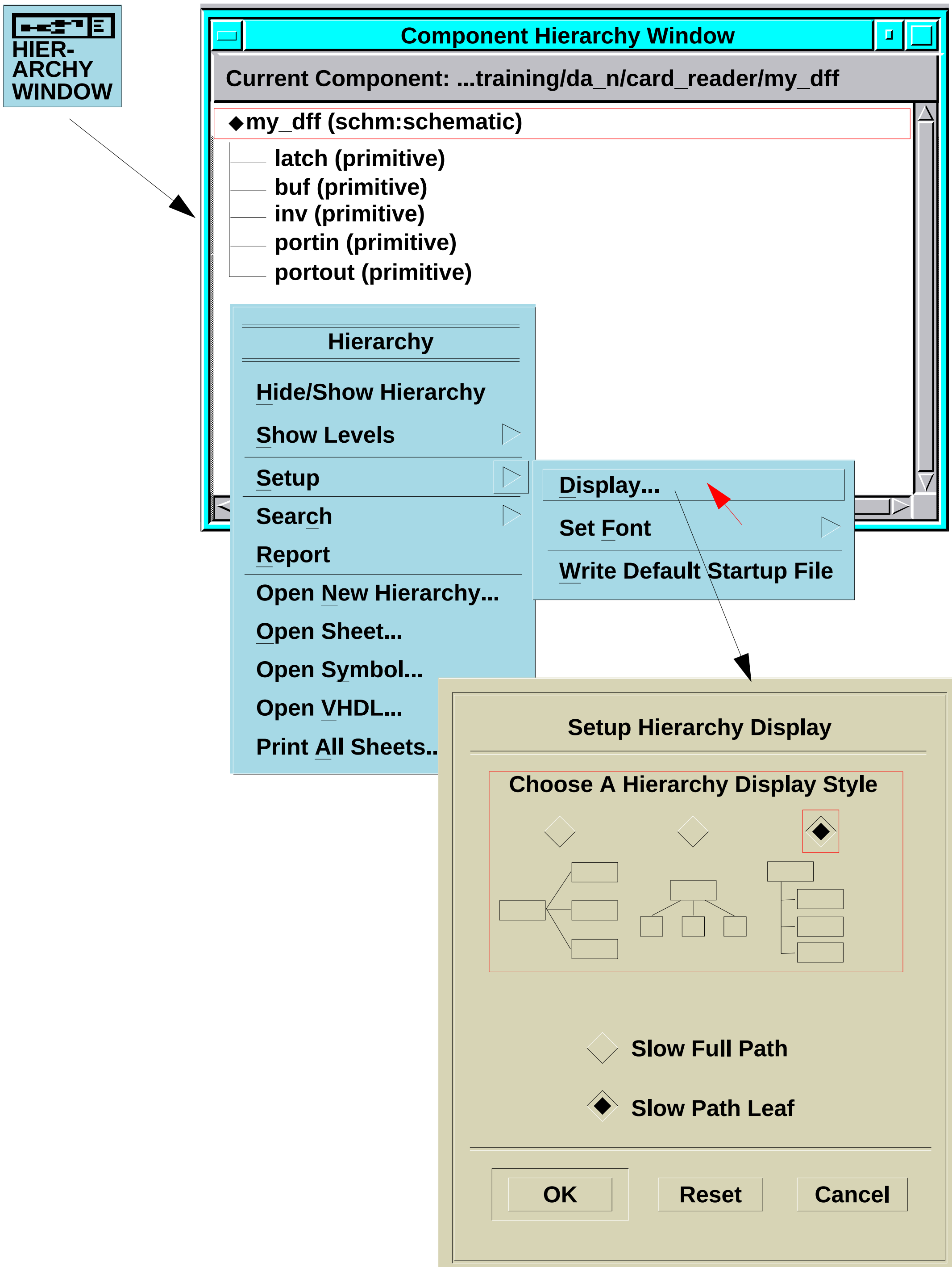


Schematic Editor Window Status Line

Now that you understand many of the basic terms and concepts of schematic capture, it is appropriate to look at the Schematic Editor status line. The status line appears below the menu bar and contains the following information:

- **Sel: (selection count)** tells you how many objects are currently selected; **+** means that the selection set is still open; closed " " means that an edit operation has been performed on the set and the set is closed.
- The sheet is either in edit mode **W** or read-only mode **w**.
- The sheet was opened in the context of a design viewpoint **D** or not **d**. (Opening on a viewpoint will be covered in a later module.)
- If the sheet has been opened in the context of a design viewpoint, either back-annotated property values are visible **A** or hidden **a**.
- If the sheet has been opened in the context of a design viewpoint, expressions are either evaluated **E** or not **e**.
- The component leaf name, the schematic name, and sheet name are displayed next.
- Active symbol information includes the leaf name of the component associated with the active symbol and the symbol name.
- The design viewpoint name is displayed after the active symbol name.
- All the way to the right is the current cursor location in user units. The point of origin is in the center of the sheet when the new blank sheet is opened for the first time.

Using the Component Hierarchy Window



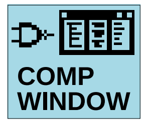
Using the Hierarchy Window

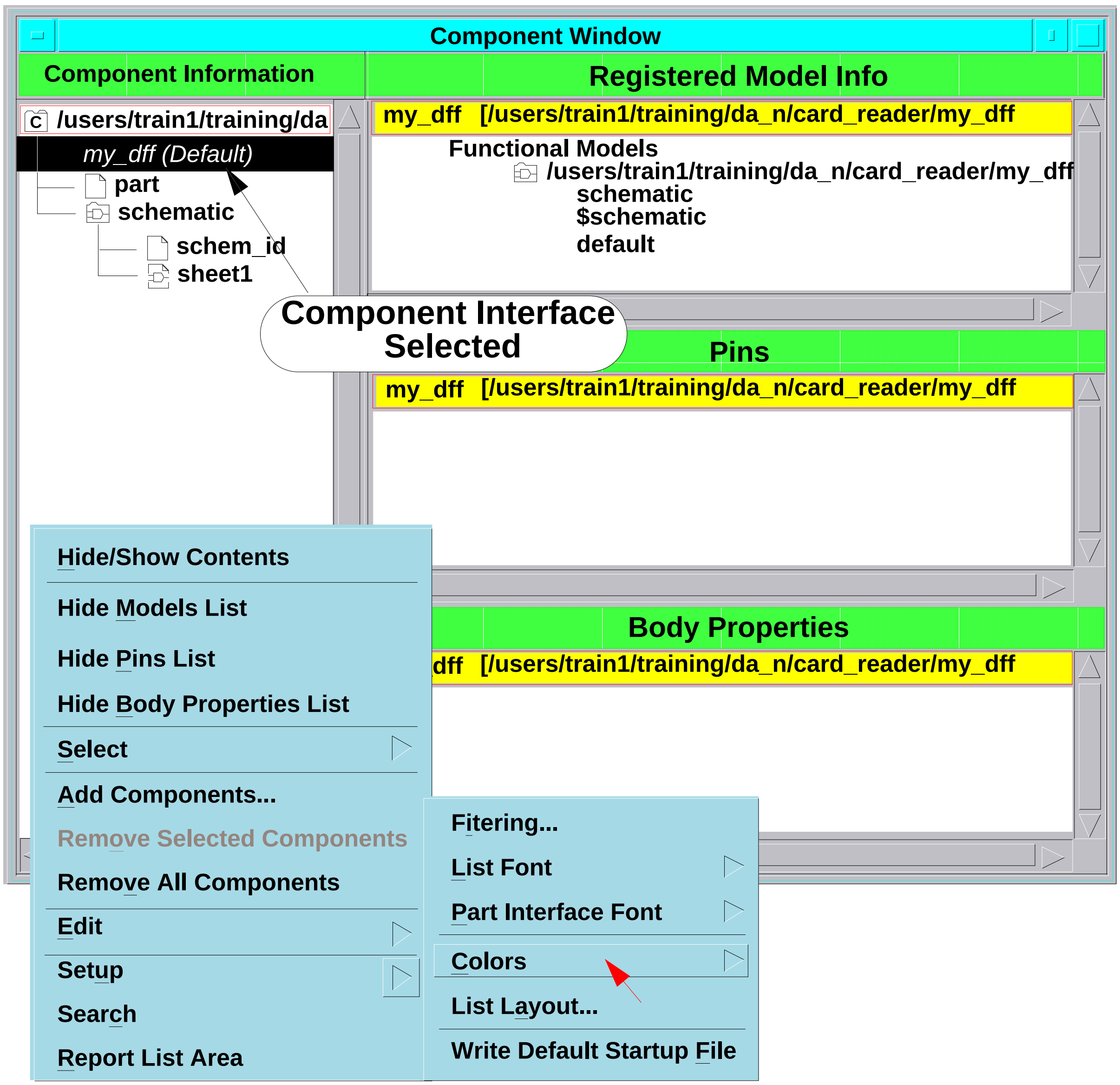
Clicking the HIERARCHY WINDOW icon in the DA Session palette brings up a Hierarchy Window on the component you specify. The illustration to the left shows a typical window. The hierarchy structure displayed can be set to three different modes through the use of the popup menu **Setup > Display...**

Double clicking on any component labeled "**schematic**" reveals the hierarchy structure underneath. Any component labeled "**primitive**" means that the component has no schematic.

The popup menu allows you to also select a component, then generate a report on its characteristics or open the Schematic Editor or Symbol Editor on it.

Using the Component Window





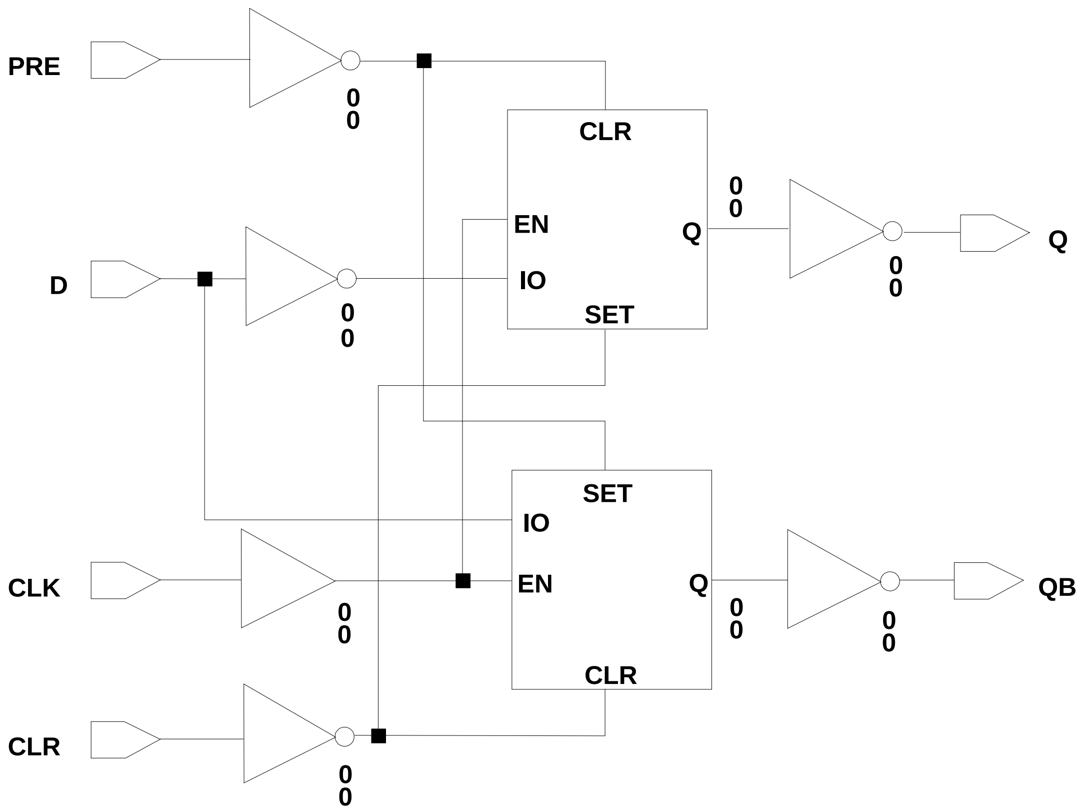
Using the Component Window

Clicking the COMP WINDOW icon in the DA Session palette brings up a Component Window on a component you specify. The Component window is a graphical view into the internal structure of the component and the Component Interface table. This window does many of the tasks that were previously accomplished by using a shell-level utility called the Component Interface Browser(CIB).

The Component Window can have up to four subwindows. The Component Information subwindow shows the component structure. Double clicking on the component icon reveals the component structure underneath and the name of the Component Interface table (italic font). In the illustration to the left, *my_dff* is the name of the Component Interface and it is marked as the default interface table. (Later you will learn that a component can have more than one table.)

When you select the Component Interface name(as shown on the left) and press the right mouse button, you can choose menu items that display information about the Model List, the Pin List, and the Body Properties List. In the illustration, the component my_dff only has a schematic model. Because there is no symbol, the Pin List is empty and the Body Property List is empty. When you double click on the schematic icon in the Model List, the labels for the schematic are listed underneath. Later you will learn that you can reference the schematic by specifying one of these labels as the value of the MODEL property on the symbol. You will also be shown how to add your own custom labels.

Lab Overview



Lab Exercises

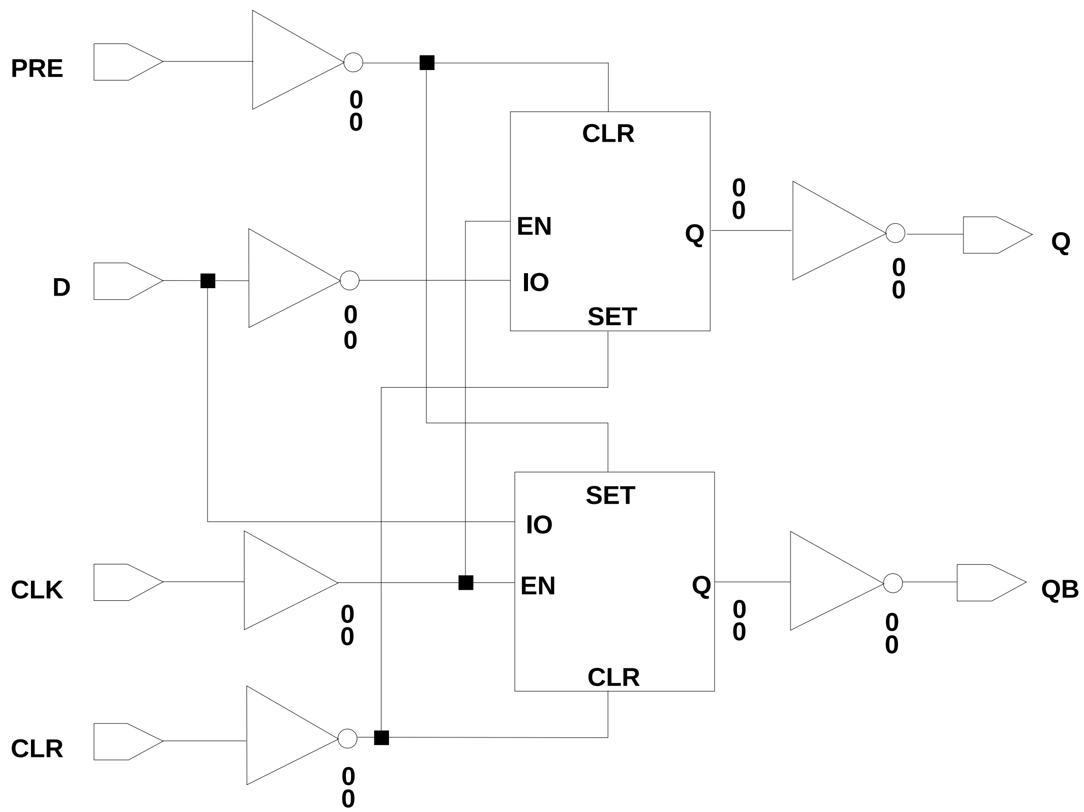
In the following exercises, you will:

- Invoke Design Architect, open a new schematic sheet and create the schematic shown on the left.
- Perform an exercise that will help you to understand and use net connection rules.
- Learn how to change the selection filter through the use of strokes.
- Browse a design hierarchy using the Component Hierarchy WIndow
- Browse a component structure and component interface table using the Component Window.

Exercise 1: Creating a Schematic

Introduction

In this exercise, you will create a new component structure and schematic for a D-type flip-flop. The schematic is shown below.



my_dff Schematic Diagram

Invoke Design Manager and Set the Working Directory

1. Log into your workstation if you haven't already done so.
2. Invoke Design Manager from a shell.
   $ **$MGC_HOME/bin/dmgr**
3. Click the Maximize button to fill the screen with Design Manager.

(Note: some window environments may have a menu choice that performs this function.)

4. Set the working directory to the the following path: *<home_directory>***/training/da_n/card_reader**.

   MGC > Location Map > Set Working Directory:

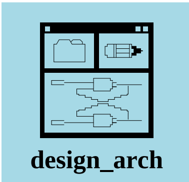
   Enter the above pathname and click **OK**.

Invoke Design Architect

1. Activate the Design Manager Tools window and invoke Design Architect.

   a. Click on the "Tools" window. The border color turns blue.

   b. Double-click on the **design_arch** icon
   .

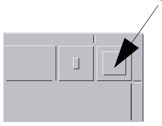


(If you don't see this icon, press the Right mouse button and execute **Update Window**. Use the left scroll bar, if necessary, to view the icon.)

After you double click the icon, the message "**Invoking Design Architect...**" appears in the message window below.

   c. Click the Maximize button to fill the screen with Design Architect.

Maximize



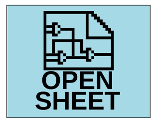
   d. Verify the setting of the current working directory.

   MGC > Location Map > Set Working Directory:

It should read <*home_directory*>**/training/da_n/card_reader**. If for some reason it doesn't, set it to this value and click **OK**; otherwise click **Cancel**.

Create a New my_dff Component Structure and Open a Sheet

1. Click the **OPEN SHEET** icon.



   The Open Sheet dialog box is displayed in the DA session area.

2. After **Component Name:** enter
   <home_directory>**/training/da_n/card_reader/my_dff**

   (Notice that the first sheet will be named **sheet1** by default.)

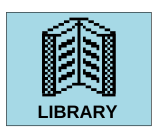
3. Press the **Return** key (or click **OK**).

   Design Architect creates a new component structure called “my_dff” and opens a Schematic Editor Window on a new (blank) schematic sheet.

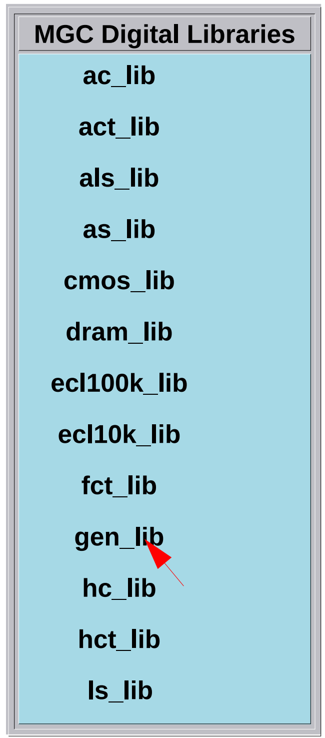
4. Click the Schematic Editor Window Maximize button.

Add an Instance of the Latch Symbol to the Sheet

1. Click the LIBRARY icon.



A list of Mentor Graphics libraries is displayed.



2. Click **gen_lib**

   A list of **gen_lib** components is displayed

3. If the scroll bars on the right side of the palettes are not displayed, press the Right mouse button and execute **Show Scroll Bars.**

4. Place the cursor on the lower arrowhead (beneath the right scroll bar). Press and hold the Left mouse button until the **latch** component scrolls into view.

5. Click on **latch**

   You will see the **latch** symbol appear in the Active Component Window.

6. Move the cursor into the Schematic Window and you will see a white image of the **latch** symbol following the cursor. Position the instance in the middle of the window and click the Left mouse button.

7. Unselect the instance by pressing the Middle mouse button and drawing a “U” stroke . Release the button.

   A red “U” is displayed, then the instance border color turns from white to blue. The instance is now unselected.

8. Draw a stroke to zoom out a level.

Add Another latch Instance to the Sheet and Flip It

1. Draw an “L” stroke .

   Again, you will see a white image of the **latch** symbol (the active symbol) follow the cursor.

2. Before placing the instance, move the cursor to the top of the window and double click the Middle mouse button.

   This point on the sheet moves to the center of the window.

3. Move the cursor so that the bottom pin of the second instance is five grid locations above the top pin of the first instance. Click the Left mouse button.

4. Flip the instance vertically by pressing the Right mouse button and from the popup menu choose:

   Rotate/Flip > **Flip** > **Vertical**

   The top instance of the **latch** is now flipped vertically.

5. Draw a “U” stroke to unselect the instance.

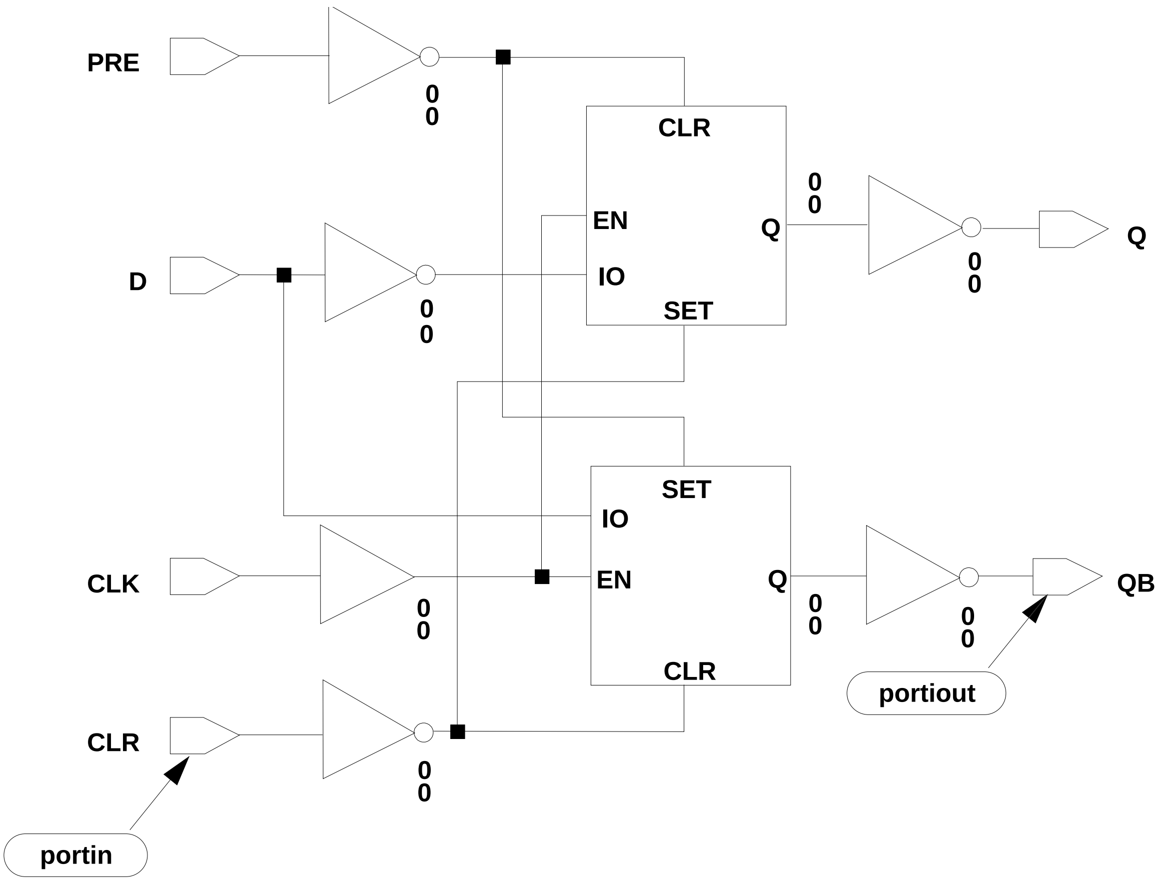
6. Draw a the stroke to include all placed objects in the viewing area.

