DesignWorks 3

Schematic Drawing Tool

for Windows

Reference Manual

April 15, 1997

206 - 360 Edworthy Way
New Westminster, B.C., Canada, V3L 5G2
(604) 522-6200

Table of Contents

Chapter I - Introduction

DesignWorks™ 3 is a powerful schematic editing tool with features designed to allow it to form the core of your electronics design system.

The package covered by this manual includes a number of tools interconnected by a common framework:

Schematic Handles all basic schematic drawing functions.

DevEditor Provides device symbol creation and editing functions.

LibIO Handles storage, retrieval and maintenance of device library files.

Find Locates objects by name or attribute value.

ErrorFind Checks the design for possible errors.

Report Generates netlists and bills of materials in standard or user-definable formats.

The functions of each of these tools are described in detail in this manual. Other optional packages such as Digital Simulation and EDIF Schematic Translation are described in separate manuals.

Where to Start

We suggest you ease yourself into the world of schematic editing with DesignWorks by taking the following steps:

STEP 1 Install the package using the procedures outlined in Chapter II - Getting Started and any separate release notes supplied with the package.

STEP 2 If you are using DesignWorks for the first time, work first through the separate demonstration manual supplied with the package. It provides step-by-step instructions for basic schematic editing.

STEP 3 Refer to Chapter IV - Schematic Editing in this manual for background on basic editing techniques.

STEP 4 Before you start a major design, refer to "Chapter VIII - Before Starting a Major Design". This will help you make decisions that will save time later!

Chapter Organization

Each chapter is divided into two parts:

- the first sections give background information on the features covered in that chapter.
- the last section gives specific "how to" procedures for using the features.

The DesignWorks/Schematic manual is divided into the following chapters:

Chapter II - Getting Started	Covers installation and initial startup.
Chapter III - Schematic User Interface	Provides detailed information on the use of windows, menus, keyboard commands and other user controls and displays.
Chapter IV - Schematic Editing	Discusses the structure of a circuit diagram and how to create one.
Chapter V - Advanced Schematic Editing	Provides information on advanced features like device packaging, power and ground connections, busses, text variables, etc.
Chapter VI - Attributes	Discusses the text attribute features of the package in detail.
Chapter VII - Hierarchical Design	Covers the hierarchical design features of DesignWorks.
Chapter VIII - Before Starting a Major Design	Discusses a number of considerations that may affect how you organize a design.
Chapter IX- Schematic Menu Reference	Describes menu commands in detail.
Chapter X - Device Symbol Editing	Provides information on creating and editing device symbols.
Chapter XI - Report Generation	Reviews the use of standard netlist formats and provides details on creating custom formats.
Chapter XII - Find and ErrorFind Tools	Describes the Find and ErrorFind tools, used to locate devices and signals.
Chapter XIII - Browser Tool	Describes the Browser, a spreadsheet-style way of editing circuit text information.
Chapter XIV - Guide to Libraries	Describes the symbol standards used in the DesignWorks libraries.

DesignWorks Features

Hierarchy

DesignWorks implements a true hierarchical design structure. This means that "device" symbols appearing in a circuit can actually represent another, nested circuit. This sub-circuit can be opened and edited at any time in separate circuit window.

DesignWorks has three hierarchy modes to address different design situations:

- "Flat" Mode - This is the simplest mode and is intended for small designs or those imported from older versions of DesignWorks.
- "Physical" Mode - This mode allows full use of hierarchical design, while still allowing data associated with individual device instances to be stored at any point in a design. It is intended for larger board-level or FPGA designs where interactive simulation is used or PCB layout information is to be kept with the design.
- "Pure" Mode - This mode stores only a single set of internal data for each type of sub-circuit in a design. This allows very large designs to be represented in a small amount of memory, but does not allow interactive simulation or physical layout data to be associated with the design. It is intended for VLSI or other large-scale design applications.

See Chapter VII - Hierarchical Design for more information.

Attributes

Each design has associated with it an "attribute definition table" which lists all the fields which will be used in the design. Fields can be associated with any device, signal or pin or with the design itself. More than 50 fields are predefined for a variety of applications, and you can add more at any time. Once a field name is defined, its name is chosen from a list of available fields when entering data. This greatly reduces that chance of entry error.