Add the buf Instance to the Sheet

1. Move the scroll bar to the top of the **gen_lib** palette, then click **buf.**

2. Place the **buf** instance on the sheet as shown in the diagram.

3. Unselect the instance with a “U” stroke.



my_dff Schematic Diagram

Add the Remaining inv Instances to the Sheet

Add the remaining **inv, portin** and **portout** instances to the sheet as shown in the schematic diagram. Zoom out with a stroke if you need to expand the view. Double click the Middle mouse button at various points to pan around the schematic. (The point double-clicked on always moves to center screen.)

If you misplace an instance, select it, draw a “shark fin” stroke , then place the instance in the correct position and click the Left mouse button.

Unselect all instances with a stroke when you are finished.

Add the portin and portout Instances to the Sheet

1. Place the mouse cursor on the Active Symbol window, press the right mouse button and choose: **Symbol History.**

2. Click on **portin** and execute the form with a stroke. The **portin** symbol now becomes the active symbol.

3. Place the **portin** instances on the sheet as shown on the schematic.

4. Repeat the above procedure by selecting the **portout** symbol from the **Symbol History** list and placing the instances on the sheet.

5. Unselect all instances with a stroke when you are finished.

Connect the Instance Pins with Wires

1. Draw a stroke to include all schematic elements in the viewing area.

2. Draw a stroke to enter the "Add Wire" mode.

   The ADD WI prompt bar appears at the bottom of the window. The editor will remain in this mode until you click **Cancel**.

3. Connect all instance pins with wires as shown in the schematic diagram. Click once at the start of each wire, once for each corner point (vertex) and twice at the ending point. Unselect the wire with a stroke before drawing the next wire.

   If you place a net vertex in the wrong place and haven't finished defining the wire with a double-click, use the BackSpace key to move backward and delete the vertices.

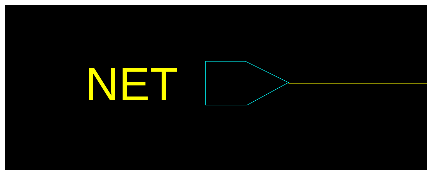
   If you finish a wire and want to remove it, make sure it is the only object selected, then draw a "D" stroke .

4. Click **Cancel** when you are finished adding all the wires.

Assign Each Port a Unique Net Name

Notice that every instance of **portin** and **portout** has a default NET property value of NET. You must change each of these to a unique name or they will be considered shorted together by the EDDM database.

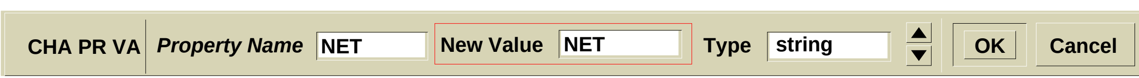
.



Select all the wires that are directly connected to the **portin** and **portout** instances. Make sure that you only have these nets selected.

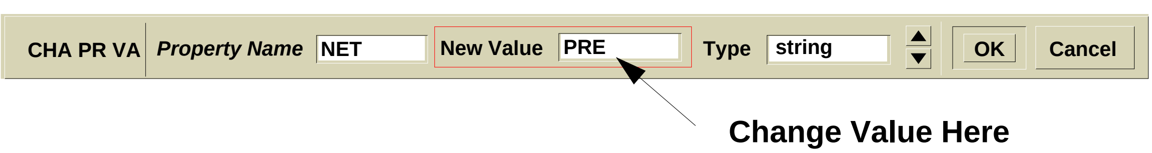
1. Press the Right mouse button and choose **Name Nets:** from the popup menu.

   This menu item lets you change the net name of each selected net in sequence. The Change Property Value prompt bar is displayed (as shown below) one-at-a-time for each net name from the top-most to the bottom-most selected net (without regard to horizontal placement.



   In this case, the first net name that is highlighted is the name associated with the top-most **portin** instance.

2. Use the BackSpace key to remove “NET” from the entry box, enter “**PRE**”, and press RETURN.



   Another prompt bar is displayed with the name of the net that is second-from-the-top.

3. Backspace over “NET”, enter “**Q**”, and press RETURN.

4. Change the net names of the remaining selected nets in a similar manner. Unselect the last object when you are finished.

Generate a Report to Discover Handle Names

The Schematic Editor assigns system-generated names to objects placed on a sheet. These system-generated names are called “handles”. Examples of handle names are I$2 for an instance, N$24 for a net, and P$26 for an instance pin. System-generated reports often refer to objects by their handle names. The following exercise will show you how to quickly discover the handle name of an object.

1. Click on one of the **inv** instances.

2. Draw a “square root sign” stroke .

   A report is generated that details the information about the object, and related objects, including the handle names. The handle name for the selected **inv** instance is ______________.

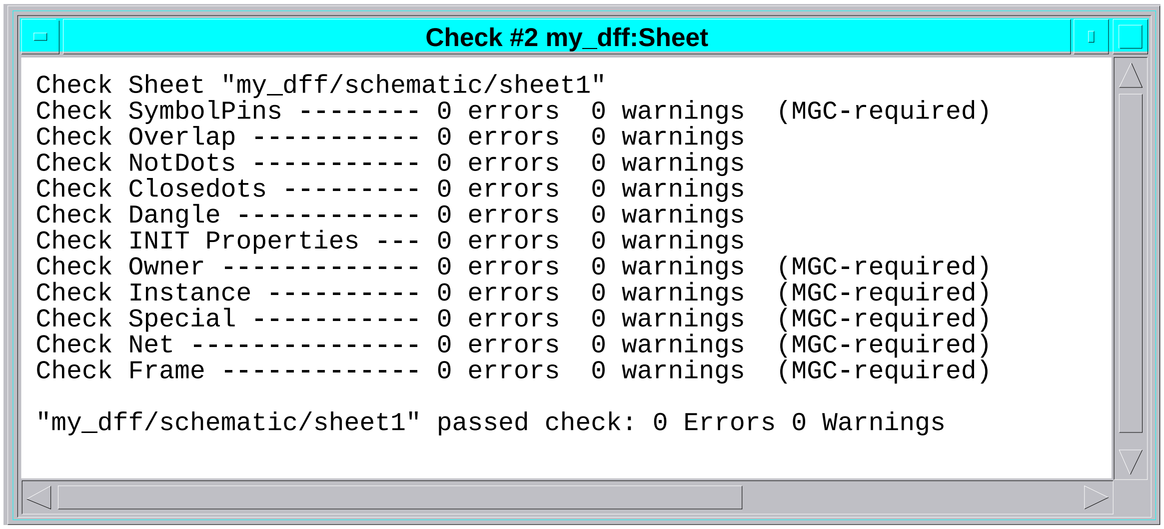
3. Close the report window with a stroke and unselect the **inv** instance with a stroke.

Check the Sheet

1. Choose the following pulldown menu item:

   Check > Sheet > With Defaults

   The Check Status window should appear in a few moments and look like the following:

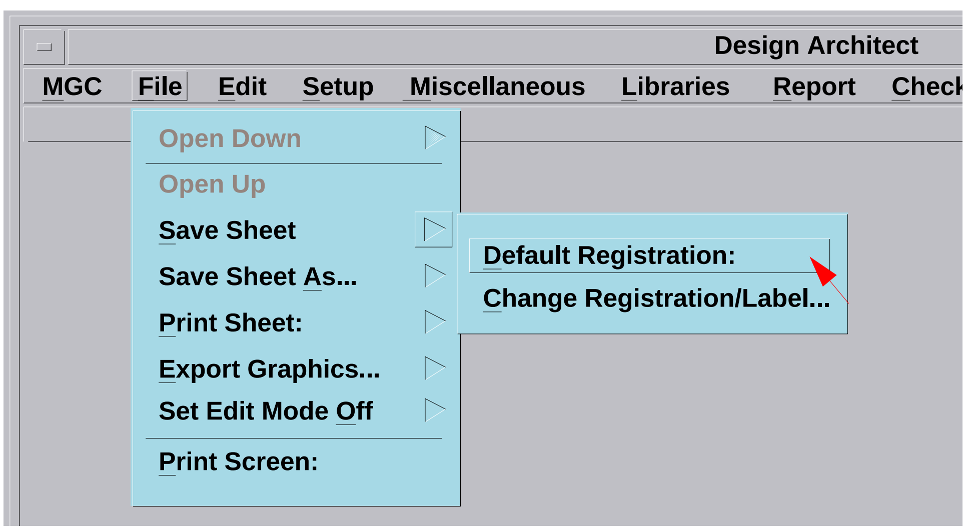


1. If your check operation reports errors, you can identify handle names by slowly clicking on the handle name in the Check Status window. The object associated with that handle name is selected on the sheet.

2. Close the Check Status window with a stroke.

3. Save the schematic to disk by executing **File** > **Save Sheet** > **Default Registration** from the pulldown menu.

.



Exercise 2: Net Connection Rules

The following exercise will help you become familiar with the net connection rules. The basic rule is that edit operations preserve the connectivity prior to the edit and do not make new connections. You must explicitly make the connection after the edit operation.

1. Draw the stroke to view the entire sheet, then the stroke to unselect all.
2. Click on a single **inv** instance and draw a "C" stroke .
3. Place the copy of the **inv** instance so that one pin touches a net. Click the Left mouse button.

   Notice what happens. A not-dot is displayed at that junction point to indicate that a connection is possible, but won't be made without an explicit command from you.
4. Make the connection by drawing a "connect all" stroke .

   Notice that the not-dot disappears and a junction dot ■ takes its place.
5. With the **inv** instance still selected, draw a "D" stroke to delete it.

   The instance and the not-dot disappear from the sheet.

Exercise 3: Changing the Mouse Selection Filter

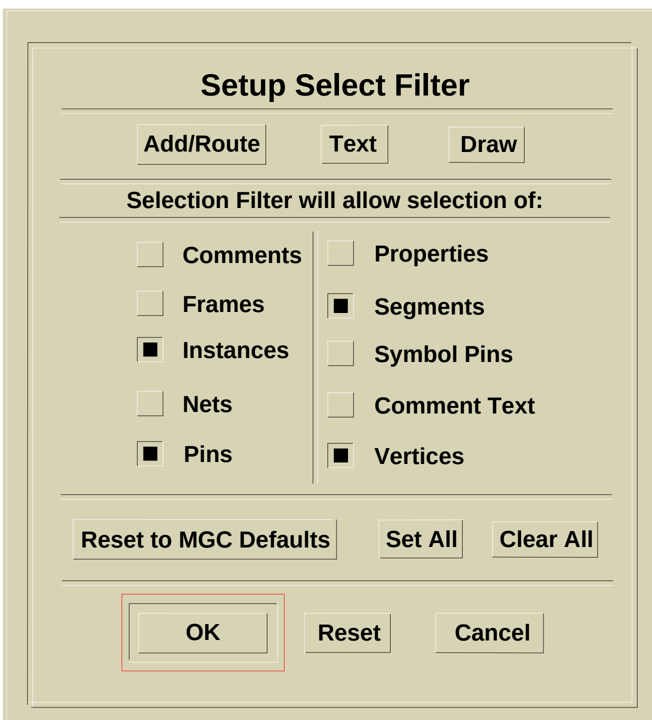
There is a selection filter attached to the mouse that controls what can be selected by pointing and clicking. Property text is not selectable with this filter set to default values. Verify this by clicking on some property text.

The Selection Filter is only associate with the mouse. When you move the mouse cursor over property text and press the F1 (Select) function key, the text is selected. This key allows you to quickly bypass the filter at times when you need to.

The following procedure will show you how to set and use the selection filter to select hard-to-reach objects. In this procedure you will set the filter to select only net vertices. You will select two vertices and move the net segment between, then reset the filter.

1. Activate the Schematic Window and draw a  stroke.

   The Select Filter dialog box is displayed as shown below:



2. Click the **Clear All** button, then click **Vertices.**

3. Draw a stroke to execute the form.

4. Now draw a selection box around two net vertices on a wire attached to the SET input of one of the latches.

5. Draw a “shark fin” stroke and move the cursor vertically to move the segment. Click the Left mouse button to place the segment.

6. Draw a stroke again and place the segment in its original position.

7. Draw a “U” stroke to unselect all.

8. Draw a stroke to bring up the Selection Filter form again.

9. Click **Set All** and draw stroke to execute the form.

   This effectively disables the filter and allows you select any object simply by clicking on it.

10. Practice selecting various objects to verify that the filter is disabled.

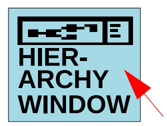
11. Close the Schematic window when you are finished.

Exercise 4: Browsing a Component in the Component Hierarchy Window

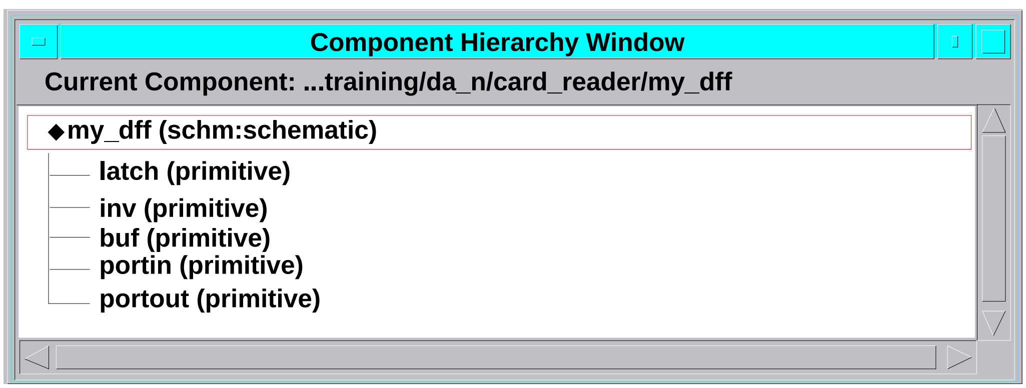
It is helpful to keep track of the hierarchical structure of your design. The following exercise will introduce you to the use of the Component Hierarchy Window.

1. Activate the DA Session window by clicking on it.

2. Click on the HIERARCHY WINDOW palette icon.



3. Select the **my_dff** component in the Navigator Window and click **OK**:

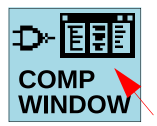


The Component Hierarchy window appears and shows you the structure of the **my_dff** schematic you just created. What does the word **primitive** mean?

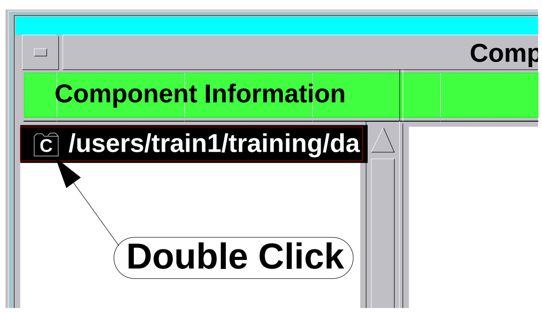
Exercise 5: Browsing a Component in the Component Window

It is helpful to understand the internal structure and content your design. You will be observing and changing this structure throughout this training course. The following exercise will introduce you to the use of the Component Window.

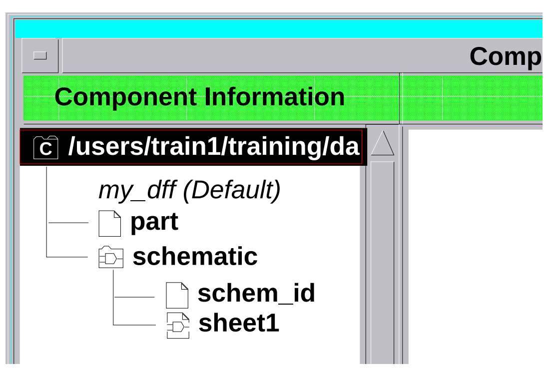
1. Activate the DA Session window by clicking on it.
2. Click on the COMP WINDOW icon:



3. Select the **my_dff** component in the Navigator Window and click **OK:**



4. Double click on the component icon:

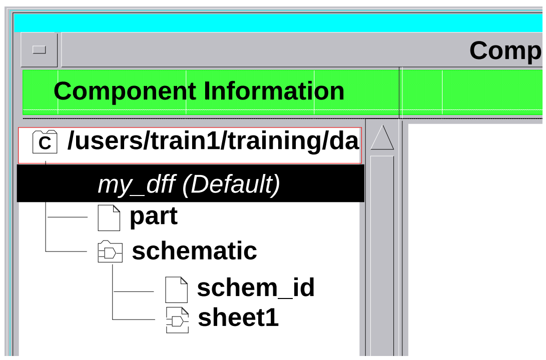


Notice that the component contains two objects, the **part** object and the **schematic**. The symbol does not show up because you haven't created it yet.

Also notice the name of the Component Interface table is *my_dff* and that it is marked as the *(Default)* interface.

5. Double click on the word “schematic” to list the content underneath.

6. Select the name *my_dff (Default)* as shown below:



The content of the Model Info window is displayed.

7. Place the cursor in the Component Information window, press the Right mouse button and select **Show Pins List.**

   Why are there no pins displayed in the **Pins** window?

   ____

   Why are there no body properties displayed in the **Body Properties** window?

   ____

8. Close the Component Window.

End of Lab Exercises

This concludes the lab exercises for this module.

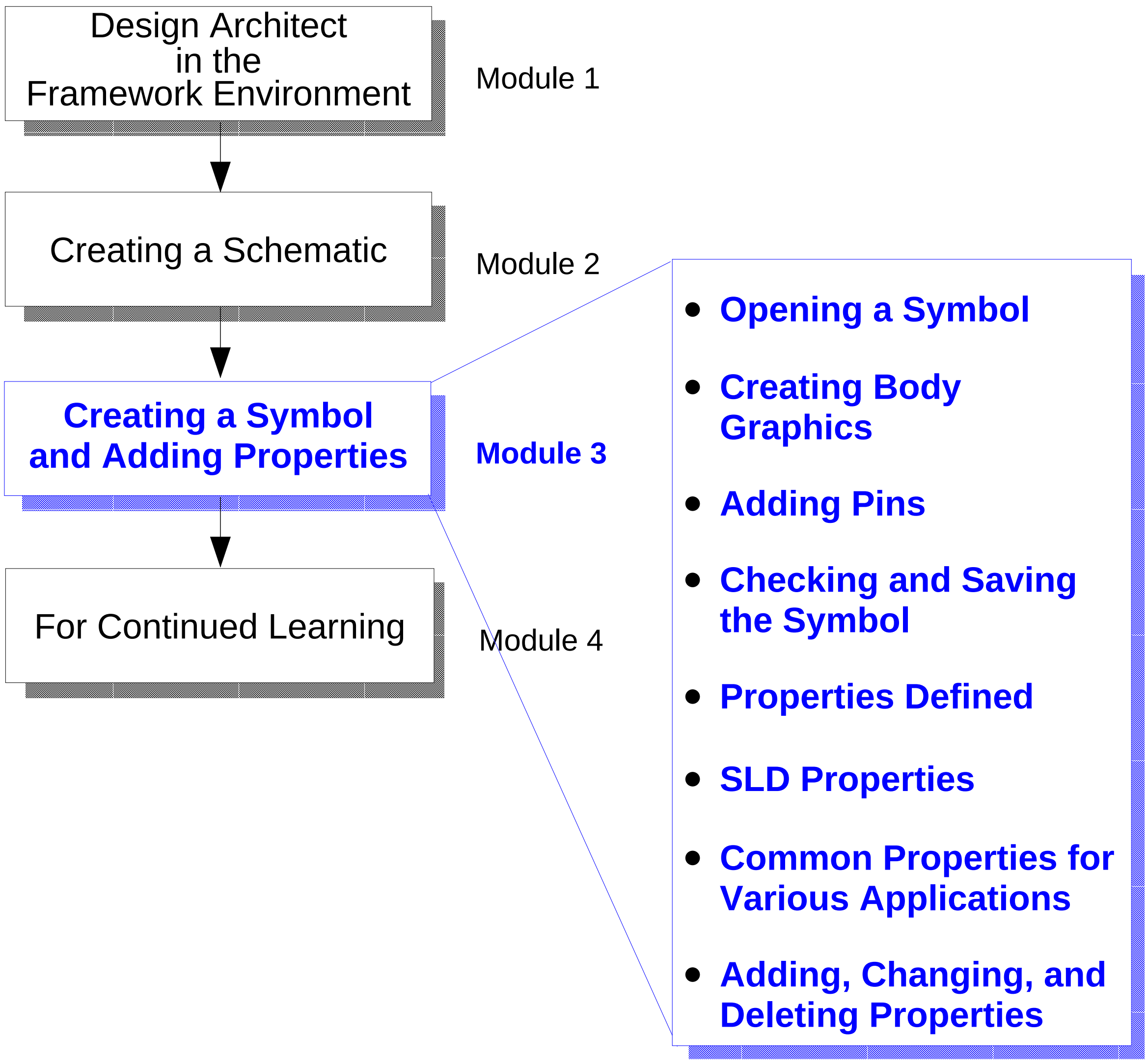
Module 3
Creating a Symbol and Adding Properties

Lesson 1 Creating a Symbol ____ 3-3

Lesson 2 Adding Properties ____ 3-23

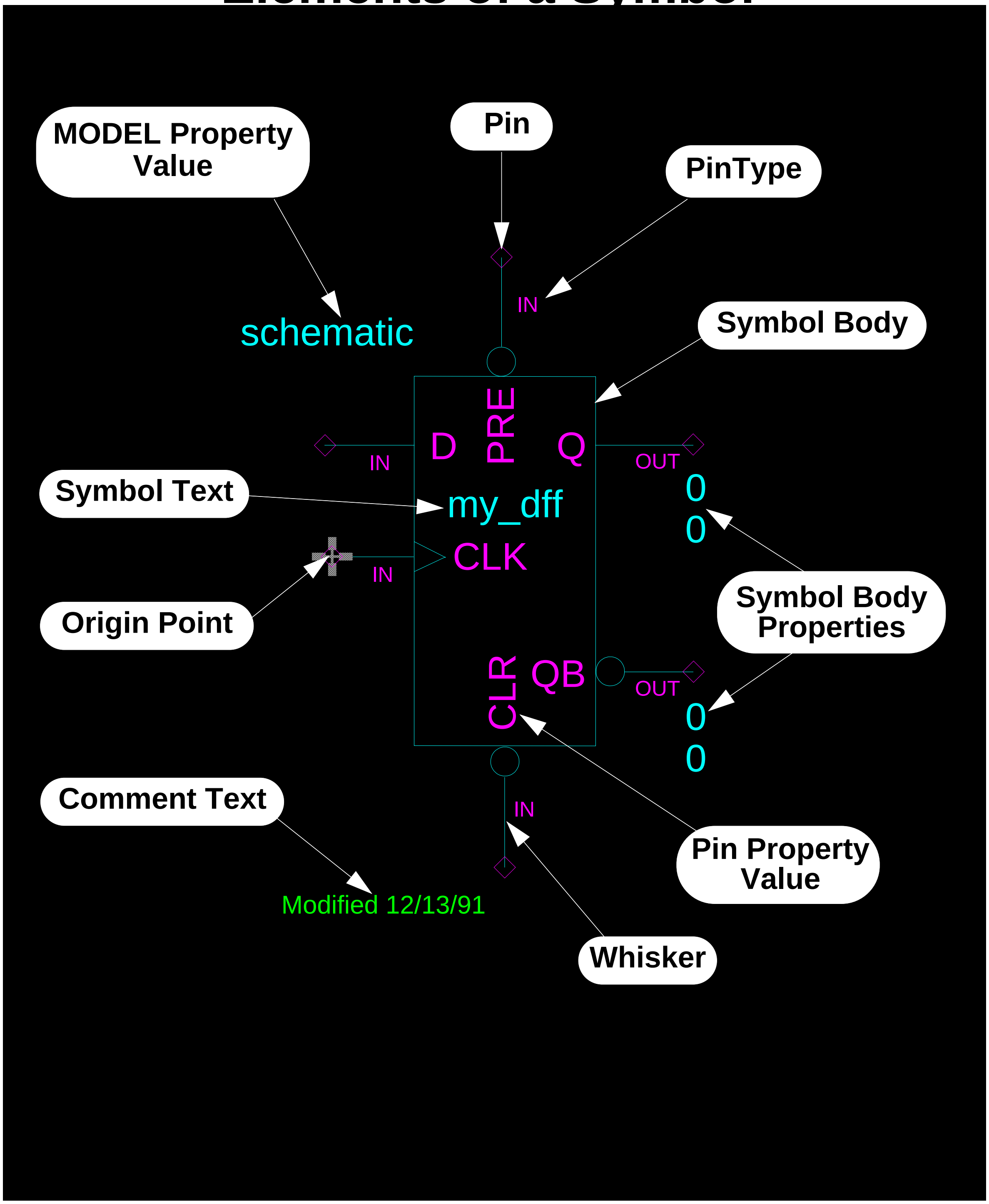
Lab Exercises ____ 3-59

Module 3 Overview



Lesson 1
Creating a Symbol

Elements of a Symbol



Elements of a Symbol

A component symbol is composed of four basic parts:

- **Shape (or symbol body)**. The graphical image of the symbol. The graphics that display the shape are called *symbol graphics*. The symbol shape usually conforms to an industry standard.

- **Pins**. Points where the symbol electronically connects when an instance is placed on a schematic sheet. Pin names must match the external nets on the schematic. These external nets are identified by the **portin** and **portout** instances.

- **Origin Point**. The reference point used to place the symbol on the schematic sheet.

- **Properties**. Provide information that can’t be represented graphically.

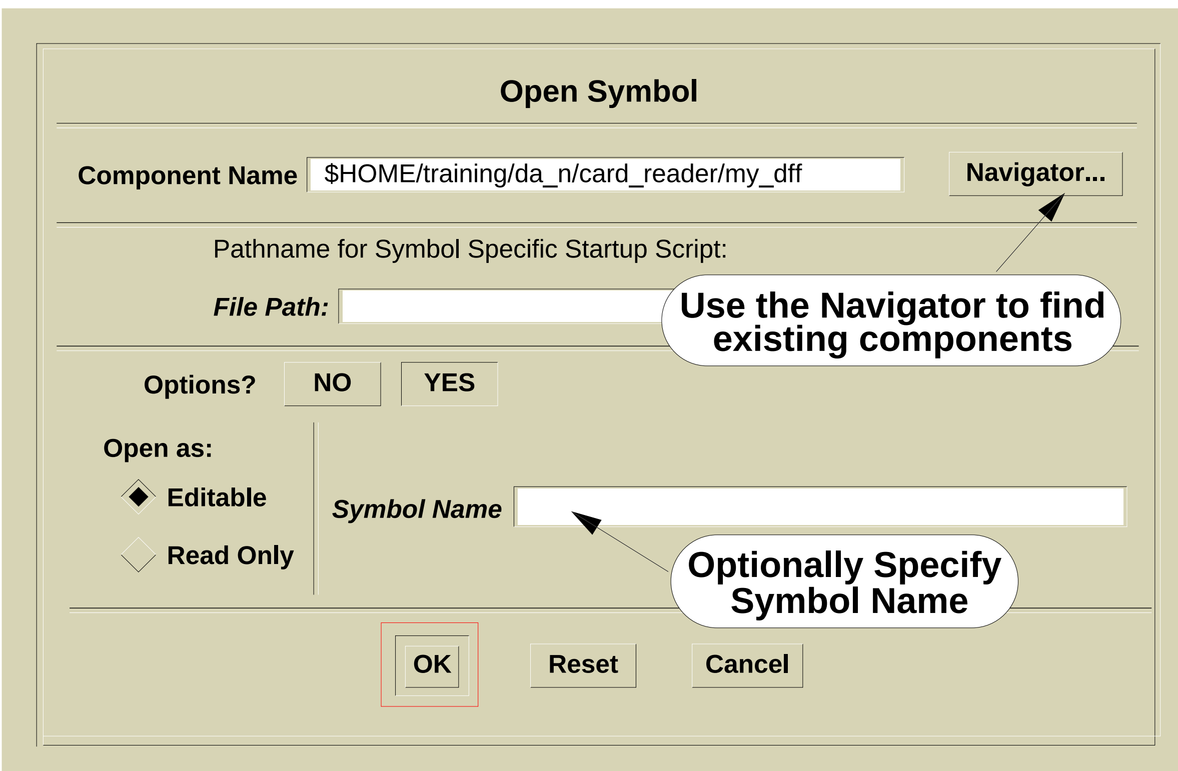
Optionally the symbol may contain the following:

- **Whiskers**. A symbol body can also have short lines called “whiskers” that project from the border of the symbol body to indicate where input and output pins are connected. These whiskers are a convention used in the Mentor Graphics component libraries, but they are not required. When you select the symbol instance on a sheet, the whiskers are also selected, as they are considered part of the symbol body.

- **Symbol Text**. Provides information about the symbol. When the symbol is instantiated on the sheet, the symbol text is visible.

Opening a Symbol

- **Click the OPEN SYMBOL icon**
- **Enter the component pathname**
- **Use the Navigator button for existing component**



Opening a Symbol

The Symbol Editor lets you create and edit component symbols that can then be placed in schematic sheets as instances. The internal functionality of the circuit being represented by the symbol may or may not be defined at the time the symbol is created.

After you click the **OPEN SYMBOL** icon, you must specify the pathname for the component. If a component structure doesn't exist at that location, Design Architect creates one before the Symbol wIndow is opened. Generally, if the component exists, it is more convenient to click the Navigator button and navigate to the component location.

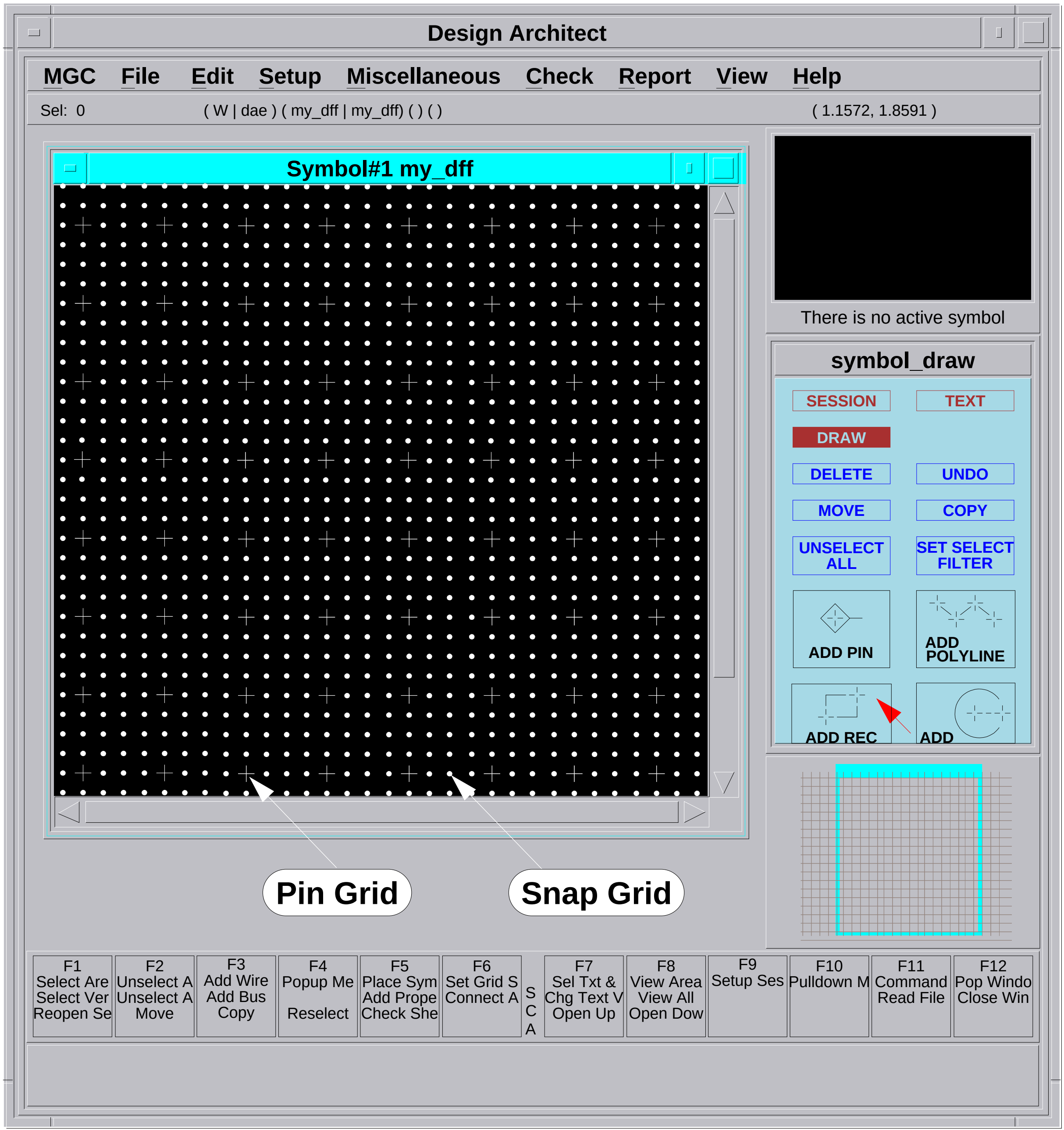
If you click **OK** without specifying a **Symbol Name**, the Symbol window will be opened on the symbol labeled "default". If a symbol doesn't exist, a blank window will come up and the new symbol will be named the same as the leaf name of the component interface. For example, **my_dff** is the default symbol of the component *.../training/da_n/card_reader/my_dff*.

You may specify a pathname in the ***File Path***: entry box. This is a pathname to an AMPLE file that contains commands that customize the Symbol Editor environment for this particular symbol.

When you click **Options? YES**, you will see the "long form" which allows you to change the default values for specific Symbol Editor attributes. It contains the additional following information:

- **Symbol Name**. The name of the symbol model. This entry box is either blank (the default symbol) or the name of the symbol that you selected through the Dialog Navigator. You can override the contents of this entry by typing a new symbol name.
- **Open as**. These are radio buttons; the default is generally **Editable**. Click **Read Only** to change the mode to read-only.

Symbol Editor Window



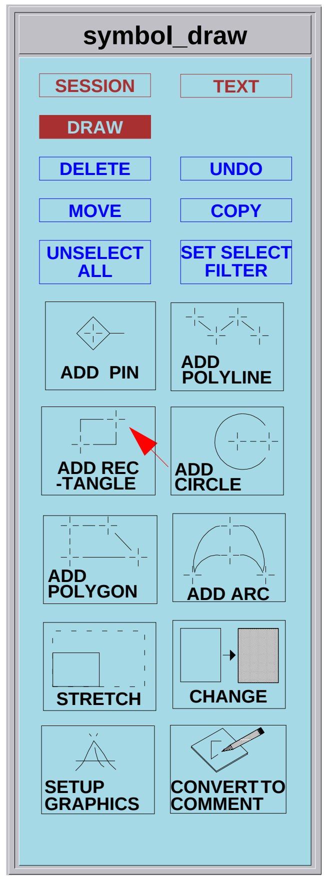
Symbol Editor Window

When the Symbol Editor window opens, it becomes the active window. The menu bar changes to symbol pulldown menus and the **symbol_draw** palette replaces the Session palette.The Context window contains the same functionality and the Active Symbol window contains a subset of the functionality available in the Schematic Editor.

The Symbol Editor window contains two grids. The coarse grid with the + marks is the Pin Grid. Symbol pins must fall in this grid. The fine grid is the Snap Grid. Symbol body graphics can be snapped to this grid.

The symbol_draw Palette

- Used to draw symbol body graphics
- Used to add pins to the symbol



The symbol_draw Palette

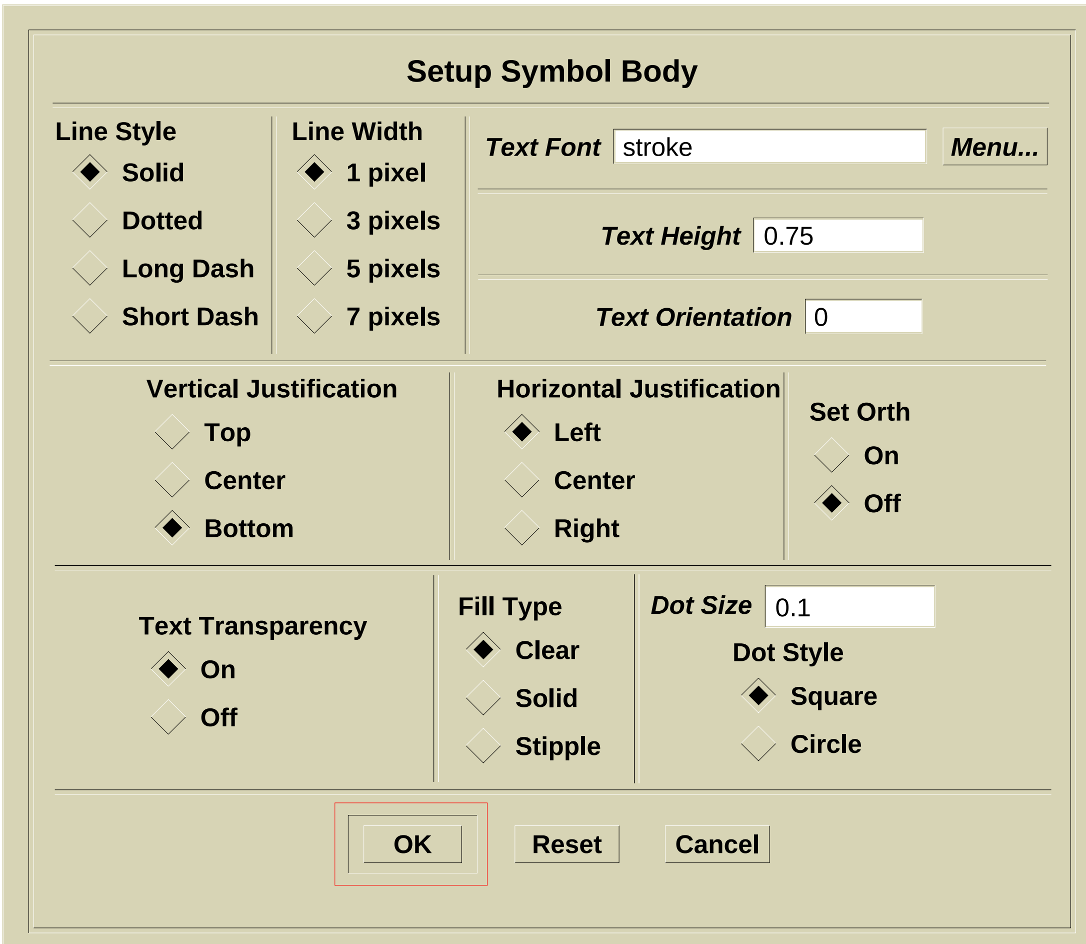
The **symbol_draw** palette is the primary palette used to draw symbol body graphics and add pins to the symbol. You can draw a rectangular body, or any shape consisting of line segments, arcs, and circles. The following table shows the number of location points and the mouse actions required.

Graphics	Prompts	Action
Polyline	initial, end segments (2 minimum)	Click at initial point; click at end of each segment. Double-click to terminate.
Rectangle	diagonal corners (2)	Press at one corner, drag, release to define opposite corner of rectangle.
Arc	initial, end, arc (3)	Click at each end, then at arc point.
Polygon	polygon points (3 minimum)	Click at each point of polygon, double-click to end; segment from last to first point is automatically drawn.
Circle	center, radius (2)	Press button at center, drag, release at radius point.
Two Point Line	initial, end (2)	Press button at one end, drag to other end, release.
Dot	point (1)	Click at dot location.
Text	text, location point (1)	Enter text in prompt bar, drag image of text, and click at desired text location.

Objects placed in the symbol edit area are automatically selected by default. You can turn auto-selection mode off by choosing **Setup** > **Other Options** > **Autoselect Off**. To turn auto-selection back on again, choose **Setup** > **Other Options** > **Autoselect On** (this menu item is a toggle). You can also resize an object by using the STRETCH feature.

Setting the Symbol Body Defaults

- **To set defaults, use Setup > Symbol Body**

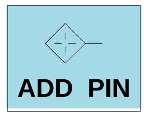


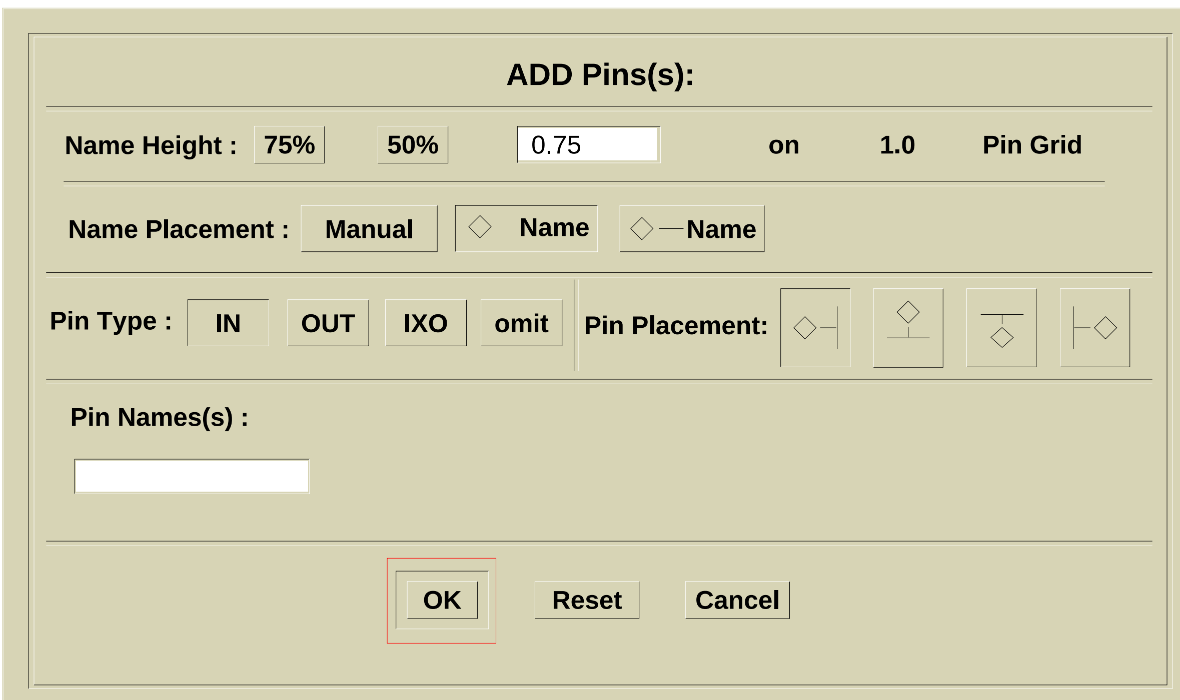
Setting the Symbol Body Defaults

When you create symbol graphics and text, default values are used for line style, line width, polygon fill, and text attributes such as font style, text height, justification, and orientation. You can change these defaults by choosing **Setup > Symbol Body** or the **Setup Graphics** icon (changes only the symbol graphics). The next time you create a symbol graphic or add symbol text, the default values that you specified are displayed.

Adding Pins

- **Electrical connection between symbol body and net**
- **Use the popup menu or the palette icon**





Adding Pins

A symbol is an abstract representation of an associated functional model like a schematic. A symbol pin on a symbol is an electrical connection that represents an external port on the functional model underneath. There must be a one-to-one match between symbol pins and the ports on the functional model. That is, the pin name must match the port name on the schematic.

When a symbol is instantiated on a schematic, the instance pins (which are an active reflection of the symbol pins) become the locations at which net connections are made. Instance pins are different than symbol pins because instance pins have a builtin net vertex.

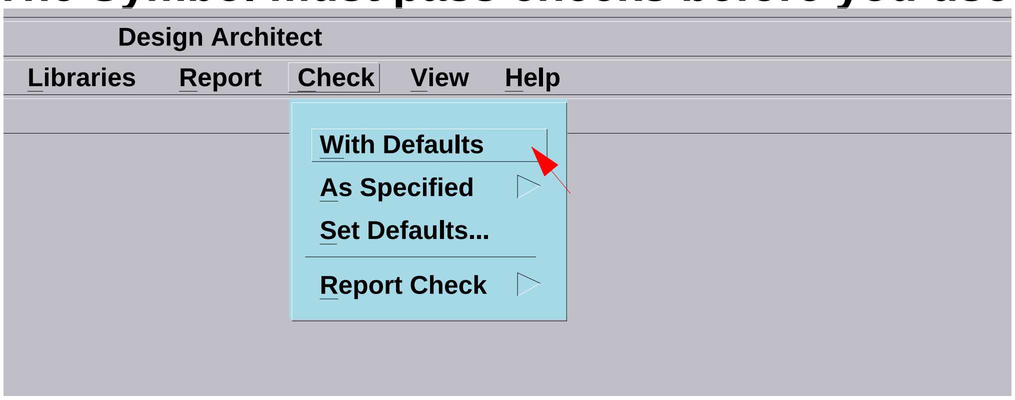
Name Height indicates the height of the pin name with respect to the pin grid. **Name Placement** has three types of pin name placement:

- **Manual**. The pin name and text location are positioned manually.
- **Name (with diamond)**. The graphical pin indicator is created and the pin name is positioned next to it.
- **Name (with diamond and whisker)**. The graphical pin indicator and whisker are created and the pin name is positioned next to it.