Other attribute features include:

- Attributes can be displayed at any position on the diagram, independently of the name.
- The same attribute data can be displayed in multiple locations.
- All text displayed on the diagram can be edited right in place.
- Attribute text can be rotated in 90° increments.
- Each field can contain up to 32,000 characters of data per object.
- Any text characters and even binary data is valid in an attribute field.

See Chapter VI - Attributes for more information.

Sheet Templates

DesignWorks allows all aspects of sheet sizes, borders, grid layouts, title blocks and text annotations to be customized. The package has no built-in knowledge of standard sheet sizes, but instead imports all information from "template files", which are actually just empty design files. Text and graphics can be placed on a sheet and marked as "border" items. They will then be automatically copied to each new page in the design and can be imported to other designs with a single command.

See Chapter V - Advanced Schematic Editing for more information.

Packaging

DesignWorks has the ability to automatically assign gates to packages, either interactively or on demand. A package table associated with the design keeps track of all package names that have been used and what units are available in each.

This feature also has the effect of automatically naming all devices, even those that do not have multiple gates per package. Names are generated using a prefix field which can be stored in the library or can refer to a default prefix for the design.

See Chapter V - Advanced Schematic Editing for more information.

Browser Tool

The new Browser tool allows you to view and edit attribute data for an entire design in a spreadsheet-style format. This saves time locating and selecting devices or signals across multiple pages and hierarchy levels.

See Chapter XIII - Browser Tool for more information.

Other Features

- The DevEditor tool allows any circuit to be assigned to a symbol. Symbols can be created using the built-in drawing tools.
- Thousands of standard device symbols are provided, including the 74XX series, 4000 series, ECL, microprocessors, memories, discretes, etc.

• A circuit schematic can consist of up to 1000 pages, each one being a single or multiple standard sheets, up to a total of 5 feet by 5 feet. Each circuit page is displayed in a separate window each with independent control of scroll and zoom. Logical connections are allowed between pages using a page connector symbol. Hierarchy levels can be nested to any depth.

• Any number of circuit windows can be open simultaneously, allowing easy copying of partial or complete diagrams from one file to another.

• Commands and drawing modes can be selected using menu items, keyboard accelerators or a floating tool palette.

• Any group of objects on the drawing can be repositioned with a simple click-drag mouse action. Signal lines are rerouted interactively to maintain right angle connections.

• Multiple signal line routing methods allow most pin-to-pin connections to be made with only two mouse clicks.

• Any combination of named signals can be represented by a single bus line and any subset of these can be brought out through a breakout at any point.

• Signal names are global across a single page and across multiple pages when a page connector symbol is added to the signal line. Like-named signal traces on a single page are thus logically connected for simulation and netlisting purposes.

• Drawing borders and reference grids can be drawn automatically. Arbitrary pictures and text created can be pasted onto a circuit schematic. Similarly, complete or partial circuit diagrams can be pasted into word processing or drafting documents.

• Objects can be drawn in user-selectable colors on machines equipped with a color display.

Interface Features

• The report generator tool produces custom-formatted net and component lists from DesignWorks circuit files. These output files can incorporate user-defined device and signal attributes such as special part numbers, manufacturer or supplier names, power consumption, package type, cost, or any other information. This allows complete production circuit data to be passed to other equipment or software, or included in documentation. Standard form files produce output for numerous industry netlist formats.

• Back-annotation to DesignWorks schematics is available for many popular PCB layout and simulation systems. This is done using add-on tools specific to the target system and these are sold separately.

Limitations In This Version

- Maximum number of pages in a circuit is 1000.
- Absolute maximum number of devices in each circuit level is 32767. There is no limitation on circuit nesting depth.
- Maximum length of a device, signal or attribute field name is 16 characters.
- Maximum length of a library part name is 32 characters.
- Maximum length of a pin number is 4 characters.
- Maximum number of characters in an attribute block is 32000.
- Maximum number of pins on a device is 800.
- The entire design must fit into available memory.
- Only TrueType fonts are supported.

Compatibility

- Requires 16MB of memory.
- Requires Windows 95 or later or Windows NT 4.0 or later.

Notes Regarding Copyright

The DesignWorks software and manual are copyrighted products. The software license you have purchased entitles you to use the software on a single machine, with copies being made only for backup purposes. Any unauthorized copying of the program or documentation is subject to prosecution.

Note regarding trademarks

A number of product trademarks are referred to in this manual. Full credit for these is given in the Appendix.

Chapter II - Getting Started

This section gives you information on installing and starting up DesignWorks.

NOTE: Since the DesignWorks package and Windows are constantly being upgraded, there may be recent changes and additional information supplied in Release Notes with the package. It is important that you review these prior to installation as they may contain information that supersedes that given here.

We will be assuming that you are already familiar with general Windows operation, such as copying files, creating folders, etc. If you are not, then work through the introductory sections of the Windows user's guide.

IMPORTANT: Make a backup copy

Before proceeding, make a backup copy and a working copy of your DesignWorks disks, then put the originals away for safekeeping.

Installation

For more detailed information on the files included with the system or directions for installing the current version, see the release notes supplied with the current software version.

General Rules

In general you can place any of the files supplied in any directory or on any disk as suits your preference. The only absolute rule is that the "DW.ini" file must not be renamed and must be in the same folder as DesignWorks itself, or it will not be found at startup and no initial setup will be done. The other general warning is that if you change the name or location of the BasicLib or Tools directory, it will be necessary to make corresponding changes to the "DW.ini" file. See Appendix for more information on how to do this.

The device libraries take up quite a bit of disk storage (around 30 megabytes for a complete installation), so if you're short of space you might want to peruse the Device Library Index (provided in paper and file form) to determine which ones you will be actively using. To really optimize space utilization, you can use the Lib Maint command functions to select only the individual devices you need from the libraries provided and make your own set.

See Chapter IX - Schematic Menu Reference in this manual for information on how to do this.

External Tools

DesignWorks 3.0 consists of a small central application (internally called System Services) and a collection of external code modules which implement most of the program functions. In order for DesignWorks to recognize and load these modules, they must be in the folder specified in the TOOLFOLDER statement in the "DW.ini" file. If there is no TOOLFOLDER statement then by default external modules will be loaded from the directory "Tools" in the DesignWorks directory. All tools located will be loaded when DesignWorks is first started and will appear in the Tools menu. Normally, the Schematic tool will be activated at startup and will display a dialog requesting the type of initial schematic to create. Other tools with more specific functions (such as Report or DevEditor) are invoked by selecting their name in the Tools menu.

In this version, the only tools that are essential to normal operation of the program are Schematic and LibIO. In order to conserve memory, other tools can be kept aside in a storage directory except when actually needed.

See the installation notes supplied with this package for up to date information on tool installation.

Default Library Directory

The DW.ini file allows you to name one or more directories as default library directories. All libraries in these directories will be opened automatically when the program is started. In the standard Ini file provided with DesignWorks, the default folder is called "BasicLib".

> *See the description of the LIBRARYFOLDER keyword in Appendix D -Ini File Format.*

The Setup File

The setup file, "DW.ini", is a text file which specifies some initial actions DesignWorks will take each time it is started up. The setup file specifies the following information:

- where to locate external code modules.
- which libraries to open when the program is started.
- which colors to use in displaying objects.
- other options.

The setup file can be created or edited using your favorite programming editor or word processor. If you use a word processor, be sure to save the file using a "text only" option. The setup file must be called "DW.ini" and must be in the same folder as the DesignWorks program itself.

A typical DesignWorks ini setup file looks like the following:

```
[System]
TOOLFOLDER=Tools

[Libraries]
LIBRARYFOLDER=BasicLib
```

The setup file is divided into sections. Each section starts with a name enclosed within square brackets. Each lines in a section has a keyword followed by its value.

- The section "System" list defaults for system services. The "TOOLFOLDER" keyword specifies the directory to look in for external code module files.
- The section "Libraries" lists directories and files to be opened as libraries. The "LIBRARYFOLDER" keyword specifies a directory to examine for device library files. All found libraries will be opened.