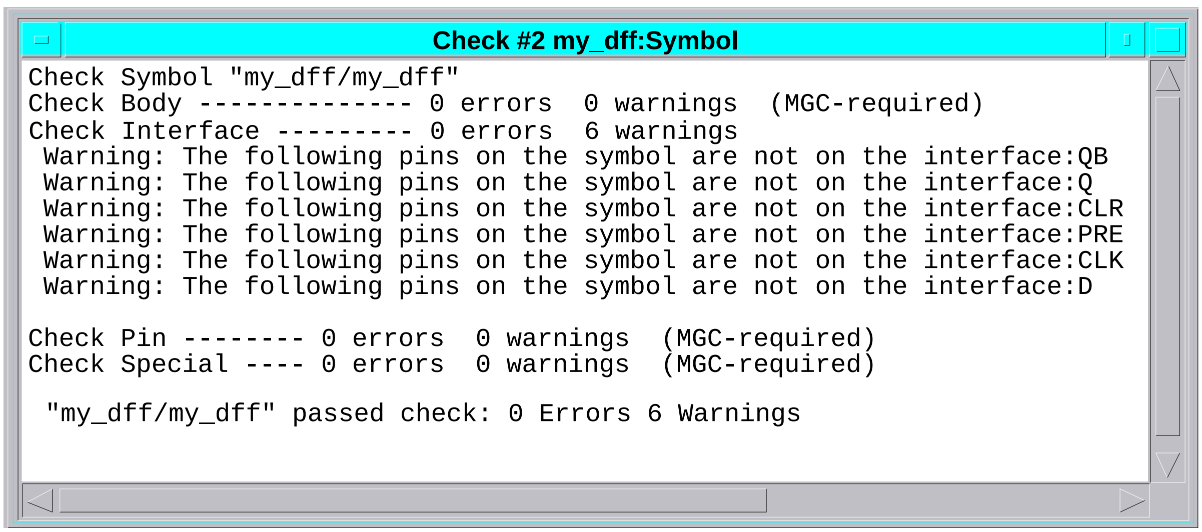
The PinType property associated with each pin can be either IN, OUT, IXO, or omitted. **Pin Placement** designates if the pin is placed to the left, top, bottom, or to the right of the symbol body.

Checking the Symbol

- **The Symbol must pass checks before you use it**



- **Typical warnings when changes are made**



Checking the Symbol

After creating a symbol body, adding pins, and adding properties, you must check the symbol. A symbol must pass certain checks before you can instantiate it on a schematic sheet. The following check categories are required by Mentor Graphics applications:

- **Symbol pin**. Verifies that at least one pin is present on the symbol, all symbol pins have unique names, and symbol property values have valid expression syntax.
- **Symbol body**. Verifies the existence of symbol graphics, checks expression syntax in property values, and looks for duplicate property names on the symbol body objects.
- **Special symbol**. Verifies proper construction of a symbol that represents a special instance, such as a bus ripper or offpage connector.

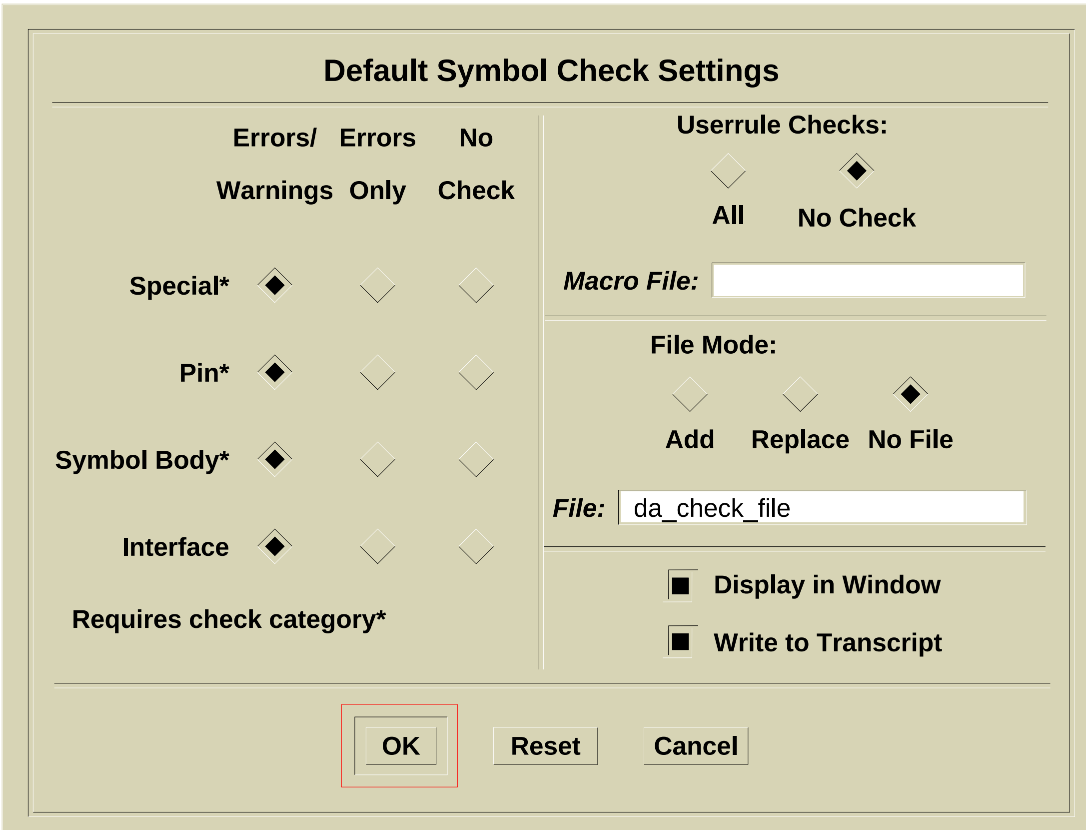
The following optional checks may also be performed:

- **Symbol userrules**. This is a user-defined category. You can create a macro file containing design rules that you want checked, and provide a pathname to the file. When the required checking is complete, the design rules specified in the macro file are checked.
- **Symbol interface**. Verifies that pins and properties match the component interface with which the symbol is registered.

To check using the default check settings, choose **Check > With Defaults**.

Changing Required Checks

- **Resets default checking requirements**
- **Check > Set Defaults... displays:**



- **Three levels of checking:**
  - o **Report all errors and warnings**
  - o **Report only errors**
  - o **Do not check**

Changing Required Checks

You can change the check defaults by choosing the **Check > Set Defaults** menu item. This brings up a Default Symbol Check Settings dialog box as shown on the left.

There are three levels of checking: report all errors and warnings, report only errors, and do not check. By default, all errors and warnings for symbol pins, symbol bodies, special symbols, and symbol interfaces are reported. You can change the check levels of symbol pins, the symbol body, special symbols, and the symbol's interfaces by clicking the appropriate check buttons.

If you want to specify a macro file containing additional design rule checks, click the **All** button under Userrule Checks, and type the pathname in the Macro File text entry box. The default userrule check setting is “No Check.”

Results normally appear in a check status report window, or in a transcript, but they are not saved to a file unless you specify a pathname and click the **File Mode: Add** button.

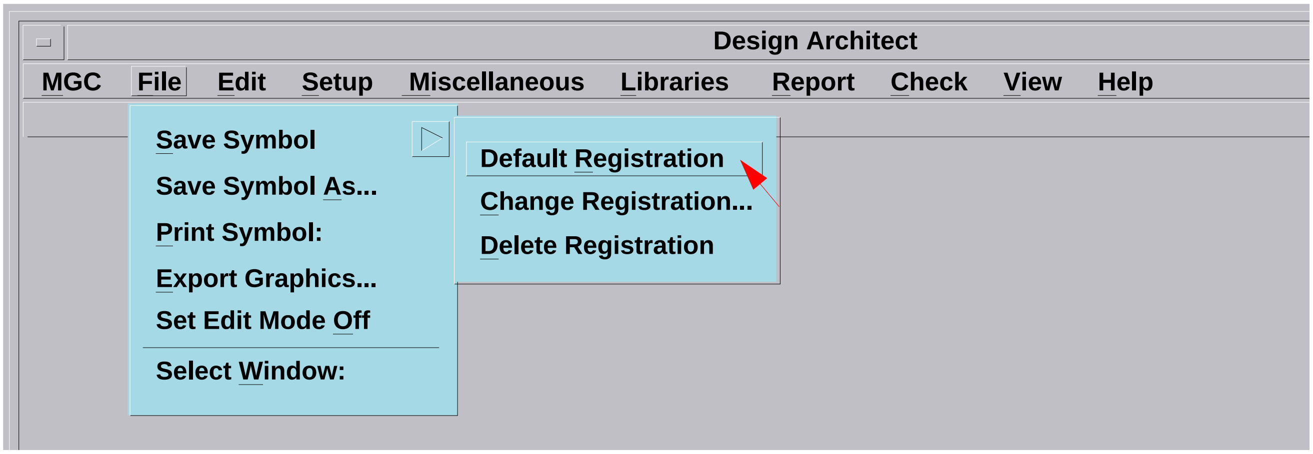
The next time that you choose **Check > With Defaults**, the symbol default check levels will reflect the modifications you made in the dialog box for the duration of the Design Architect session.

Note

If you change the checking level of required checks to **No Check**, Design Architect views the symbol as not passing the required checks. Therefore, you will not be able to instantiate the symbol on a sheet.

Saving the Symbol

- **File > Save Symbol**



- **Always check the symbol before saving**
- **Unchecked symbols can't be instantiated**

Saving the Symbol

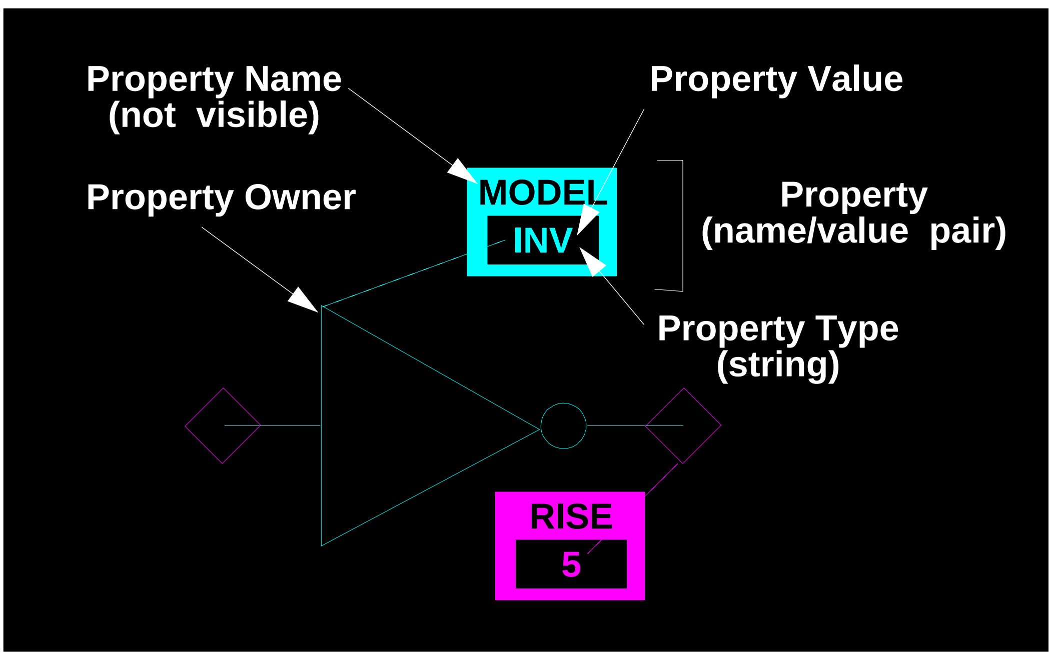
You choose the **File > Save Symbol** pulldown menu item to save your symbol to disk.

Always check your symbol before you save it. If your symbol is not successfully checked, it can still be saved to disk, but it cannot be instantiated.

Lesson 2
Adding Properties

What is a Property?

- A name/value pair
- Typically attached to a design object
- Conveys information that can't be represented graphically
- Can be used to establish horiz/vert connectivity (Structured Logic Design properties)
- Has the following characteristics



What is a Property?

A *property* is a name/value pair that is typically attached to a design object. Properties define design characteristics that cannot be easily represented graphically on a schematic. For example, in the illustration to the left, the output pin on the inverter owns a property called RISE with a value of 5 nanoseconds. This represents the total rise time of the gate. The instance body owns a property called MODEL with a value of INV. This MODEL property tells the logic simulator to use a builtin primitive for an inverter.

As shown in the illustration, properties have the following characteristic:

- **Name**. Typically identifies the purpose of the property value. The property name (and surrounding box depicted in the illustration) is never visible on the schematic or symbol. The name is retrieved through the use of reports that are generated on the owner or value.
- **Value**. Describes the design characteristic in the form of text. The text may or may not be visible on the schematic. If visible, the text takes on the color of the owner. If you select the text value and move it, a rubber banding line will show you the owner object (as shown in the illustration).

  In some cases, a property name may be defined and attached to an owner without defining a value. The value is then defined later in a downstream application. An example is assigning a pin number property (pin_no) to a logical symbol pin without declaring a value, then allowing Board Station to assign a physical pin number when the design is physically packaged.
- **Type**. Identifies the kind of value that is valid for a particular property. The valid types include string, number, expression, and triplet.
- **Owner**. Properties can be restricted to being owned only by certain object types for which the property information is most meaningful. For example, pins own Pin_no properties; the value of Pin_no properties are meaningful only to pins.

Most property names and values are defined when a symbol is created, but additional properties may be added to instances on a sheet.

Property Ownership

- **A property is typically attached to a design object called an "owner"**
- **Properties can be restricted to certain types of owners**
- **Allows you to manipulate properties by specifying the owner object**
- **The following design objects can own properties:**
    - **Symbol Bodies**
    - **Symbol Pins**
    - **Instance Bodies**
    - **Instance Pins**
    - **Nets (wires and buses)**
    - **Comment Graphics (not Comment Text)**
    - **Frames**

Property Ownership

Properties are typically attached to a single object in the design. When you create a property, you can restrict which objects can own that property through the **Setup > Property Owner/Type > Property Owner...** pulldown menu item. Other common Mentor Graphics properties have predefined owners assigned. You can see a list of these properties by choosing the pulldown menu item **Report >Default Property Values...** Once property ownership is defined for a property, Design Architect does not allow that property to be assigned to owners that are not valid for the property name.

In Design Architect, the following objects can own properties:

- **Symbol Bodies**. Pertains to the symbol body graphics in a Symbol window.
- **Symbol Pins**. Pertains to the symbol pins in a Symbol window or a Schematic window
- **Instance Bodies** Pertains to instance bodies on a schematic sheet.
- **Instance Pins** Pertains to instance pins on a schematic sheet.
- **Nets** Pertains to wires or buses on a schematic sheet.
- **Comment Graphics**. Pertains to comment graphics in a Symbol window or a Schematic window. Comment text cannot own a property.
- **Frames**. Pertains to frames on a schematic sheet.

Because extensive property manipulation functions are provided, you can manipulate the attached properties of selected owner object(s). This concept is discussed later in this module.

Property Types

- **A property value must have a property type assigned to it**
- **Legal types are:**
  - **String: IN**
  - **Number (integer, real, exponential): 2.5**
  - **Expression (string or arithmetic): x + 5**
  - **Triplet (3-valued property): 5 10 15**

Property Types

A property value must have a property type assigned to it. A *property type* identifies the property value's data type. The legal property types are:

- **String**. An ASCII character string.
- **Number**. An integer, real, or exponential number.
- **Expression**. A combination of variables, constant values, and arithmetic or logical operators defined by AMPLE expression syntax. For example, the expression (x+5), contains a variable, “x”, a constant value, “5”, and an arithmetic operator, “+”.

  Expressions are typically used to redefine property values in commonly used components without having to redesign the component. Expressions can also be used within the range specification for a net or pin name (net and pin property values). Variables in expressions are evaluated as needed. For example, expressions are evaluated when a sheet is checked, or when a design viewpoint is created with an expression defined.
- **Triplet**. The special property type “triplet” is a 3-valued property used to describe the best-case/typical/worst-case timing values used in timing analysis. The three values of a triplet can be separated by commas or spaces (for example, “5,7,10”). The value, whether entered as a string, number, or an expression type, is evaluated as a number.

  If one value is specified, it is used for the best-case, worst-case, and typical values. If two values are specified, the first value is used for the best-case value, and the second value is for worst-case and typical values.

It is important for you to know what the property type is before assigning a property value to a particular property. For example, if the property type of property “A” is a character string, and you assigned the value of 95, this value is interpreted as a character string “95”, not the numerical value of 95.

Property Text Attributes

- **A property value is also called “property text”**
- **Attributes of property text include:**
  - o **Justification**
  - o **Orientation**
  - o **Height**
  - o **Font**

Property Text Attributes

When the value of a property is displayed, the value is called *property text*. Property text has several attributes that define what the text will look like once it is placed on the design. Each property can have its own unique set of attributes. Attributes that can be defined and later modified are:

- **Justification**. Sets the text's horizontal and vertical justification relative to a location point. Horizontal justification can be set to the left edge, center, or right edge of the text string. Vertical justification can be set to the top edge, center, or bottom edge of the text string. The justification point serves as a point around which it can be rotated. The system-defined default is bottom left.
- **Orientation**. Indicates the direction the text faces. Choose 0 to indicate text is oriented horizontally and read from left to right, or 90 to indicate text is oriented vertically and read bottom to top. The system-defined default is 0.
- **Height** A positive, non-zero number that specifies the height in user units. The system-defined default is 0.1875 inches.
- **Font** A registered font family that is valid for the workstation you are using. The system-default font type is “stroke.”

Symbol Property Text Switches

- **Visibility Switch: Determines whether a property value is visible on an instance**
  - **Hidden**
  - **Visible**
- **Stability Switch: Determines if a property value is changeable on an instance**

Switch Type	Change Values on Instance?	Delete on Instance?
Fixed	No	No
Protected	At instantiation time only	No
Variable	Yes	Yes
Nonremovable	Yes	No

Symbol Property Text Switches

Two property switches control whether a graphic symbol property value is visible on the symbol instance, and changeable on the symbol instance.

The *property visibility switch* controls whether the value of the property is visible (**Visible**) or invisible (**Hidden**) on the instance.

The *property stability switch* controls whether the value of the property can be changed on an instance. It can have one of the following values:

- **Fixed**. Values cannot be altered or deleted on an instance at instantiation time.
- **Protected**. Values can be altered on an instance at instantiation time. Once instantiated, the instance-specific property value cannot be changed.
- **Variable**. Values can be altered on an instance at instantiation time and through the Change Property commands.
- **Nonremovable**. Values can be altered on an instance and changed through Change Property commands; they cannot be deleted from the instance.

Other property attributes on an instance can be altered using Change Property commands.

Pin properties are assigned the values Visible and Fixed by default when the Pin property is created. Properties other than pins are assigned Visible and Variable by default. However, you can override these switches at Add Property time, in addition to changing the values of selected properties. You can also change the default values through the Setup Property Text command (discussed later in this module).

SLD Properties

- **Structured Logic Design(SLD) properties are special properties used to define and pass connectivity information**
- **The following list defines SLD properties and their owners:**

SLD Property	Owner	Purpose
NET	**Net Vertex** **Symbol Pin** **(with a “pin and net” owner)**	Establishes horizontal connectivity.
PIN	**Symbol Pin**	The value defines the pin name.
RULE	**Symbol Pin**	Defines which wire(s) get ripped from the bus.
INST	**Instance Body**	The value defines a unique name for the Instance.
CLASS	**Symbol Body** **Instance Body**	The value defines a Special Connection.
GLOBAL	**Symbol Body**	The value defines a design-wide net name.
FREXP	**Frame**	The value defines the FRAME expression.

SLD Properties

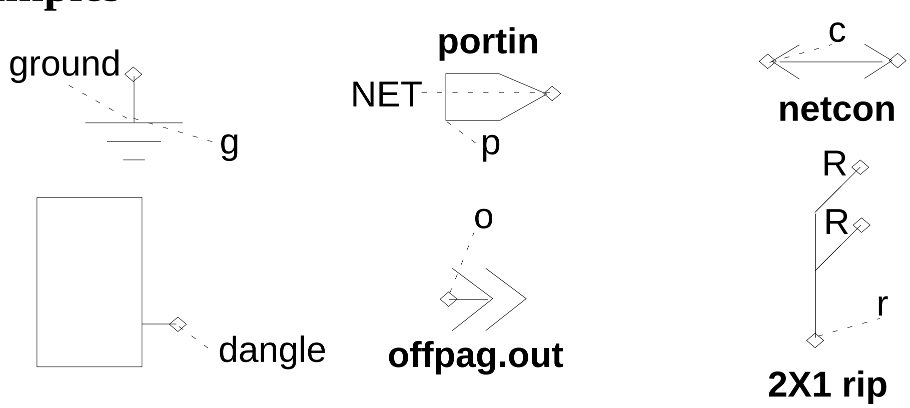
Structured Logic Design (SLD) properties are special properties built into Design Architect. They pass object identity and connectivity information about the design to the Design Viewpoint Editor (DVE) and downstream applications. SLD properties are unique in that SLD property values cannot be modified in downstream applications. SLD properties have default property owners. The list below briefly describes these properties.

- **NET**. Design Architect automatically assigns a name of the form N$ to each net. This property specifies horizontal connectivity by assigning unique names to wires, buses, ports, and offpage connectors. If a NET property and/or value is assigned to a Symbol Pin and an owner of “pin and net” is specified, the NET property and value are transferred to the instance pin vertex, when the symbol is instantiated.
- **PIN**. Specifies a unique name for a logical pin on a symbol.
- **RULE**. Specifies which wires are diverted from a bus to an output pin on a bus ripper component.
- **INST**. Design Architect automatically assigns a handle name like I$2 to each instance. The INST property allows you to give a custom name to a particular instance.
- **CLASS**. Specifies special connector devices, such as ports, bus rippers, offpage connectors, and net connectors.
- **GLOBAL**. Specifies net connectivity without drawing the wires on a schematic. Global connectivity is established both vertically and horizontally in the design hierarchy by giving nets the same name as the Global property value. Examples are the $MGC_GENLIB components **vcc** and **ground**.
- **FREXP**. Specifies the frame expression of a Frame. Frame expressions allow frames to be identified by type and value, so that applications which use frames can interpret the information.