For more information on the meaning and format of these statements, see Setup File Format in the Appendix.

Chapter III - Schematic User Interface

This chapter provides general information on the use of windows, drawing tools and other user-interface features of DesignWorks.

Mouse Button Usage

Three different mouse button actions are used for various functions in DesignWorks. For clarity we will use the following terminology when referring to these actions in the remainder of the manual:

Click - means press and release the left button without moving the mouse. Example: To select a device, click on it.

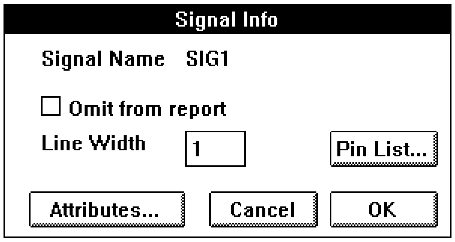
Click and drag - means press the left mouse button and hold it pressed while moving the mouse to the appropriate position for the next action. Example: To move a device click and drag it to the desired new position.

Double-click - press and release the left button twice in quick succession without moving the mouse. Example: To open a device's internal circuit, double-click on the device.

Dialog Boxes

Many DesignWorks functions require information to be displayed or prompted from the user. To do this a special window called a "Dialog Box" is displayed, such as the following one which is used when a Get Info command is executed for a signal.

In general, the controls in dialog boxes will behave according to Windows standards.


Signal Info

Signal Name SIG1

□ Omit from report

Line Width 1

In most dialog boxes requiring text entry, the keyboard equivalents for the clipboard command Cut (Ctrl-X), Copy (Ctrl-C) and Paste (Ctrl-V) are active and can be used to transfer text to or from a text box.

Window Usage

Circuit Windows

Each circuit window displays one page of a circuit schematic. The title on a circuit window will be the name of the circuit file with the number of the page displayed in that window, as in “Interface Board:3”. Any number of pages in a given circuit and any number of circuits can be displayed simultaneously. At any given time only one page of one circuit is “current”, that being the one in the topmost window. Any other window can be made current simply by clicking the mouse anywhere in the window.

Tool Windows

External modules, or Tools, can create their own windows which may be displayed concurrently with other DesignWorks windows. In general these obey the normal rules for resizing, etc., but their function is totally defined by the module.

Closing a window

Selecting Close from the system menu on a Circuit window has the effect of closing the circuit page. If the page being closed was the only one open on the circuit then the circuit is closed.

Redisplaying a window

When a circuit page window is closed it is removed from the Window menu to avoid clutter. To reopen a page in the current circuit level, use the Page command in the Drawing menu. To reopen an internal-circuit window, double-click on the device in question.

Resizing a window

To enlarge or reduce a window, position the pointer in the border of the window and press and hold the mouse button. As long as the button is held down, an outline of the window will track the mouse movements. When the button is released the window will be redrawn with the new size and shape.

To expand a window to fill the whole screen, click the "Maximize" icon in the upper right corner of the window.

Moving a window

To move a window, position the pointer in the title bar anywhere along the top edge of the window and press and hold the mouse button. As long as the button is held down, an outline of the window will track the mouse movements. When the button is released the window will be redrawn at its new position.

The Window Menu

The Window menu provides a means of bringing to the front any window currently open.

Keyboard Usage

The keyboard is only absolutely required when entering names for devices or signals, or for placing random text notations on the drawing. However, the control and shift keys on the keyboard can be used with many editing operations to invoke special features such as auto-alignment, auto-naming and different signal line drawing methods. In addition the arrow "cursor" keys (if available) can be used as a convenient way of setting symbol rotation while placing devices or pasting circuit groups. These options are described in detail in the relevant chapters.

Pop-up Menus

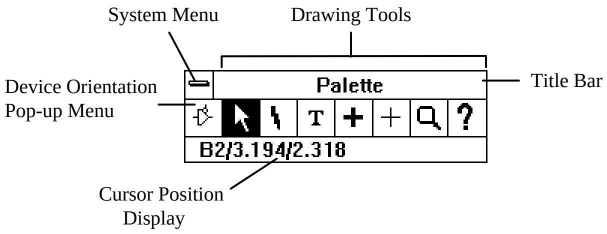
At any time while editing a schematic diagram you can click the right mouse button on a schematic object. A pop-up menu will appear under the cursor allowing you to select from commands appropriate to that object. E.g. the menu for a device contains commands to get device information, edit attributes, open the internal circuit, flip or rotate the symbol, Cut and Copy operations, change gate package assignment, etc.