Class Property Values

Property Value	Description
c	**Connector:** Connects differently named nets together.
g	**Global:** Connects a net both vertically and horizontally across a hierarchical design.
p	**Port:** Defines an external net on a schematic.
r	**Ripper:** Extracts a range of nets from a bus.
o	**Off-page connector:** Identifies nets by the same name that reside on different sheets of a schematic.
n	**Null:** Defines an object as electrically inert.
dangle	Tells the system that a dangling instance pin or net vertex should not cause a check warning.

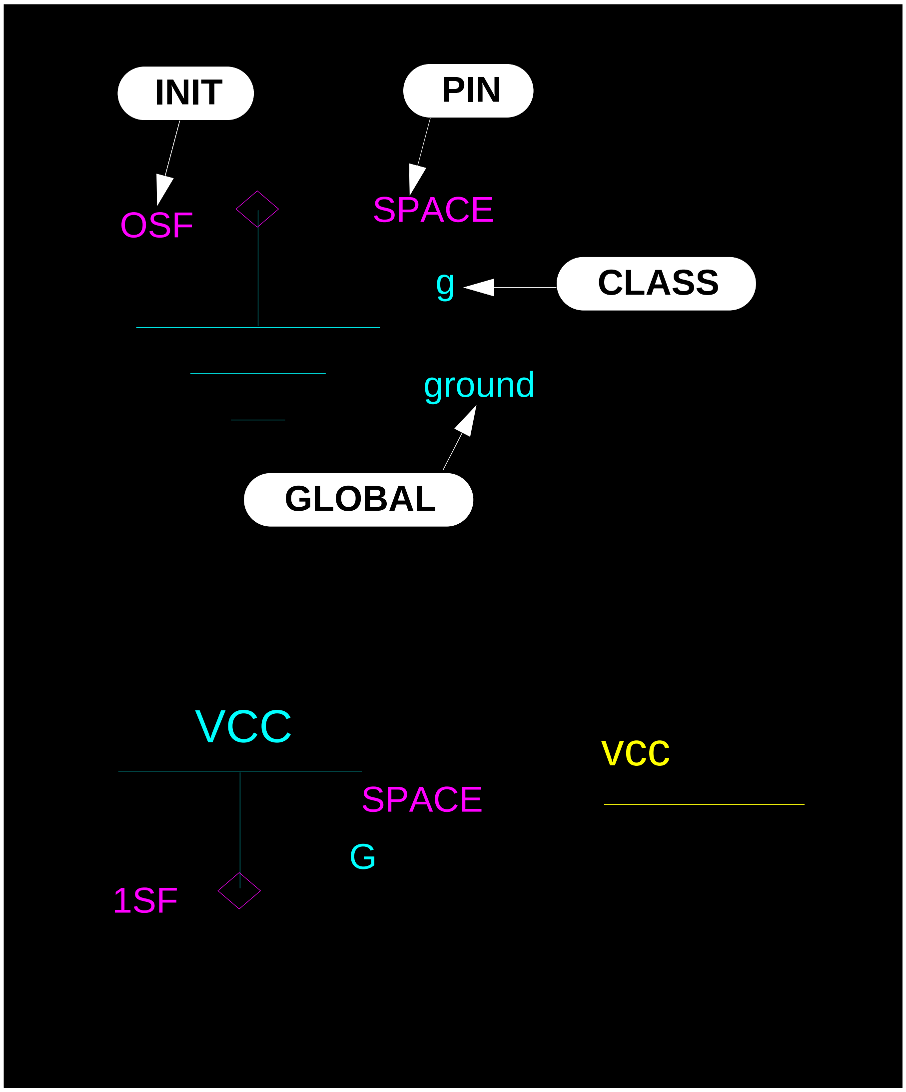
Examples



Class Property Values

The CLASS property tells the system what kind of connection or connector the owner represents. The different CLASS property values are defined on the left. Examples of the different connectors are shown below the table.

Examples of Global Nets



Examples of Global Nets

ground

The ground symbol on the left defines a global net named **ground** for the design. The CLASS property is attached to symbol body. The value **g** tells the system that the instance represents a global net. The GLOBAL property defines the net name which is **ground**. The PIN property specifies the name of the pin and is a required property; the value is **SPACE**, but it could be defined to be any string. The INIT property is attached to the pin and tells the simulator to initialize the pin with a logic 0(zero), with a strength STRONG, and a force type of FIXED. It is common to initialize a global net to a constant value, but this is not a requirement.

VCC

The VCC symbol is similar to the ground in that a CLASS property with a value of **G** is attached to the symbol body. Notice the value can be upper or lower case. The value of the GLOBAL property defines the net name as **VCC**. The INIT property is attached to the pin and causes the simulator to initialize the pin to a logic 1, drive strength STRONG, and a force type of FIXED. To the right of the VCC symbol in the diagram is a net. The net vertex owns a NET property with a value of **vcc**. This causes a connection with the VCC global net, even though there is no physical connection between the net and the pin on the VCC instance.

Attaching Nets to PCB Power Planes

PCB power planes are represented by net names that are defined as “global”. If a net is attached to something other than a “global” net, the net is treated as a normal trace on the signal layer.

Other Global Nets

Global nets can also be defined for things like clocks and set/reset lines. How would you create a global net for a Master Reset line? ______________________

____

____

Common Digital Simulation Properties

- **Properties on Symbol Pins**
  - **PINTYPE**
  - **RISE**
  - **FALL**
  - **DRIVE**
- **Properties on Symbol Bodies:**
  - **MODEL**
  - **MODELFILE**
- **The instance-specific property values can be changed in the downstream analysis tool**
- **You can back annotate changed values to the source schematic using Design Architect**

Common Digital Simulation Properties

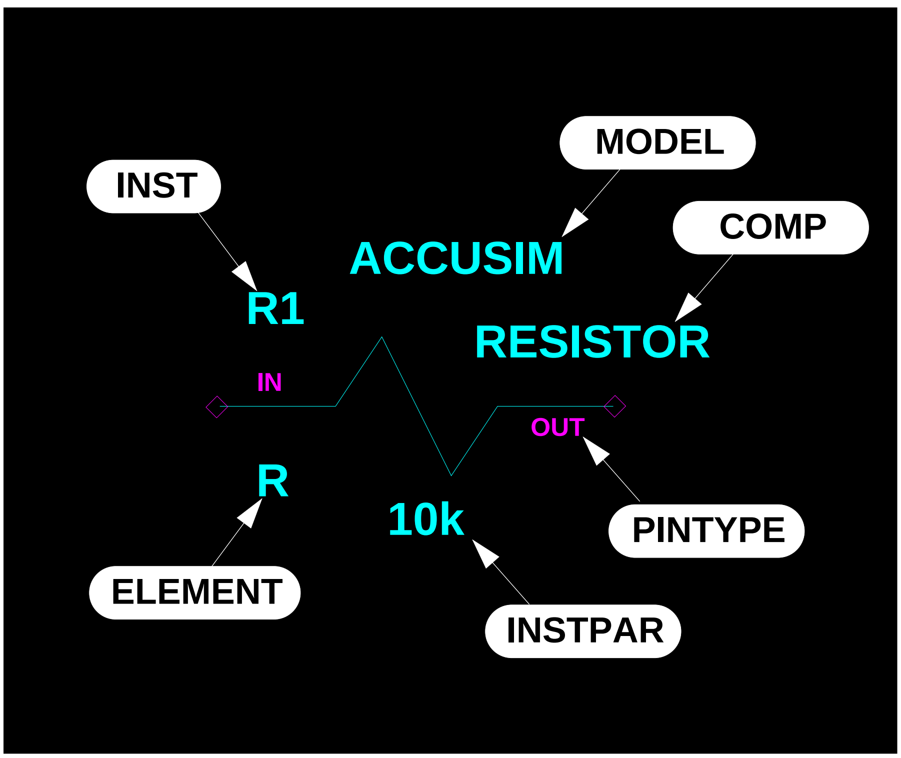
Mentor Graphics provides a list of predefined properties that are reserved for Mentor Graphics applications. This list includes properties with user-defined values which are used for the downstream applications and Structured Logic Design (SLD) properties.

Many downstream applications require a minimum set of properties that must be attached to the design. You must add or modify these properties to your design before you begin evaluation in any of the downstream applications. Once in the downstream application, you can change instance-specific property values and *back annotate* them to the design viewpoint. They can then be merged to the source schematic sheet at a later time using Design Architect. The following is a list of common properties that are used by the QuickSim digital simulator:

- **PINTYPE**. Specifies the direction of a pin. Possible values are IN, OUT, and IXO. This value may also be omitted.
- **RISE**. Specifies minimum, typical, and maximum rise delay for an instance. The time is specified in nanoseconds.
- **FALL**. Specifies minimum, typical, and maximum fall delay for an instance. The time is specified in nanoseconds.
- **DRIVE**. Specifies the output pin's drive strength. Some possible values are SSS (for CMOS, TTL, and ECL technologies), SRR (for NMOS technologies), and RRS (for PMOS technologies).
- **MODEL**. Specifies which functional and timing descriptions are to be used for digital simulation. The MODEL property value is matched against the model labels in the model table to determine which model is chosen for a particular instance.
- **MODELFILE**. Specifies one or more pathnames of text files containing programmable logic device, memory, or analog device information.

Common Analog Simulation Properties

- The analog properties used depend on the type of analog device
- The following are common analog properties for a resistor:



Common Analog Simulation Properties

The analog-specific properties attached to a particular analog device depend on the type of device and the simulator being used. In total, there are more than thirty different analog-specific properties that may be assigned.

The illustration to the left shows typical analog properties assigned to a resistor. The properties are defined as follows:

MODEL Used in conjunction with the ELEMENT property to identify the functional description to be used for analog simulation.

ELEMENT Specifies a built-in instance type that is recognized by the analog simulator. This property is used in conjunction with the MODEL property to identify the functional description to be used for analog simulation.

COMP Provides a unique name, usually to identify a specific manufacturer's component.

INSTPAR Used to specify analog device parameters. In this case, the value of the resistor is 10k ohms.

INST Used to give a unique name to this particular instance. In this case, the resistor is labeled R1.

PINTYPE Specifies the direction of the pin. Possible values are IN, OUT, and IXO. This value may also be omitted.

Common PCB Layout Properties

- **The following properties must be added in Design Architect in order for Board Station to map logical symbols to physical parts:**
  - **All Symbol Pins must own a PIN_NO property.**
  - **All Symbol Bodies must own a COMP property.**
  - **All Symbol Bodies must own an instance-specific REF property.**
- **The <u>values</u> for these properties can be assign for the first time or changed in Board Station**
- **You can back annotate changed property values to the source schematic using Design Architect**

Common PCB Layout Properties

Board Station requires a minimum set of properties that must be attached to instances in order to map them to physical devices. You must add these properties in Design Architect before you invoke Board Station tools on the design. Although the properties must be defined and attached to the right owners, assigning values to the properties may be done in Board Station.

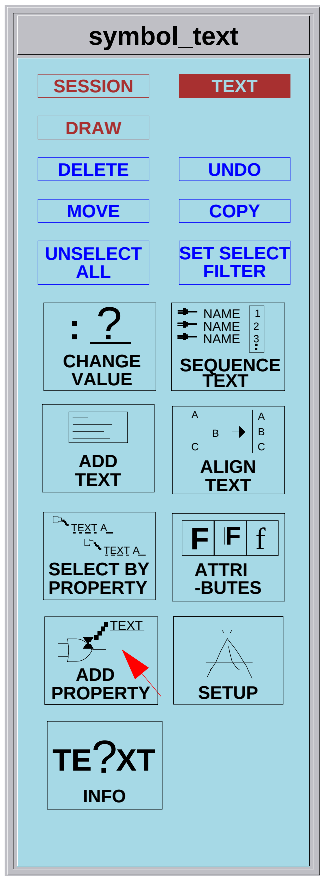
When these values are assigned for the first time or changed in Board Station, they are captured in a back annotation file. The back annotated properties may be merged back to the source schematic at a later time using Design Architect. This process will be discussed in more detail in a later module.

Board Station requires that the following properties must be on logical instances in order to map them to physical devices:

- **PIN_NO** Specifies the physical pin number to which the logical symbol pin is assigned.
- **COMP** Provides a unique name to help you identify the various components in a design. An instance with a COMP property is considered primitive by the PCB layout tools.
- **REF** Specifies which components are to be packaged together in the same physical package. This allows you the convenience to design with single gates, but then use a common REF property on those gates located in the same physical package. Some components use the Uxxx designation.

Adding Properties to Symbol Graphics

- Add properties to selected pins or symbol body
- Add properties to “logical” symbol
- Use the “text” palette



Adding Properties to Symbol Graphics

Properties can be attached to the symbol body, a symbol pin, or a comment object. It is also possible to add properties that have no graphic owner. This type of property is called a *logical symbol* property. These are properties not attached to the symbol body graphics, but rather owned by the “logical symbol”.

Using the **ADD PROPERTY** icon in the palette or the Shift-F5 function key only lets you add properties one at a time.

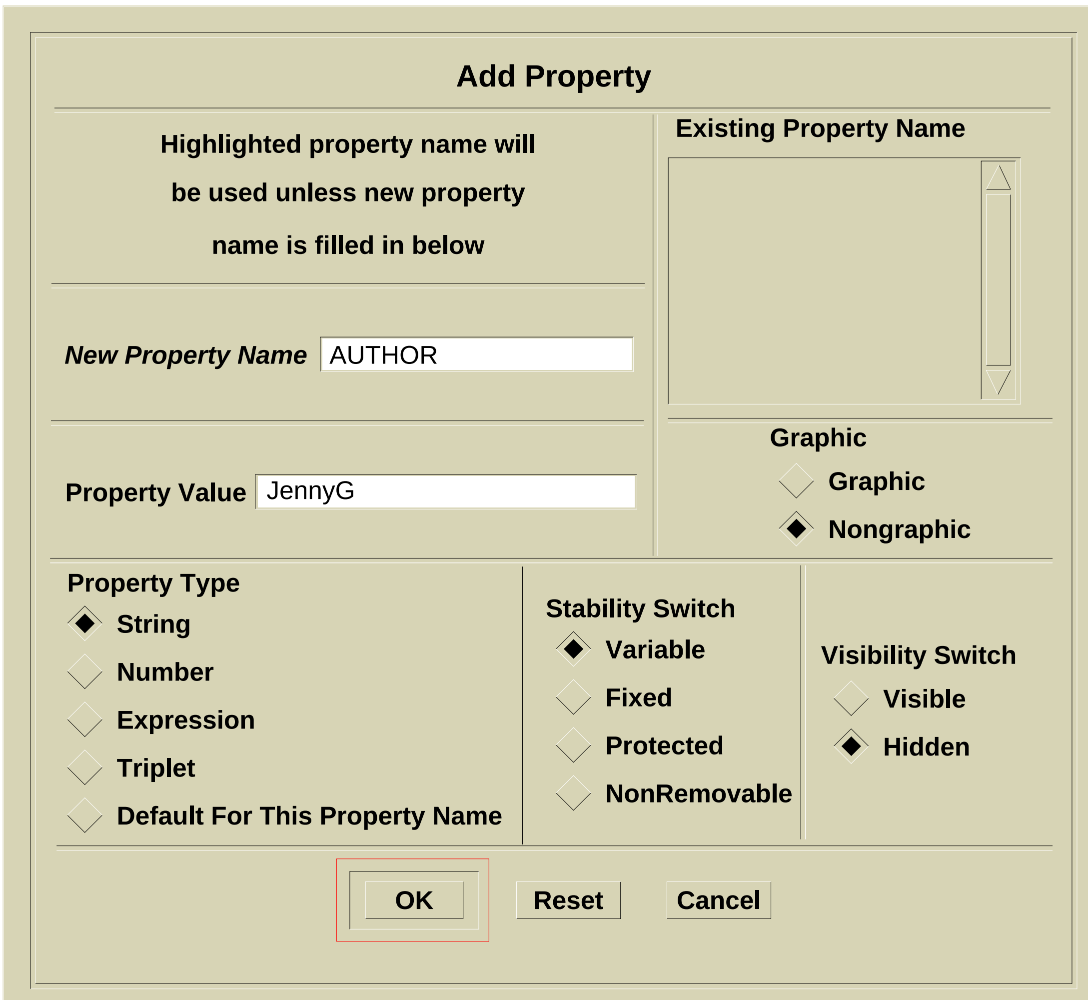
The **Properties > Add >** menu items provide methods of adding multiple property name/value pairs to objects. The following table shows the differences.

Menu Path	**Property Attributes (like “type”)**	**Selection Set**
Add Multiple Properties...	must be the same	same
Repeat Adding Single Properties > Use Current Selection...	can be different	same
Repeat Adding Single Properties > Use Changing Selection...	can be different	different

Adding “Logical Symbol” Properties

1. **Unselect All**

2. **(popup)Properties(Logical) > Add Single Property**



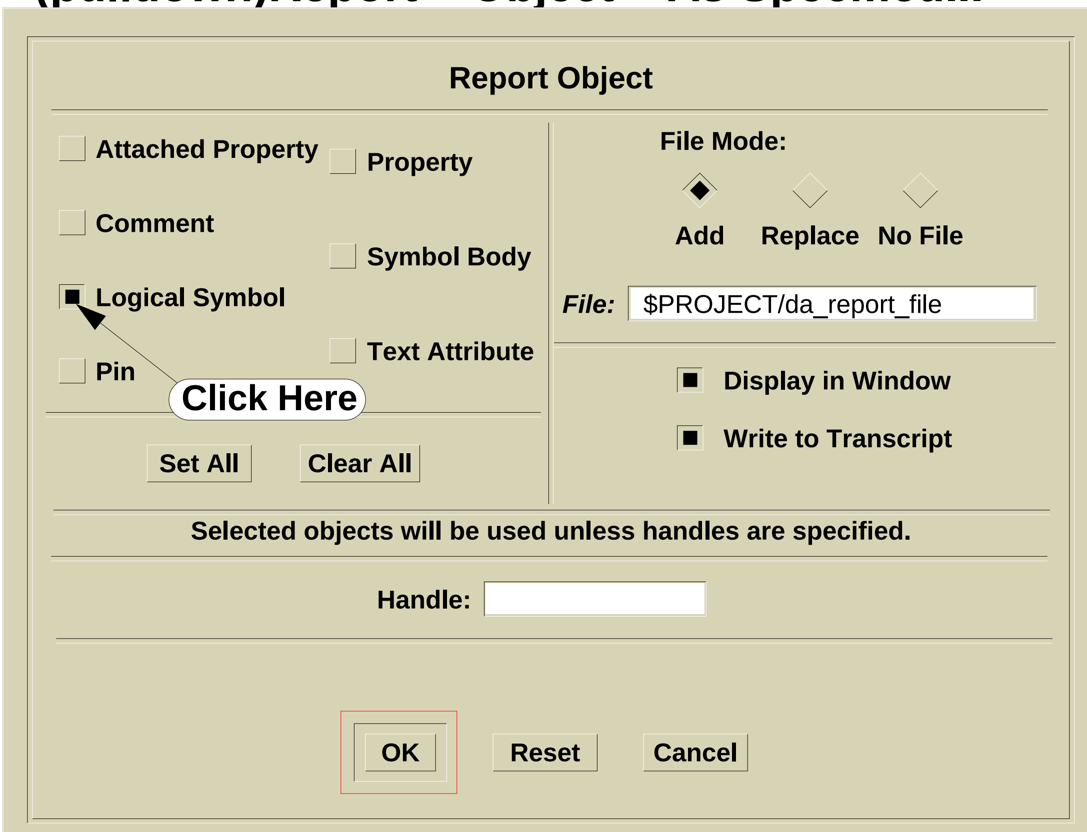
Adding "Logical Symbol" Properties

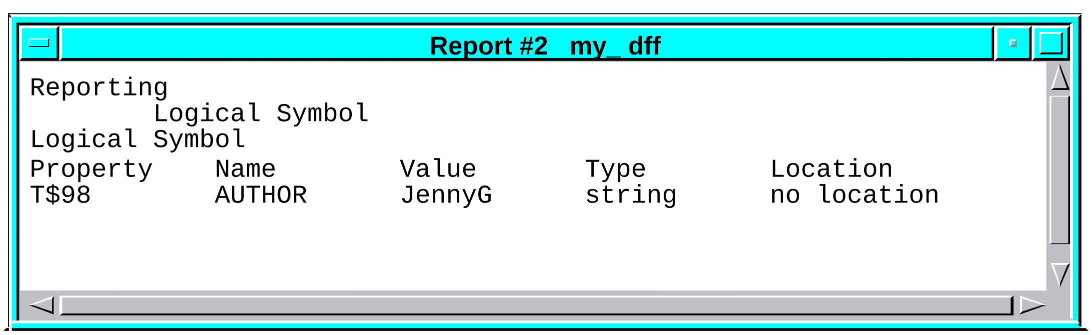
Logical symbol properties are properties that are not attached to any design object on the symbol. They "free float". If they are made "Graphic", they show up in the Symbol Editor window in the "gold" color. If they are defined as "Nongraphic", they are invisible and the only way to see them is to generate a report on Logical Symbol properties.

Logical symbol properties are usually defined to hold information that is normally not to be seen by the end user. In the example on the left, the property AUTHOR is defined as a logical symbol property in order to identify the creator of the symbol.

Reporting On and Deleting “Logical Symbol” Properties

- **(pulldown)Report > Object > As Specified...**





- **Edit > Edit Operations > Delete > Property...**

Reporting On and Deleting "Logical Symbol" Properties

The only way to see nongraphic logical symbol properties is to generate a report as shown on the left. From the menu bar, choose **Report> Object> As Specified...**, click the **Logical Symbol** button on the form, and click **OK**.

If you create a symbol, add properties to the symbol body graphics, then delete the graphic object, the properties are not deleted but turn into logical symbol properties. If you then recreate the symbol body graphics and try to add the same properties, the system won't let you because the properties are already defined as logical symbol properties.

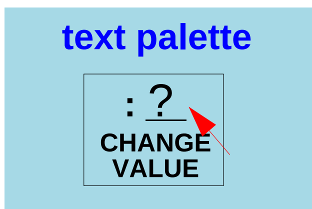
The only solution to the above dilemma is to first delete the logical symbol properties (by name), then re-attach the properties to the newly created symbol body graphics.

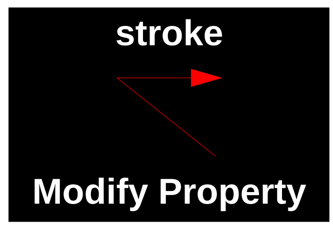
The pulldown menu path **Edit > Edit Operations > Delete > Properties...** is the only way to delete nongraphic logical symbol properties.

Changing Property Values

- **Ways to select properties**
  - **By owner** — **Select > Property Owner**
  - **By text** — **Select > Area > Property**
  - **By name** — **Select > By Property>Name**

- **Commonly used icon and stroke**





- **Commonly used popup menu items:**
  - **Change Values**
  - **Change Height**
  - **Change Attributes**

Changing Property Values

Selecting Properties

When you change a property value or attribute, at least one object must be selected. The object can be the *owner* of the property, such as a net vertex. a pin or an instance body.

You can also change property values and attributes by selecting the property text itself. In the example on the facing page, you choose the **Select > Area (All) > Property** to select the property text in a bounding box.

Finally, you can select property text by specifying the property name. You choose **Select > By Property > Name**, then specify the property name.

Changing Property Values

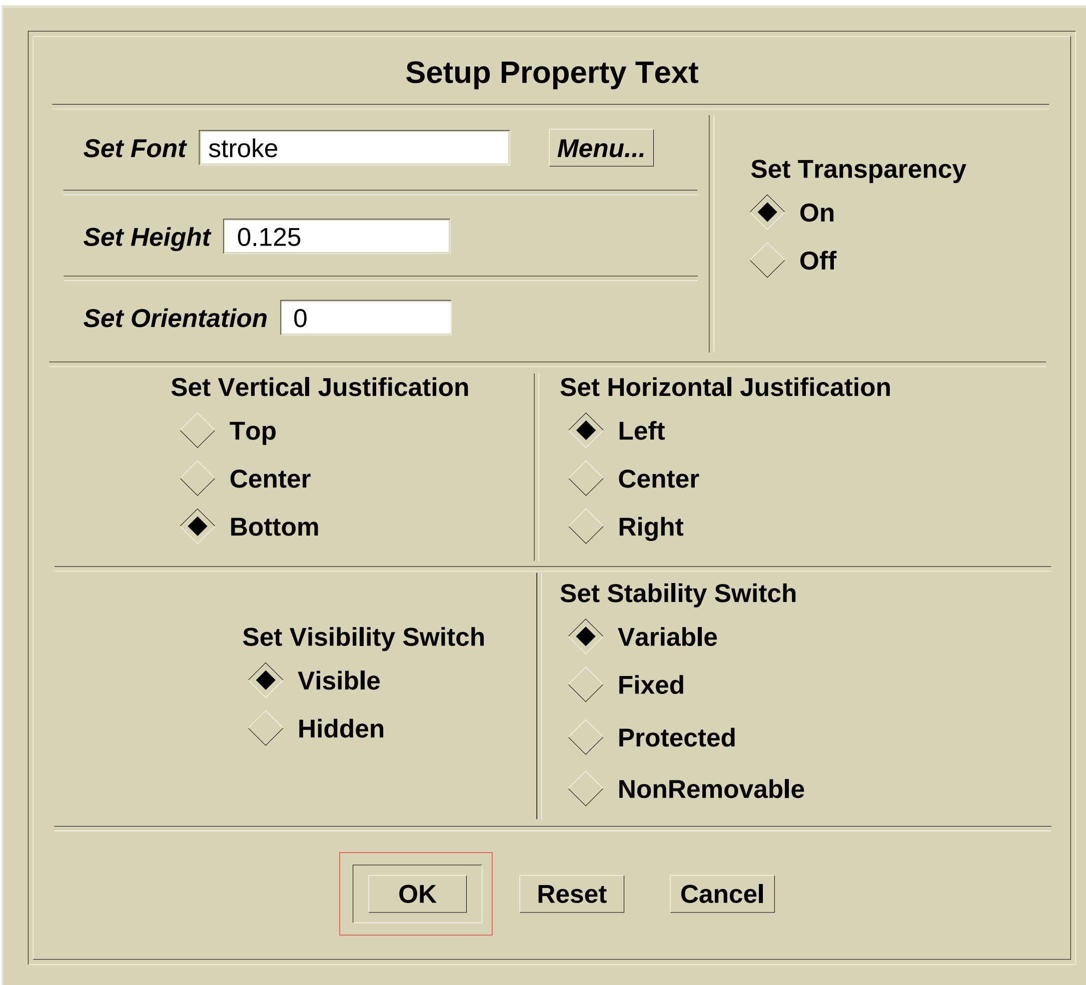
A quick way to change a property value after it is selected in to click the CHANGE VALUE icon or draw the Modify Property Stroke.

Changing Property Attributes

Changing property attributes, such as height, is more easily done by using the popup menu items. Three choices are shown on the left.

Setting Up Property Text Attributes

- **Changes the default values for property attributes**
- **See results next time a property is added**
- **Setup > Property Text displays (schematic version):**



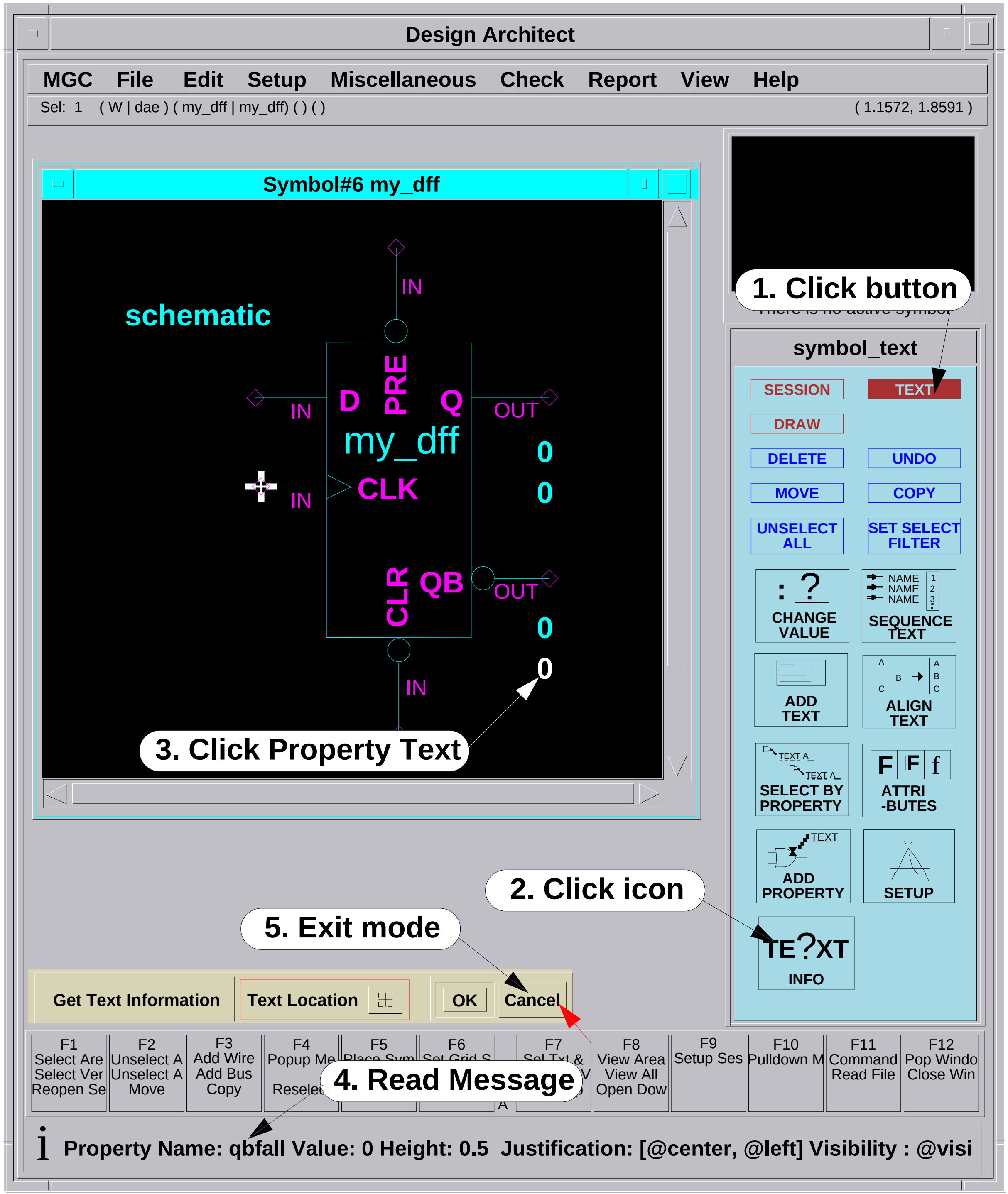
Setting Up Property Text Attributes

The **Setup > Property Text** menu item lets you change property text attributes used when adding properties in the Symbol Editor and the Schematic Editor. After the menu item has been chosen, a Setup Property Text dialog box is displayed. You can change the following attributes:

- **Font**. Name of a registered font family available on your workstation.
- **Height**. A positive, non-zero number that specifies the height of the property text.
- **Orientation**. Value of 0 (horizontal) or 90 (vertical) that specifies the orientation.
- **Transparency**. If text transparency is on, objects under a text string are visible; if text transparency is off, objects under a text string are hidden.
- **Vertical Justification**. Specifies the vertical justification.
- **Horizontal Justification**. Specifies horizontal justification.
- **Visibility**. Specifies whether the values of instance-specific properties are visible or hidden on the schematic sheet.

The results of changing the default values are displayed the next time property text is added.

Quick Report on Property Text



Quick Report on Property Text

A quick way to generate a report on unknown property values is to use the TE?XT icon on the text palette. The procedure is as follows:

1. Click the **TEXT** button on the palette.
2. Click the **TE?XT** icon. The **Get Text Information** prompt bar appears.
3. Click on a piece of property text.
4. Read the message about the text in the message window.
5. Keep clicking on pieces of unknown text until you are satisfied.
6. Click the Cancel button on the prompt bar to exit the **Get Text Information** mode.

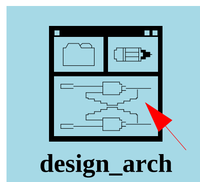
Lab Exercises

Exercise 1: Creating a Symbol

In this lab exercise, you will create the symbol for the schematic sheet that you created in the last module.

Invoke Design Architect

1. Double-click the **design_arch** icon in the Design Manager Tools window
.



(or, type *$MGC_HOME/bin/da* in a shell).

2. Click the Maximize button to fill the screen with Design Architect.

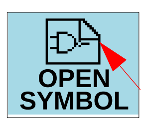
3. Verify the setting of the current working directory.

MGC > Location Map > Set Working Directory:

It should read <*home_directory*>**/training/da_n/card_reader**. If is doesn't, set it to this value and click **OK**; otherwise click **Cancel**.

Open the Symbol Editor on the my_dff Component

1. Click the **OPEN SYMBOL** icon.



The Open Symbol dialog box is displayed in the DA session area.

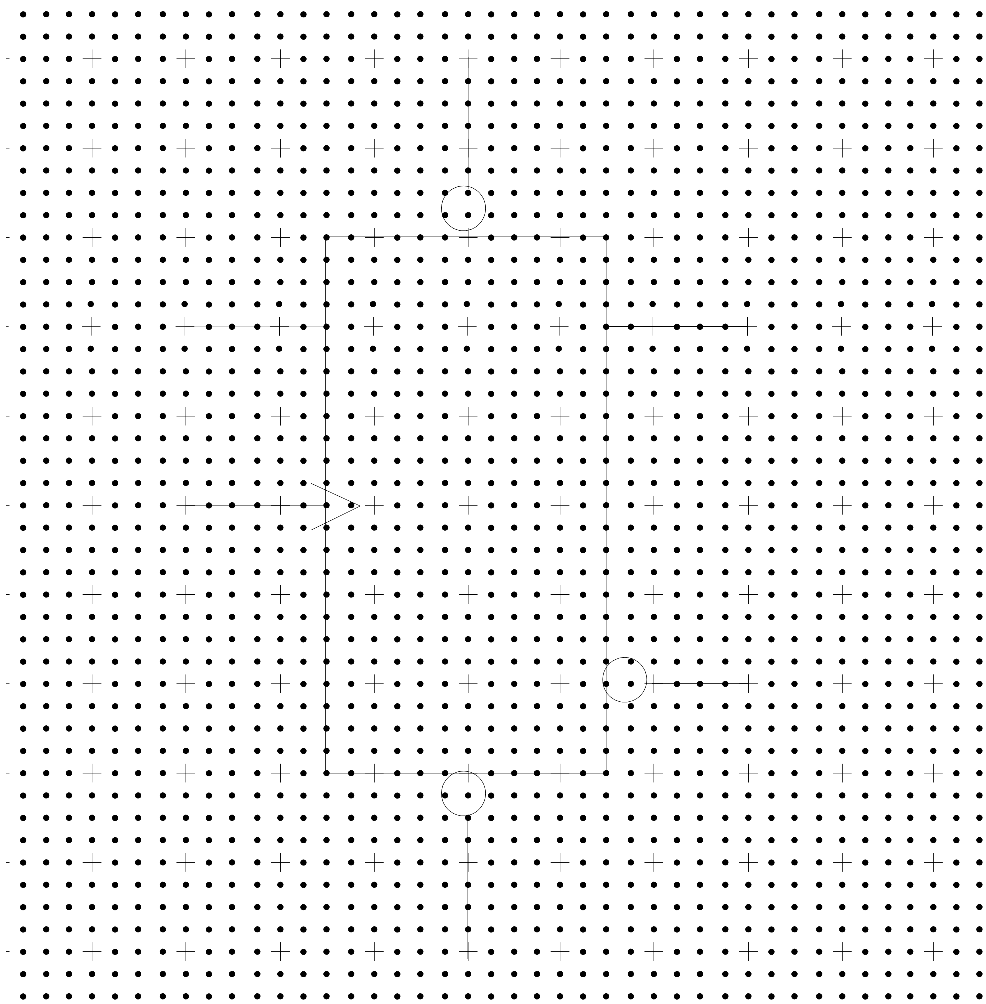
2. Click the Navigator button, navigate to and select the **my_dff** component, then click OK.

Design Architect opens a new (blank) Symbol Editor window on the **my_dff** component. The symbol is named my_dff by default, because you did not specify an alternate name in the dialog box.

3. Click the Maximize button on the Symbol Editor window.

Creating a Symbol Body

Use the following figure to determine the size of the rectangle, circles, and lines that you will use to create a symbol body. Place the symbol graphics <u>exactly as shown</u> in the figure.



Create a Rectangle

1. Click on the **ADD RECTANGLE** icon. Press, but do not release, the Select mouse button at the location of the first corner of the rectangle. Drag the mouse button to the opposite corner and release.

2. Draw a stroke.

3. If the rectangle is not exactly as shown, click the STRETCH icon, click the mouse pointer on an edge you want to move, then move the pointer to the new edge position and click.

Create Circles

1. Click on the **ADD CIRCLE** icon. Press, but do not release the Select mouse button at the center location of the circle. Drag the mouse button to the perimeter of the circle and release it.

2. With the circle still selected, draw a stroke to make a copy and place it in the second location for a circle.

3. Repeat the above step for the third circle, then draw a stroke to unselect all.

Create Lines

1. Click on the **ADD POLYLINE** icon. Draw the upper three whiskers by clicking the Select mouse button at one end of the whisker and double clicking that the other end. After the first line is drawn, use the copy stroke to create and place the other two lines.

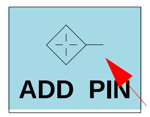
2. Perform this action for as many times as there are whiskers. Make sure that you end each line on a “+” pin grid. Pins will be attached to the end of these lines and pins must snap to these grid points.

3. Draw the “>” polyline by clicking once at the starting point, once at the tip, and twice at the end.

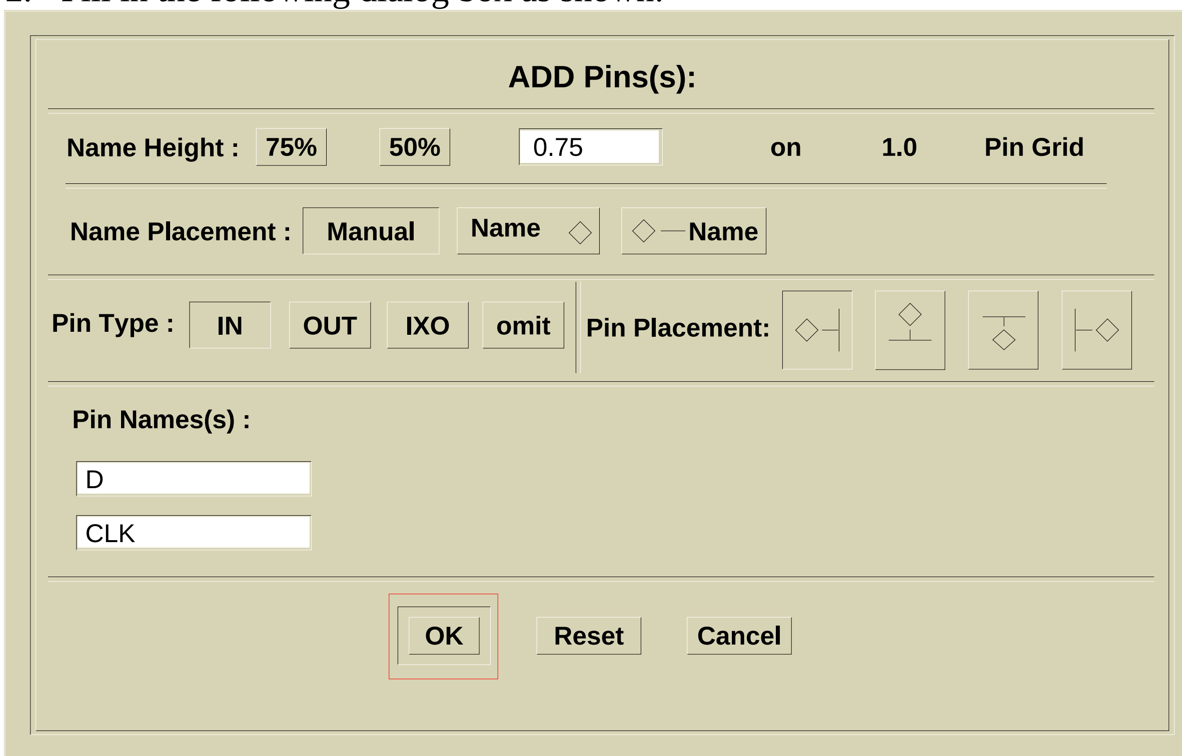
4. Unselect All with a stroke

Add Pins to the Symbol Body.

1. Click on the **ADD PIN** icon.



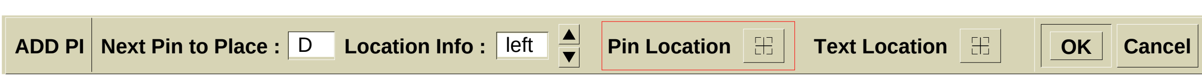
2. Fill in the following dialog box as shown:



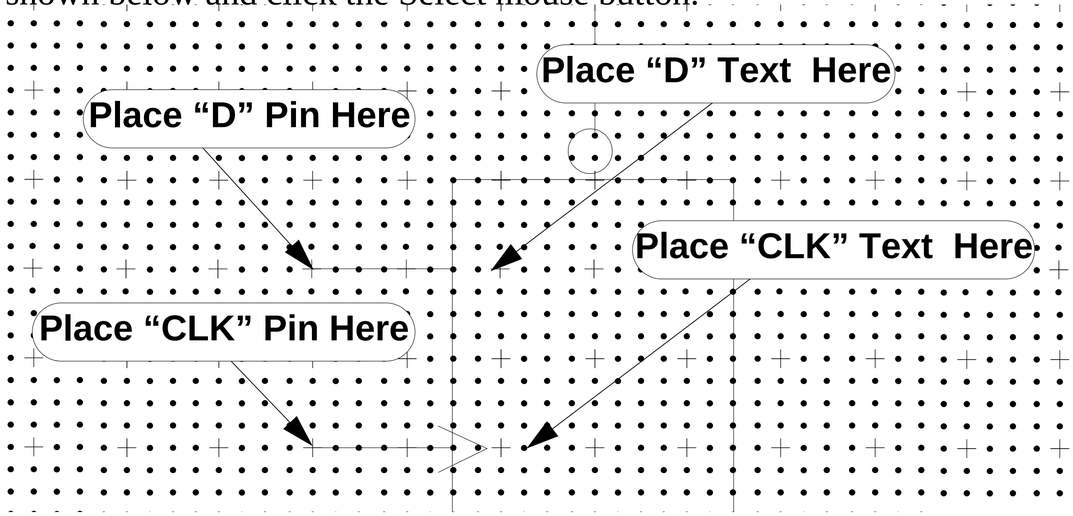
(Make sure you click the **Manual** button.)

3. Click **OK** or press **RETURN**.

The dialog box disappears and the Add Pin prompt bar is displayed as shown below.



4. Move the cursor to the desired location of the pin at the end of the line as shown below and click the Select mouse button.



The location cursor moves to the Text Location text entry box. You will see text “D” appear near the pin location.

5. Move the cursor to the desired location of the text, as shown above, and click the Select mouse button.

A diamond-shaped object representing a pin is placed at the end of the line. The Pin Name (D) is positioned to the right of the pin, and the PINTYPE property value (IN) is displayed.

The prompt bar disappears and another Add Pin prompt bar is displayed. The **Next Pin to Place** entry box is filled in with “**CLK**”. The **Location Info** entry box is filled in with “**left**”. The location cursor is on the **Pin Location** prompt text.

6. Move the cursor to the desired location of the pin at the end of the line, as shown above and click the Select mouse button.

The location cursor moves to the **Text Location** entry box. You will see text “CLK” appear near the pin location.

7. Move the cursor to the desired location of the text, as shown above, then click the Select mouse button.

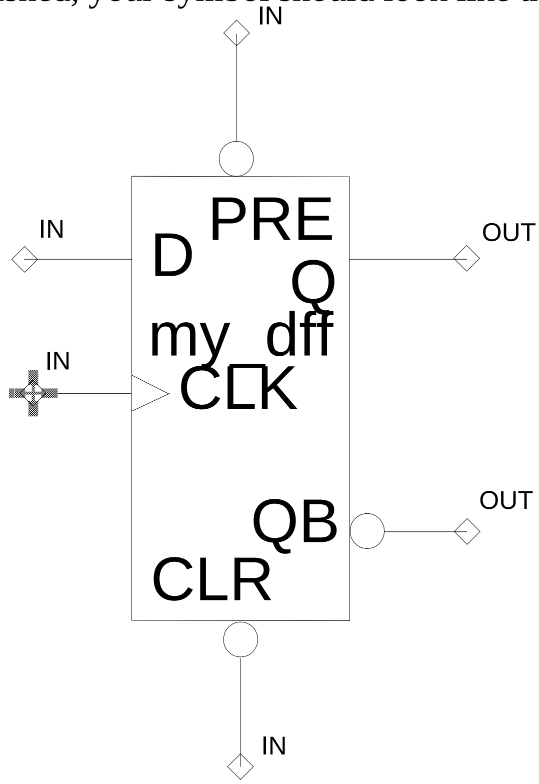
A diamond-shaped object representing a pin is placed at the end of the line. The Pin Name (CLK) is positioned to the right of the pin, and the Pintype property value (IN) is displayed.