Separate pop-up menus are available for devices, signals, pins, attributes, and (if clicking in open space on the drawing) the circuit itself.

For details on commands in pop-up menus see Chapter IX - Schematic Menu Reference.

Tool Palette and Status Display

The tool palette is a small window which always remains in front of any circuit windows currently open. The individual items in this display are described in their own sections below.



Hiding the Tool Palette

The tool palette can be removed from the screen by selecting "Close" from its system menu. To re-display the palette, select the Show Tool Palette menu item in the Edit menu.

Status Display Line

The lower area of the tool palette window is used to optionally display several items of status information.

Cursor Position

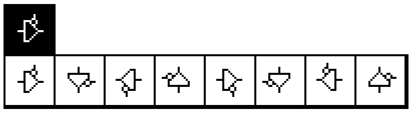
The cursor position is shown as a grid reference and an X and Y distance in inches from the sheet origin (which may be any corner, depending on sheet setup). The cursor position display can be enabled or disabled in the Design Preferences command in the Drawing menu.

Orientation Control

The orientation icon shows the orientation that will be used when the next device or circuit scrap is pasted into the schematic. The orientation can be changed by clicking on the icon, using the arrow keys on the keyboard, or by using the Orientation menu command.

Orientation Pop-up Menu

Clicking and holding the orientation icon in the tool palette will cause the following pop-up menu to appear:



Choose any one of the 8 possible orientations by sliding across this menu. The selected orientation will be displayed in the tool palette icon.

This selection has no effect on items already placed in the schematic. It affects only the flickering image of items currently being placed and future devices selected from the parts palette.

More information on symbol rotation is given in Chapter V - Advanced Schematic Editing.

Schematic Drawing Tools

In DesignWorks there are seven drawing tools. When selected they set a number of different cursor modes which determine what action will be performed when the mouse button is clicked. Following is a summary of the cursor modes. More detailed descriptions of operations performed in each mode are provided in later reference chapters.

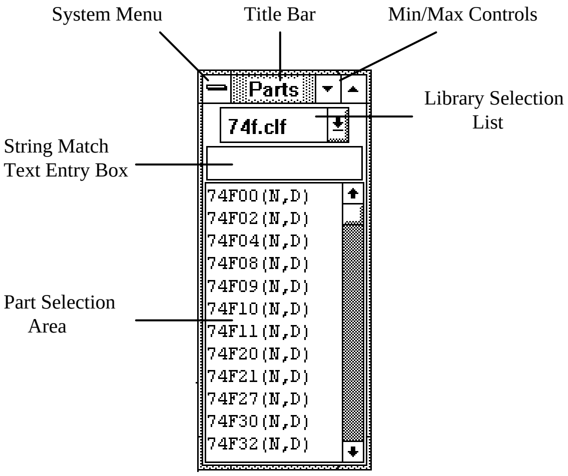
Note that the cursor shape sometimes differs from the tool palette icon for ease of pointing.

Tool Palette Icon	**Initial Cursor Shape**	**Equivalent Menu Command**	**Description**
↖	⇖	Point	Used to select or drag objects, extend signals.
✚	✖	Draw Bus	Used to create a new bus line or extend an existing bus. Clicking once fixes a corner, double-clicking terminates the line.

+	×	Draw Sig	Used while creating a new signal line or extending an existing signal. Clicking once fixes a corner, double-clicking terminates the line. Note that most signal drawing operations can also be done in Point mode.
T	✎	Text	Used to select a signal or device to name, or to place random text on the diagram. Point at the item you want to name and press and hold the mouse button. Move to where you want the name to appear, then release the button.
ϟ	ϟ	Zap	Used to remove single objects. Press the button to remove whatever the tip of the cursor is pointing at. Objects can also be removed in groups by selecting them and using the Clear command or delete key.
🔍	🔍	Magnify	Used to zoom in and out. Clicking on a point or dragging down and right zooms in, dragging up and left zooms out.
?	?	Attribute Probe	When the "dot" portion of the cursor is clicked on a device, pin or signal, the contents of the primary attribute fields for that object are displayed.