8. Continue adding the Pin and Pintype properties using the following information.

Pin Names	PinType	Pin Placement
PRE	IN	Top
CLR	IN	Bottom
QB	OUT	Right
Q	OUT	Right

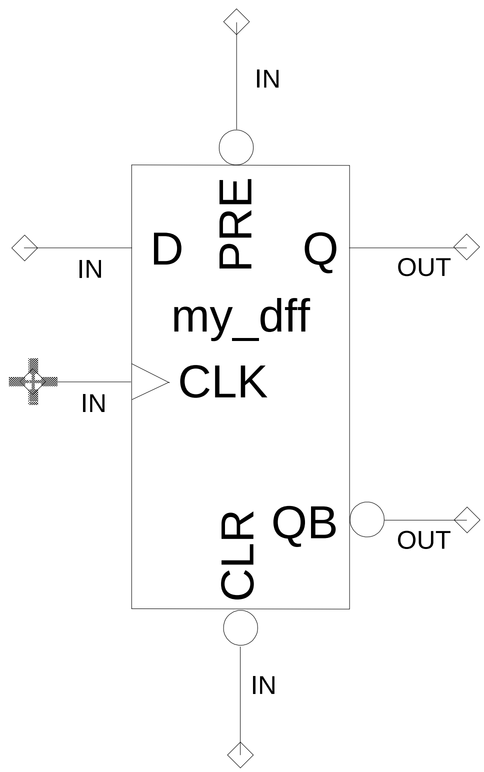
9. Add **my_dff** symbol text by clicking on the TEXT palette button and the ADD TEXT palette icon. Enter the text into the prompt bar entry box.

10. When you are finished, your symbol should look like the following:

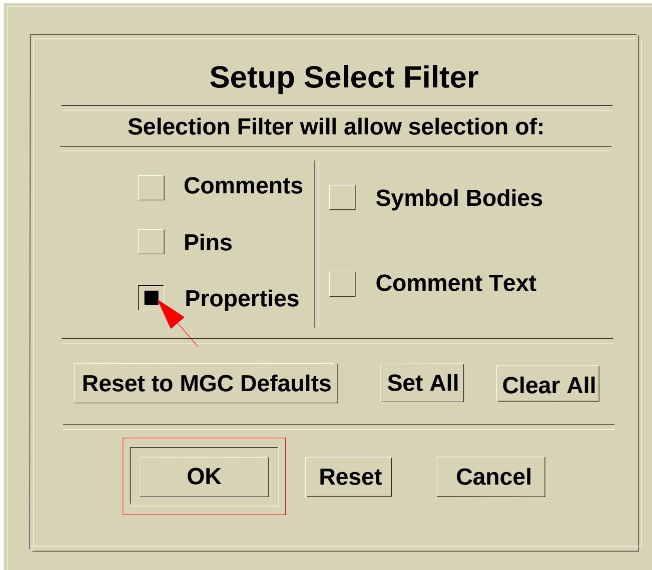


Resizing, Rotating, and Repositioning Property Text

Next, you will resize, rotate, and reposition property text to make you symbol look like the following illustration:



1. Set the Selection Filter for Property Text only by drawing a stroke.



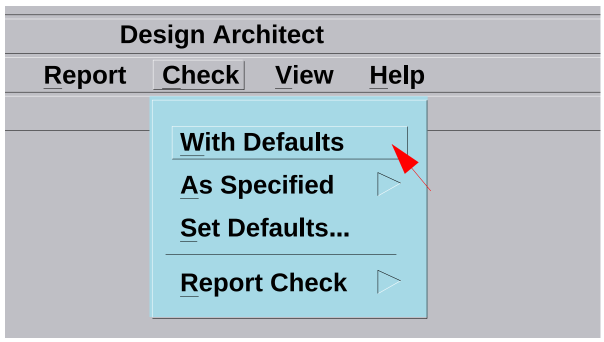
2. Click the **Clear All** button, then the **Properties** button
3. Draw a stroke to execute the form.
4. Now draw a selection box around the property text on the interior of the symbol rectangle. All the Pin property text is selected.
5. Place the mouse pointer on the symbol text my_dff and tap the F1 function key. The symbol text should be selected and added to the selection group.
6. Press the Right mouse button and choose the **Change Height** > **0.5 x Pin Spacing.** The text is reduced in size to 1/2 the pin grid spacing.
7. Draw a “U” stroke to unselect all.
8. Click on the “PRE” property text and draw a stroke. The text is rotated 90 degrees.
9. Draw a “shark fin” stroke , then move the PRE text to the position shown in the symbol illustration on the previous page.
10. Draw a “U” stroke to unselect all.

11. Repeat the above procedure to rotate and move the CLR text.

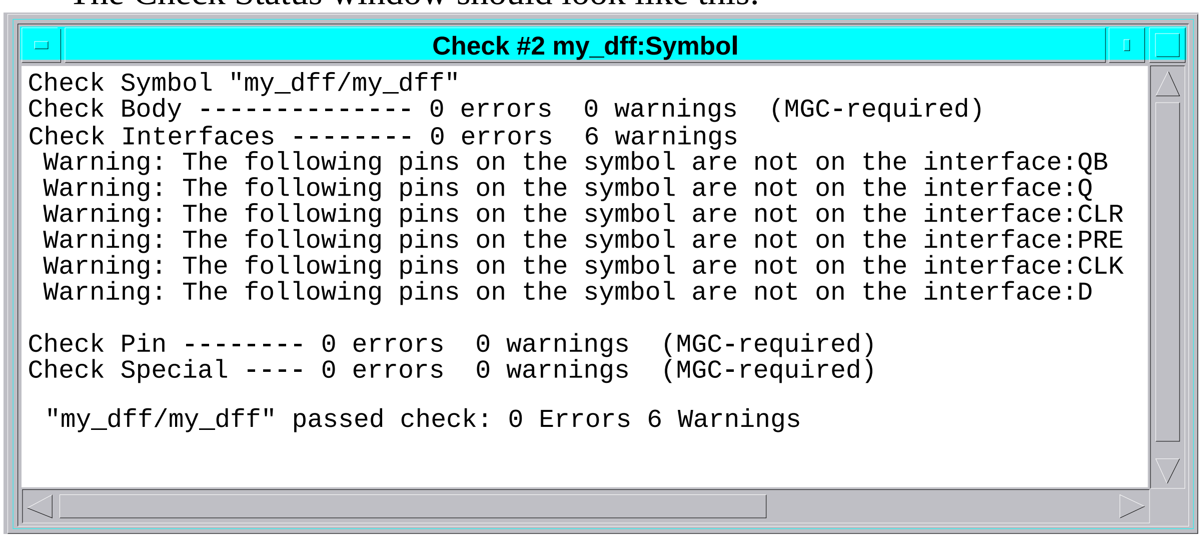
12. Individually select and move the remaining pin and pintype property text to the positions shown in the symbol diagram.

Check the Symbol

1. Using the default checking levels, check your symbols for errors and warnings. Choose the following pulldown menu item:



The Check Status window should look like this:



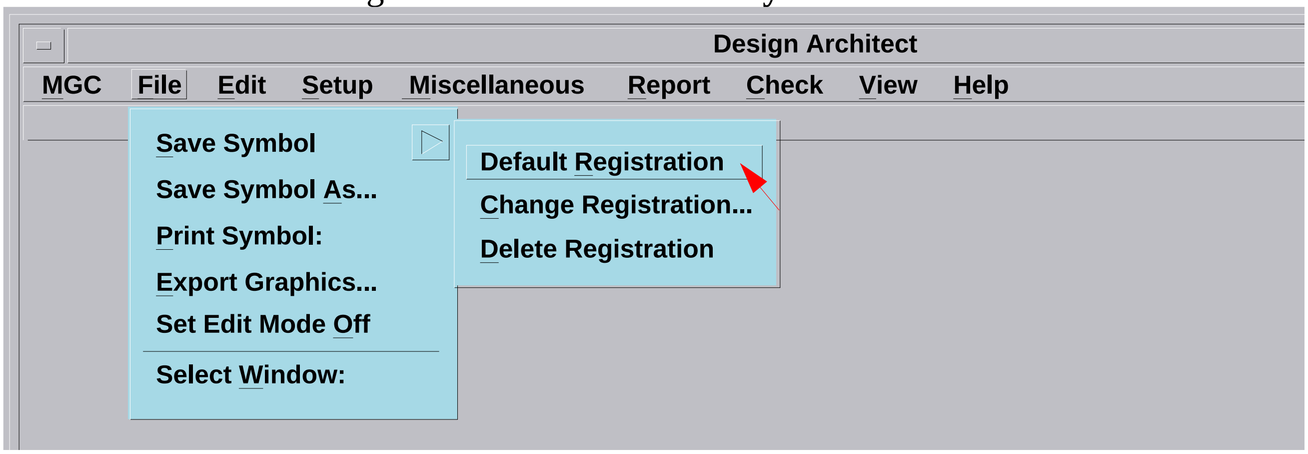
Notice that the Check Status window displays several warnings. These warnings are always given when pins are changed or added to a symbol. They are normal when saving a symbol for the first time because the pins list in the component interface is changed from zero pins to a number matching the new pins on the symbol.

a. If your check results in errors, you can identify handle names (like I$3) by clicking the handle name in the Check Status window. The object associated with that handle name is selected on the sheet.

b. Close the Check Status window. The symbol window is automatically reactivated.

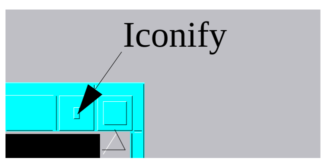
Save the Symbol

1. Choose the following menu item: **File** > **Save Symbol**



Iconify the Symbol Window

1. Iconify the Symbol Window



Exercise 2: Adding Properties to a Schematic

In this lab exercise, you will modify properties on the **my_dff** schematic sheet that you created in the previous module.

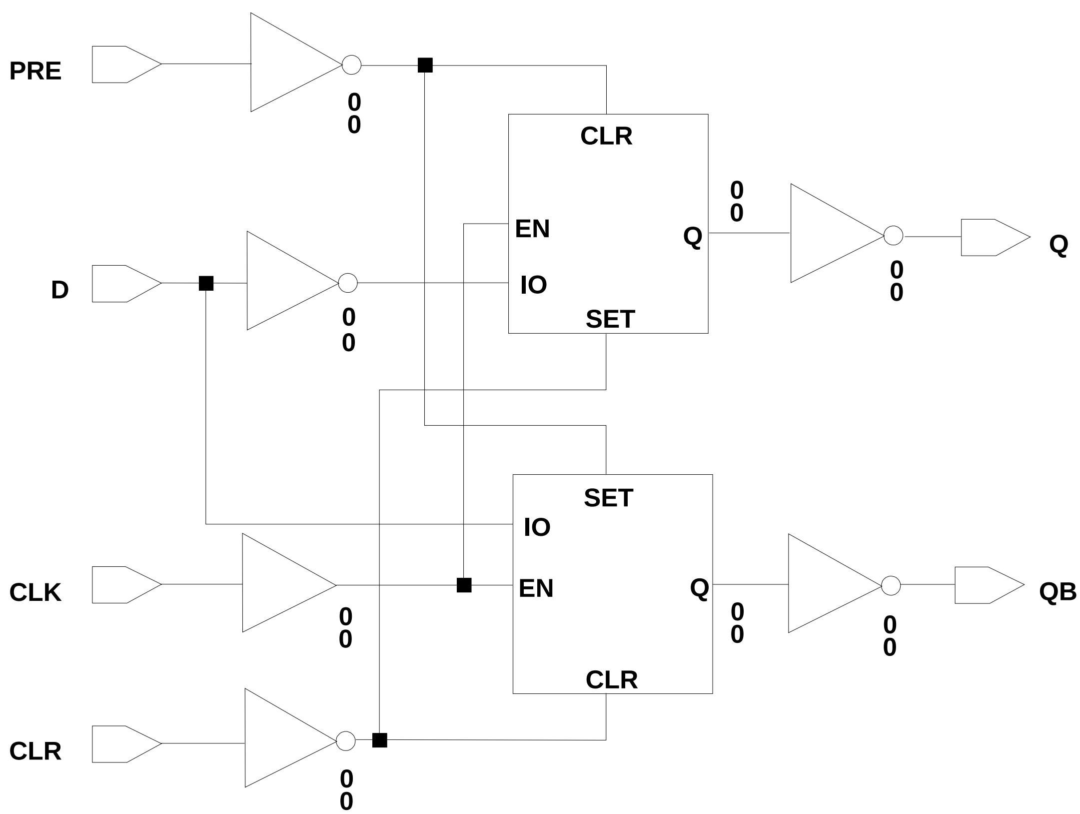
Open the my_dff Schematic

1. Click on the DA Session Window so that the Session palette appears.

2. Open a Schematic window on the **my_dff** component.

3. Maximize the Schematic Window

4. Draw the  stroke to view the entire sheet. The schematic should look like the following:



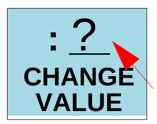
Change the Values of the Rise and Fall Properties

Notice that the current values of the Rise and Fall properties on the **inv** output instances are set to 0. In this exercise, you will change them to the expressions; (QRISE) and (QFALL) for the top output inverter; (QBRISE) and (QBFALL) for the bottom output inverter.

Use the following method to change the property values.

1. Set the selection filter to **Properties** (only) and with the Select mouse button draw a bounding box around the Rise and Fall properties of the output **inv** instances. The four 0s should be highlighted, and the select count should be 4.
2. Click the 

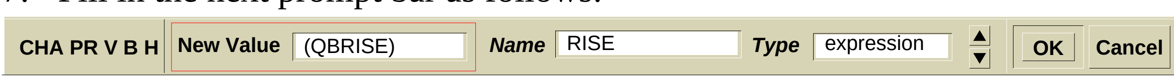
palette button, then click the CHANGE VALUE icon.



3. Fill in the prompt bar as follows:

4. Click **OK**. The top property value is changed, the next to the top property value is selected, and prompt bar reappears.
5. Fill in the prompt bar as follows:

6. Click **OK**
7. Fill in the next prompt bar as follows:



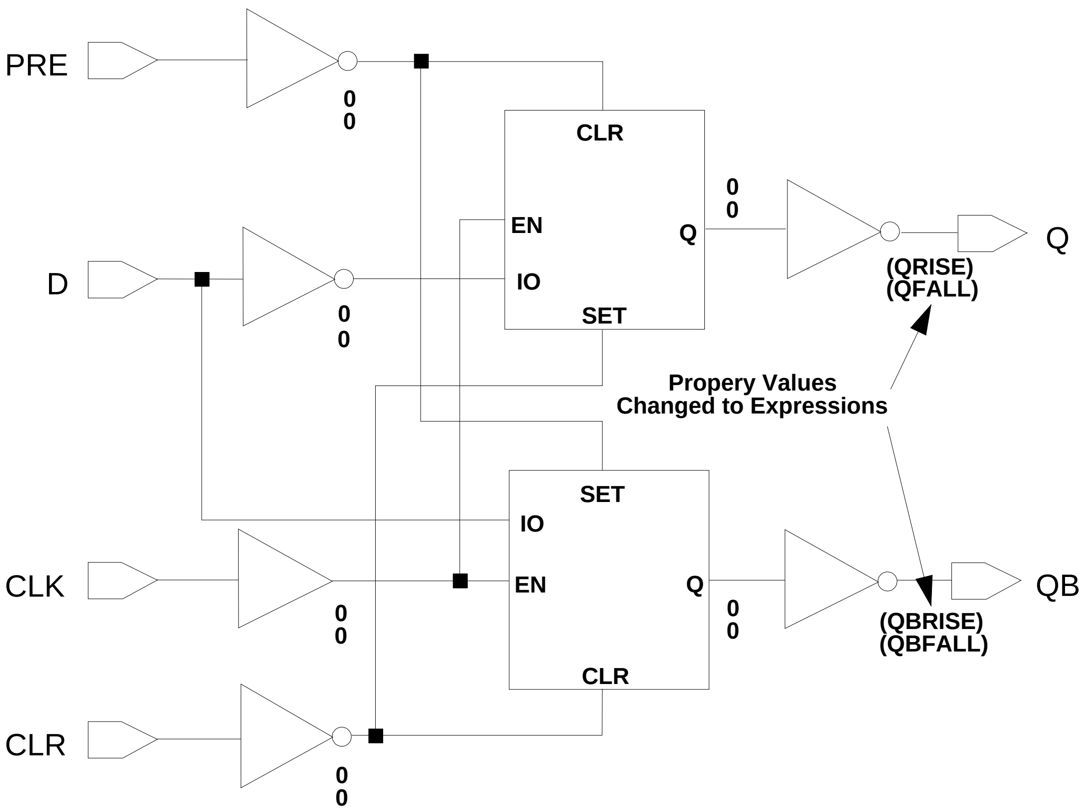
8. Click **OK.**

9. Fill in the next prompt bar as follows:

10. Click **OK**.

11. Unselect All before going on to the next step.

The results of this change should appear as follows:



Check the Sheet

1. Using the default checking level, check your sheet for possible errors or warnings:

Check > Sheet > With Defaults

Your Check Status window should look similar to the following

```
Check #3 my_dff:Sheet

Check Sheet "my_dff/schematic/sheet1"
Check SymbolPins -------- 0 errors  0 warnings  (MGC-required)
Check Overlap ----------- 0 errors  0 warnings
Check NotDots ----------- 0 errors  0 warnings
Check Closedots --------- 0 errors  0 warnings
Check Dangle ----------- 0 errors  0 warnings
Check INIT Props -------- 0 errors  0 warnings
Check Owner ------------- 0 errors  0 warnings  (MGC-required)
Check Instance ---------- 0 errors  4 warnings  (MGC-required)
  Warning: Unable to evaluate property "fall" on I$2
       Unable to resolve expression symbol qbfall
  Warning: Unable to evaluate property "rise" on I$2
       Unable to resolve expression symbol qbrise
  Warning: Unable to evaluate property "fall" on I$3
       Unable to resolve expression symbol qrise
  Warning: Unable to evaluate property "rise" on I$3
       Unable to resolve expression symbol qbfall

Check Special ----------- 0 errors  0 warnings  (MGC-required)
Check Net --------------- 0 errors  0 warnings  (MGC-required)
Check Frame ------------- 0 errors  0 warnings  (MGC-required)

"my_dff/schematic/sheet" passed check: 0 errors 4 warnings
```

Notice that you received warning messages about the current values of the Rise and Fall properties. These messages indicate that the Rise and Fall property values are expressions and can't be evaluated at this time. This is normal.

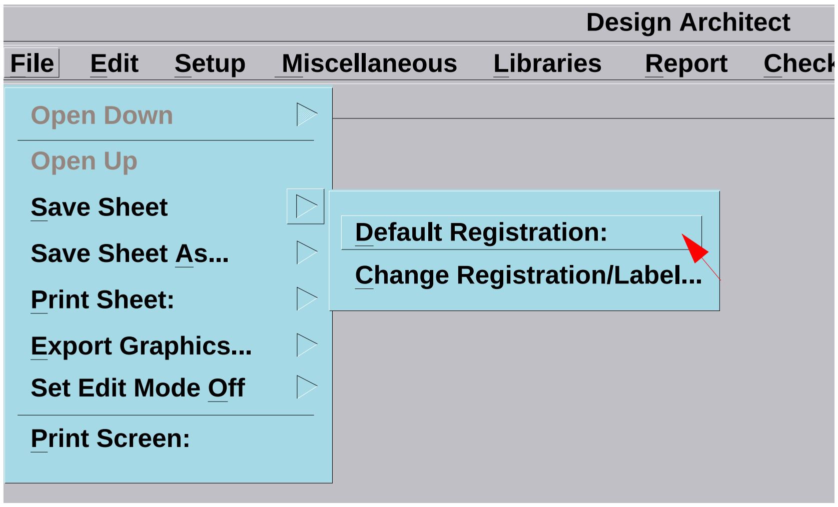
If your check results in errors, you can identify handle names (like I$2) by clicking the handle name in the Check Status window. The object associated with that handle name is selected on the sheet.

2. Close the Check Status window with a → stroke. The schematic window is automatically reactivated.

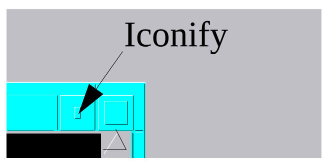
Save the Sheet

1. Save the schematic to disk by executing **File** > **Save Sheet** > **Default Registration** from the pulldown menu.

.



2. Iconify the Schematic Window

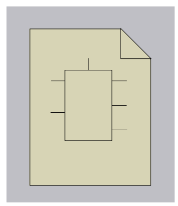


Exercise 3: Adding Properties to a Symbol

In this exercise, you will complete the symbol that you began in Exercise 1 by adding additional properties to the symbol body.

Expand the Symbol Window

1. At the end of Exercise 1, you iconified the Symbol Editor Window for **my_dff**. Find this icon in the DA Session window and double-click on it. The window

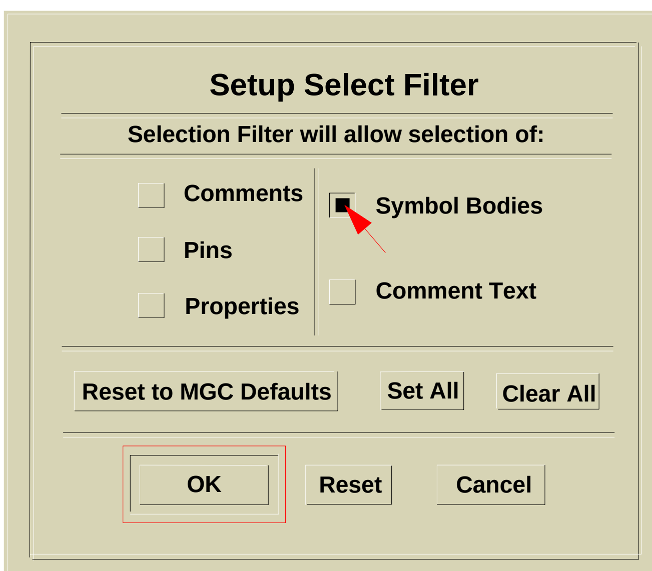


should expand to its original size.