Parts Palette

Parts library contents are displayed in a floating palette window which looks like the following:



The Parts palette displays the contents of the selected library file and allows any item to be selected for placement in the schematic.

Moving the Palette

The Parts palette can be moved to any desired location on the screen by clicking and dragging in its title bar.

Hiding the Palette

The Parts palette can be removed from the screen by selecting "Close" in its system menu. To re-display the palette, select "LibIO" from the "Tools" menu in the DesignWorks window.

Choosing a Library

The palette displays only the contents of only one library at a time. The library currently being displayed is shown in the library selection list box at the top of the palette. Clicking in this box allows you to select from among any of the open library files. Library files are opened by selecting the Open Lib command in the Parts palette pop-up menu discussed below.

Selecting a Part

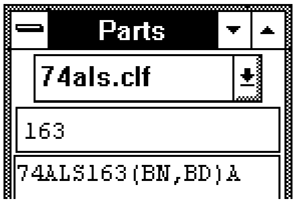
To select a part for placement in the schematic:

1) If necessary, use the scroll bar to scroll the list until the desired part name is in view.

2) Double-click on the part name in the list.
3) Move the cursor to the current schematic window.

Using the String-Matching Box

The string match text box allows you to type characters which will reduce the size of the list and make it easier to locate the desired part. Simply click in the text box and type the desired characters. After a brief pause, the displayed part list will be reduced to only those parts that contain the typed string of characters. For example:



Parts

74als.clf

163

74ALS163(BN,BD)A

To return to the full selection, click at the end of the characters in the box and backspace over them until the box is cleared.

Parts Palette Pop-up Menu

Pressing the right mouse button anywhere in the Parts palette window will display the Parts Palette’s pop-up menu.

> *See Chapter IX - Schematic Menu Reference for specific information on these items.*

Chapter IV - Schematic Editing

This chapter provides information on the elements of a DesignWorks design and how one is created.

Design Structure

What is a Design?

In DesignWorks the term “design” refers to a complete, independent logical entity. The following rules outline how a design is stored:

- A single design is stored in a single file and no logical connections are made between designs. All information required to display and edit a design is stored in the design file.
- A design never makes reference to external library files. When a library device is used, all information needed is read from the library and stored with the design. Changing the original library definition will not automatically update the design. Library devices can be updated manually using the Update from Lib command.
- A design has a top-level circuit, referred to as the *master* circuit. This circuit may contain any number of devices which themselves can contain circuits, called *sub-circuits.* Sub-circuits can be nested to any desired depth, limited only by available memory.
- When a design file is opened, the entire contents of the design are read into memory. This means that design sizes are limited by the available memory in your computer.
- A number of user-settable parameters are stored with the design and affect the entire design when changed. These include:
    - attribute and pin number text style settings
    - the attribute definition table
    - signal and device auto-naming settings
    - display options, such as crosshairs and printed page breaks
    - page reference format
    - hierarchy mode

What is a Circuit?

A design consists of a single master circuit with any number of nested sub-circuits.

- Each master circuit or sub-circuit can consist of one or more pages (up to 1000 in this version). The user can elect to draw the entire circuit on a single page or divide it up functionally onto a number of pages.
- Each page is viewed in a separate circuit window and any or all pages in a circuit can be displayed on the screen simultaneously.
- A page is drawn on the screen as if it was a single piece of paper, although it may have to be broken up into a number of individual sheets of paper for printing or plotting.
- Logical connections can be made between pages using the Page Connector device allowing for full simulation and netlisting across pages.
- If the circuit is a sub-circuit, then logical connections to the parent device symbol are made using the Port Connector device. Port connectors in the sub-circuit are matched by name with pins on the parent device.

Types of Objects in a Circuit

A DesignWorks circuit is made up of four types of entities: devices, signals, text objects and picture objects.

- A device is an object having a symbol, signal connection points called "pins", and optional attributes, internal circuit and simulation information. A device in DesignWorks can correspond to a physical device in a circuit, or it can represent a sub-circuit block or a *pseudo-device*, such as a page connector or bus breakout.
- A pin is a connection point on a device. A pin is not an independent entity since it only exists as part of a device and cannot be created or deleted separately. However, pins can have attributes, pin numbers and other parameters that may be different from pin to pin on the same device. The Get Info command can be used on a selected pin to view and set pin parameters. A *bus pin* is a special type of pin that represents an arbitrary number of internal pins. The internal pins are not visible on the schematic but can still have the same logical properties as other pins.
- A signal is a conductive path between devices. Signal connections can be made visually by drawing lines between device pins, or logically by name or bus connection.
- A text object is used to place a title block or other notation on the diagram. Text can be typed and edited directly within DesignWorks, or can be created externally and pasted onto the diagram from the clipboard. Text objects are

not associated with any other object and are not accessible through net or component lists. The attribute facilities should be used to associate text with specific devices or signals. Text operations are described later in this chapter.

- A picture object is used to place any graphics item imported from the DevEditor tool via the clipboard. The Schematic tool cannot edit picture objects directly, other than moving, deleting and copying them whole. They can, however, be pasted into a DevEditor window, edited using the device symbol tools, then copied and pasted back into the schematic. Picture operations are described later in this chapter.

Classes of Devices

For the purposes of this section, device types in DesignWorks can be divided into four groups:

- Schematic symbols: These are the normal symbols used to represent physical devices on a schematic. Their simulation function, if any, is determined solely by defining an internal circuit.
- Pseudo-device types: These are the symbols used for page connectors, power and ground symbols, etc.
- Simulation types: These are special device symbols intended for use with the DesignWorks Digital Simulator option.
- "Behavioral model" devices have a simulation function defined by program code using the MEDA interface.

See a description of primitive types in DesignWorks in Appendix B - Primitive Device Types.

Device Libraries

The symbols and related parameters for DesignWorks devices are stored in data files called device libraries. Libraries can be opened and closed using the Open Lib and Close Lib commands, or using entries in the setup file.

For each device type in a library the following data is stored:

- general information on the type, such as number of pins, number inputs, number of outputs, type name, default delay, default attributes, position, orientation and type of each pin, etc.
- a picture representing the symbol for this type.

- a rectangle outlining the symbol, used for highlighting and erasing the symbol.
- an optional reference to an external circuit definition.

How Devices are Created and Stored

Libraries are created and modified using the DevEditor tool, which is described elsewhere in this manual.

See Chapter X - Device Symbol Editing for more information.

Device Names

Note re terminology: In this manual, we use the term "device name" to refer to the character string that identifies a unique device in the circuit. Typical device names might be "U23", "C4", "IC12A", etc. This is distinct from the "type name" or "part name" that is used to distinguish the type definition that is read from a device library. Typical part names are "74LS138", "MC68000L8", "SPDT Switch", etc.

Device names may contain any letters, numbers or special characters that you can type on the keyboard, but are restricted in length to at most 15 characters. The name associated with an object can be placed anywhere on the diagram and will be removed if the object is removed.

NOTE: Although DesignWorks does not enforce restrictions, we recommend not using blanks or special characters in device or signal names. This can result in problems exporting data to external systems such as PCB or simulation tools.

Drawing Signals

Signal lines are drawn in either Point mode or Signal Drawing mode.

Interconnecting Signals

If you draw a signal line such that the end of the line contacts a second signal line, then those two signals will be interconnected. Also, if you place a new device such that one of its pins contacts an existing signal line, that pin will be connected to the signal. If both of the two signals being connected were named, then you will be prompted to choose the name of the resulting signal. Whenever three or more line segments belonging to the same signal meet at a given point, an intersection dot will be placed at that point automatically.

NOTE: For efficiency, signals are only checked for connections at their endpoints and only signals actively being edited are checked. It is possible to create overlapping lines that do not connect by unusual combinations of editing operations. This situation is usually visually apparent at the time the editing is done since the intersection dot will be missing and the entire signal will not highlight when clicked on. You may wish to also use the ErrorFind tool to locate these situations.