2. Draw an Unselect All stroke to make sure that all objects are unselected.

Set the Select Filter

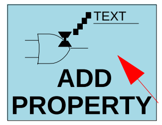
1. Set the selection filter to Symbol Bodies



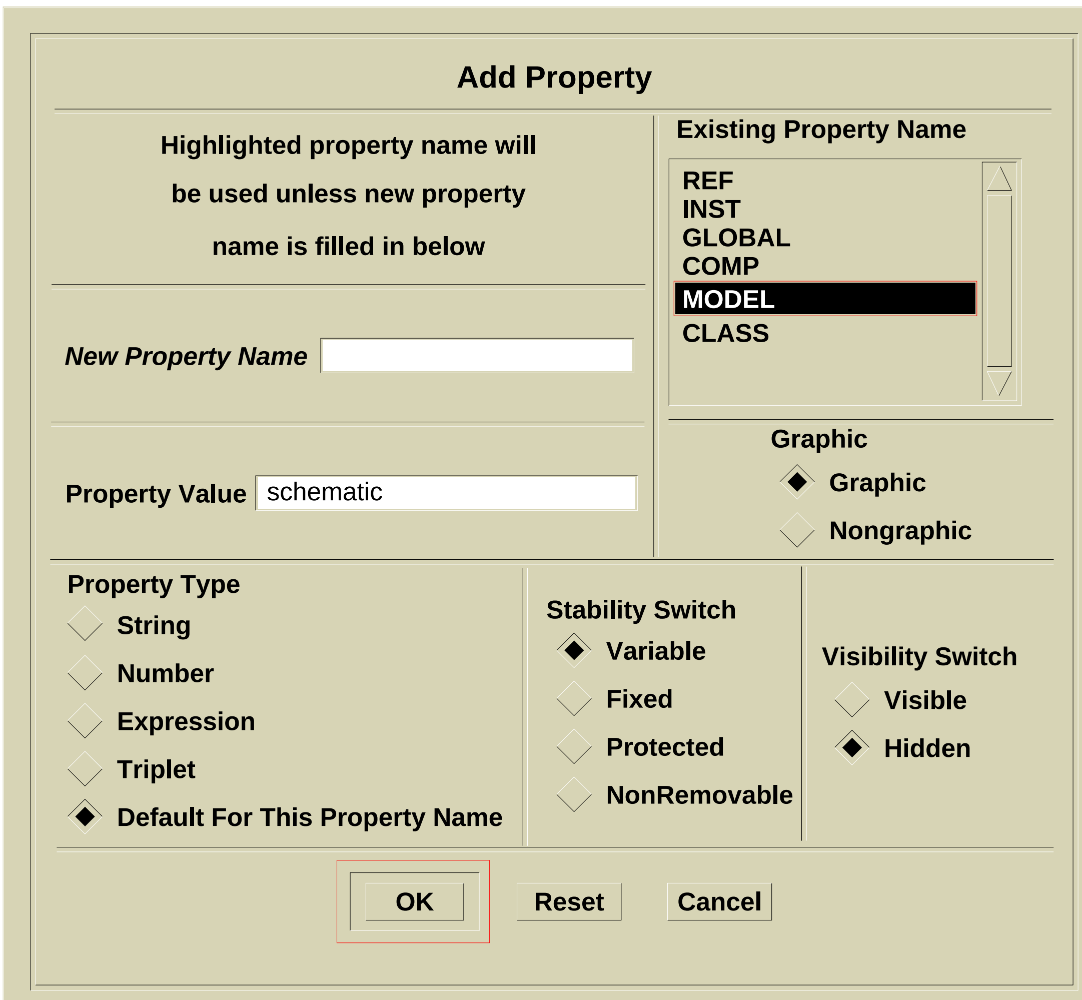
Add a MODEL Property

a. Click on the symbol body to select it.

b. Add a MODEL property to symbol body by clicking on the ADD PROPERTY icon.



Fill in dialog box as shown in the following illustration:.



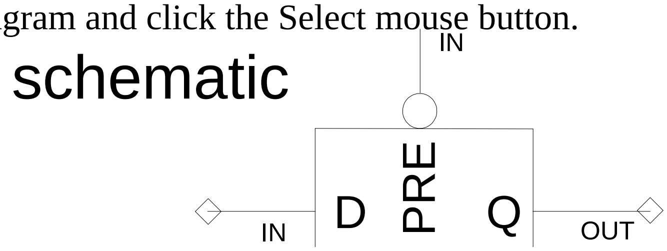
Notice that the Existing Property Name list contains names of properties that are typically owned by the selected object. If the property that you want to add does not exist in this list, type the property name in the ***New Property Name*** entry box.

In this case, click on the MODEL entry in the **Existing Property Name** scrolling window. Now you only need to change the **Property Type** when you are adding a property that does not exist in the list

Make sure that you don't forget to change the Visibility Switch to **Hidden**.

c. Click **OK.** The **Add Property** prompt bar appears.

d. Move the ghost image of the text to the location shown in the symbol diagram and click the Select mouse button.



The text "schematic" appears next to the symbol body.

Why is the "schematic" text visible if you specified that it be Hidden?

____

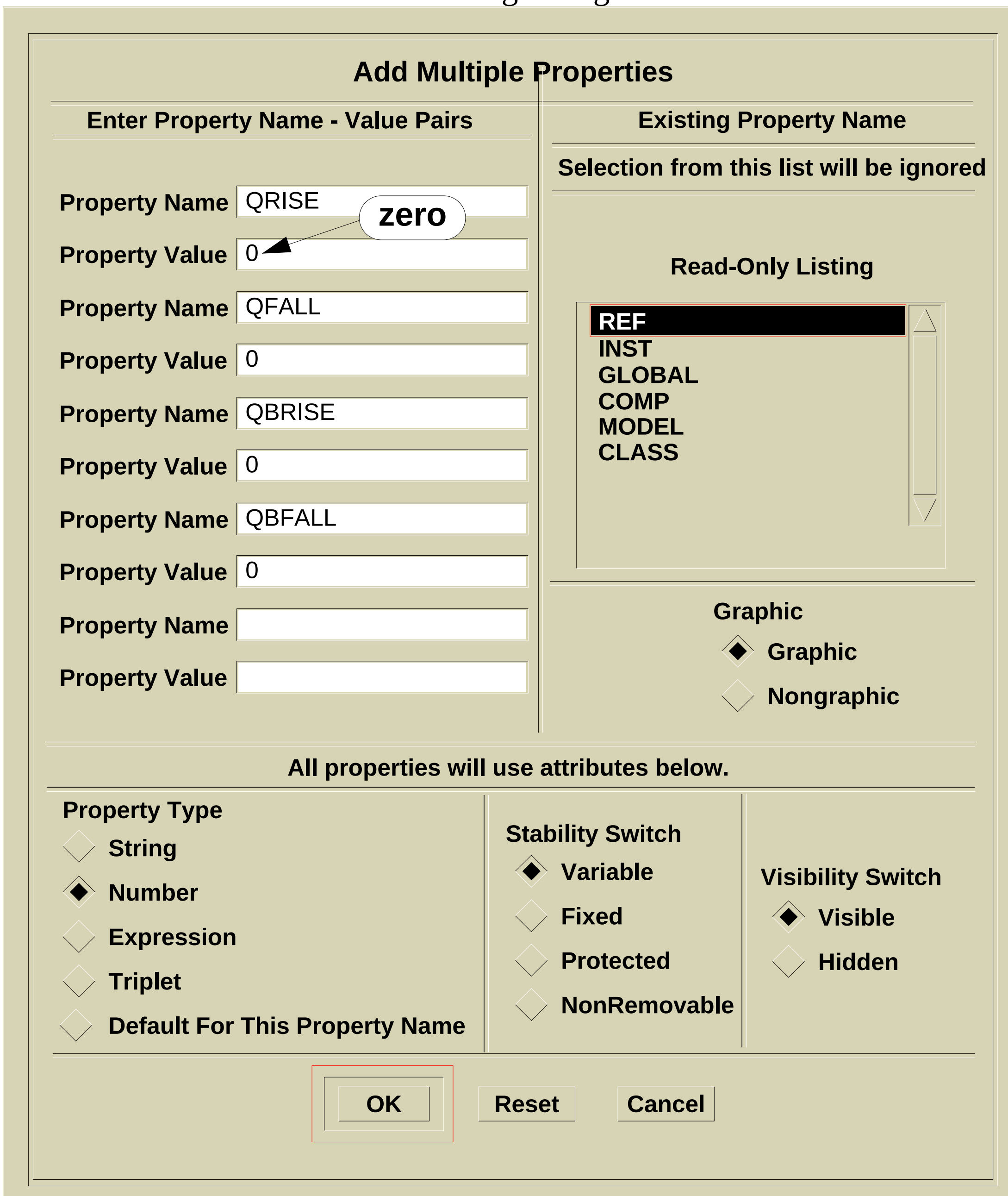
Add Parameters as Property Values

Parameters are defined as values that are passed to variables in an expression from an outside source. In the last exercise, you defined four single variable expressions for the RISE and FALL properties on the Q and QB output of the schematic. In this exercise, you are going to attach property names to the symbol body that match the variable names in the expressions. When the **my_dff** schematic gets "evaluated" by a downstream analysis tool, the values assigned to the properties that are attached to the **my_dff** symbol will be passed to the expressions on the sheet so that the rise and fall times can be resolved to constant values. Add the properties to the symbol body as follows:

1. With the symbol body selected, choose the following popup menu item:

Properties > Add > Multiple Properties...

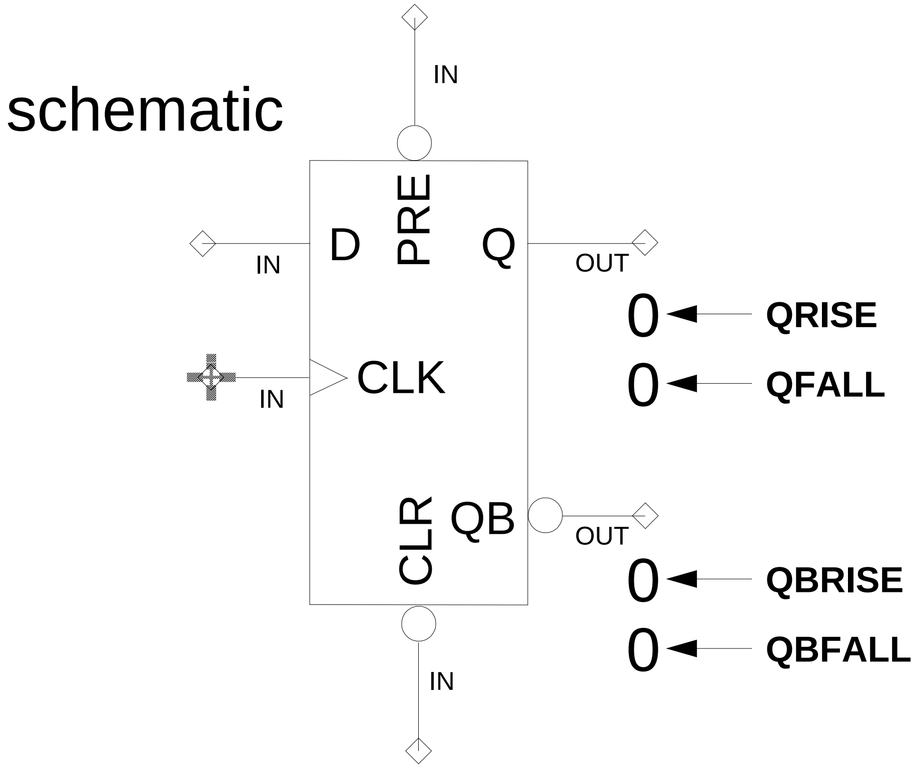
2. Enter the values as shown in the following dialog box:



3. Click the **OK**.

The dialog box disappears and the Add Property prompt bar is displayed.

4. Position the four new property values as shown in the following figure:



5. Unselect All.

Change the Height of the Newly-Added Property Text

Change the height of the new property text as follows:

1. Place the mouse pointer over each piece of new property text and tap the F1 function key. This allows you to bypass the select filter and select whatever object is underneath the pointer.

2. Press the Right mouse button and choose the **Change Height > 0.5 x Pin Spacing.** The text is reduced in size to 1/2 the pin grid spacing.

3. Draw a “U” stroke  to unselect all.

Verify that the Property Values are Placed Correctly

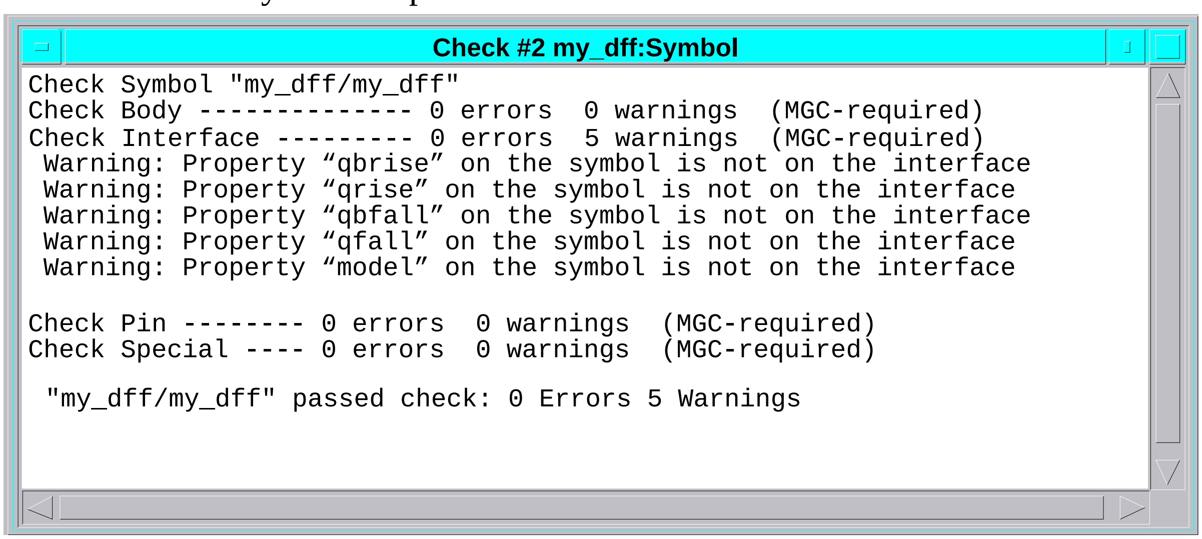
Since all four property values are the same value(zero), you can't tell just by looking at them that they are placed correctly. Do a quick check as follows:

1. Click the TE?XT icon:
2. Select the top piece of property text that you just added and read the message in the message window. The message should verify that the property is “qrise”.
3. Select the property text that is second from the top. The message should verify that the property is “qfall”.
4. Select the next two property values, one-at-a-time, and verify that they are qbrise and qbfall, respectively.
5. Click the **Cancel** button on the prompt bar to exit TE?XT mode.

Check the Symbol

1. Check your symbol for errors or warnings. **Check > With Defaults**

   The following Check Status window is displayed showing the results of the symbol check. Notice that the check window displays several warnings. These warnings indicate that the symbol has changed, but the component interface table has not yet been updated. This is normal.

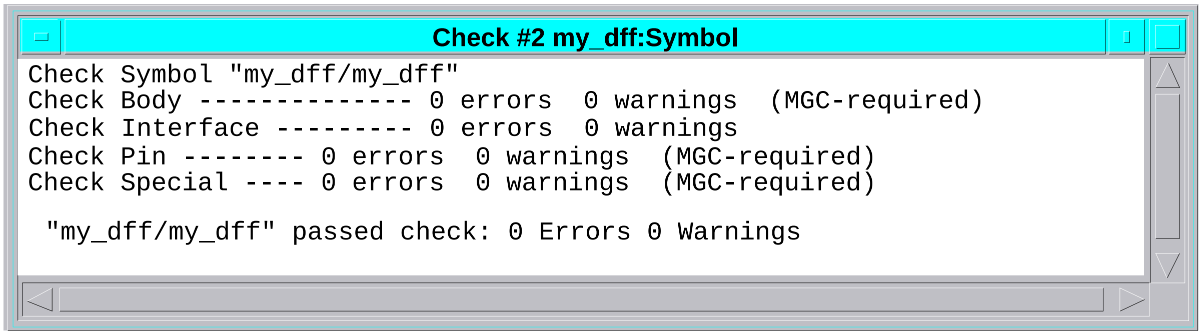


a. Close the Check Status window. The symbol window is automatically reactivated.

Save the Symbol

1. Select the following pulldown menu item: **File** > **Save Symbol**

2. Check the symbol again. The Check window is shown below.



Notice this time there are no errors or warnings. When the symbol was saved, the component interface table was updated to match the new information on the symbol.

3. Close the Check Status window, then close the Symbol window.

Resave the my_dff Schematic

When you resave a symbol that has changed, the component interface table is updated. Because the ports on the schematic may no longer match the pins on the symbol, the schematic model is automatically marked “Not-Valid”. In order to run the Validation routine and verify that the schematic ports still match the pins on the symbol, you must check and resave the sheet.

1. Double click on the Schematic window icon in the DA Session Window.

2. From the pulldown menu choose: **Check** > **Sheet** > **With Defaults.**

3. Close the Check Window with a stroke.

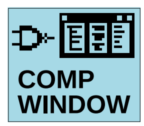
4. From the pulldown menu choose: **File** > **Save Sheet> Default Registration**.

5. Close the Schematic Window

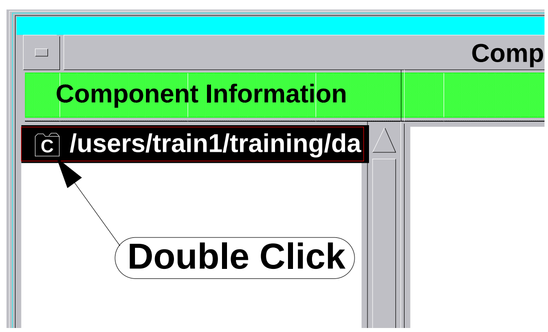
Exercise 4: Browsing the my_dff Component in the Component Window

It is helpful to understand the internal structure and content your design. You will be observing and changing this structure throughout this training course. The following exercise will give you more practice using the Component window.

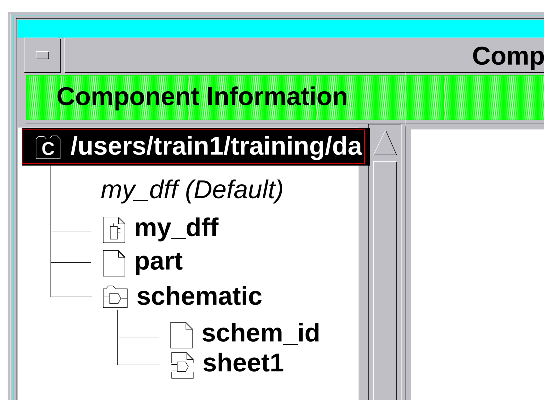
1. Click on the COMP WINDOW icon:



2. Select the **my_dff** component in the Navigator window and click **OK:**

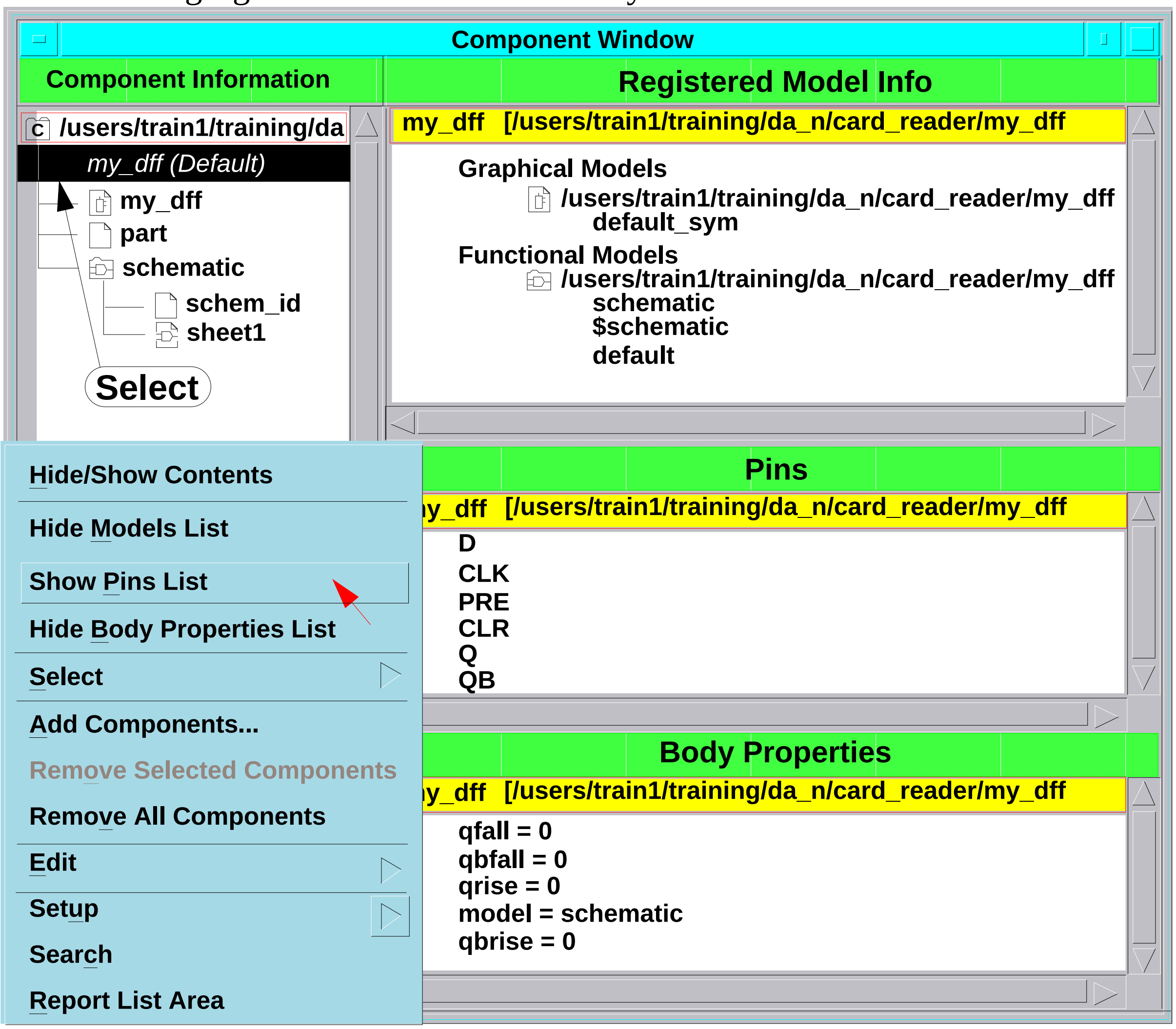


3. Double click on the component icon, then on the schematic icon:



Notice that the component now contains three objects, the **part** object, the **schematic**, and the new **symbol** object

4. Select the component interface name ***my_dff (Default)***, then from the Component Information window popup menu choose **Show Pin List**. The following figure shows the information you should see in the window:



Notice that the symbol model is now listed in the Registered Model Info window, the symbol pins that you added to the symbol are listed in the Pins window, and the symbol body properties that you added to the symbol graphics are listed in the Body Properties window.

5. Close the Component Window.

End of Lab Exercises

This concludes the lab exercises for this module.

Module 4
For Continued Learning . . .

Mentor Graphics is dedicated to helping customers be successful by our offering a variety of training products. Training is available in different formats, including Getting Started Training Workbooks, Personal Learning Programs, and Instructor-Led Workshops. Many training workbooks are available online in INFORM, as well as in hardcopy form. For continued learning, refer to the following products.

Getting Started Training Workbooks

Getting Started with Falcon Framework

Personal Learning Programs

Design Architect Training Workbook

Instructor-Led Workshops

To determine the workshops you should attend, consult your Mentor Graphics account manager or application engineer. You may also contact any of our training instructors at Mentor Graphics Training Centers.