See more information on connection checking under the Paste command in Chapter IX - Menu Reference.

Naming Signals and Busses

Names may contain any letters, numbers or special characters that you can type on the keyboard, but are restricted in length to at most 15 characters. The name associated with an object can be placed anywhere on the diagram and will be removed if the object is removed.

Connecting Signals by Name

Signal names can be used to make logical connections between wires that are not visually connected on the schematic. The following rules apply:

- Signal names must be visible to be checked for connections, unless a Signal Connector device (e.g. Ground) is attached. More information on invisible names is given in the following section.
- Signal names are known throughout a single page. Like-named signal traces on a single page are thus logically connected for simulation and netlisting purposes. Whenever a signal name is added or changed, the circuit is checked for a change in connectivity. If the name is now the same as another signal on this page, the two signals are merged into one. If this signal segment was previously connected by name to others and the name is changed, then the logical connection is broken. Whenever a name change causes two signals to be connected, both parts will flash on the screen to confirm the connection.
- Signal names are global across all pages of the circuit when a page connector symbol is added to the signal line. Thus if the name is changed on a signal line having a page connector then all circuit pages are checked for like-named signals having page connectors. If any such signals are found then a logical connection is made between them.
- Signals which are contained in busses are a special case. All signals contained in busses have a name, even if this is not displayed on the

diagram. However, the names of bussed signals will not be used to make logical connections unless an explicit name label has been added to the signal line.

For example, if you have a bus containing a signal named CLK and a separate signal line also named CLK, there will be no logical connection between these two signals. The name appearing on the bus breakout is part of the breakout symbol and is not considered to be a name label. If an explicit label is added to the bussed CLK signal (using the Name cursor) then the two CLKs will be logically connected.

- The same rules discussed above for signals also apply to busses. Whenever two busses are logically connected, all like-named internal signals also become logically connected. Note that connecting busses across pages requires a "Bus Page Connector" device rather than a "Page Connector" to avoid compatibility problems between bus and signal connections.

What Signal Names are Used For

The signal name is referenced by the following DesignWorks functions:

- The signal name is used in Report Generator output, such as netlists and simulation data reports.
- Signals can be logically interconnected by name.
- The signal can be located by name using the Find tool.

Invisible Signal Names

DesignWorks allows you to assign and edit signal names in a circuit without making them visible on the diagram. This can be used to assign names that cannot be conveniently placed on the diagram, or that are needed for report generation purposes only. Signals with invisible names are not connected by name, except for invisible names created by a Signal Connector device, as described above.

Auto-Naming Editing Features

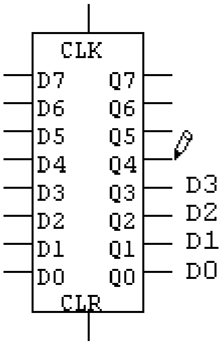
Two features are available to simplify the naming of groups of related signals, devices and pins. These features are activated by holding down the Control, Shift, and/or Tab keys while selecting the signal to be named with the Text cursor.

Auto name generation - If the Control key is held down while a signal, device or pin is selected, a new name is generated automatically for this item. The new name will be the same as the last one entered, except that the numeric part of the name will have been incremented. If the previously-entered name did not have a numeric part, then a "1" digit will be appended to it. The numeric part of the name may be decremented by additionally holding the Shift key.

Auto-alignment - If the Tab key is held down while the signal is selected, the text insertion point will be positioned horizontally aligned with the last signal name that was entered. The vertical position is determined by the vertical position of the line that was clicked on. This feature works only with signal names, not with devices or pin numbers.

Sequential Naming - The above two features can be used in combination to perform easy naming of sequential signals. The normal symbol standard in DesignWorks is to position the highest numbers at the top:

- number the bottom-most line in the group (e.g. D0) using the normal naming technique, described above, then hold Tab and Control while clicking on successive higher-numbered lines.



Note that when you select each successive line, the new name appears, then it is necessary to click again or press Enter to make the name permanent.

Pin Numbering

Pin numbers may contain 1 to 4 characters and are always positioned adjacent to the associated pin.

Uses of Pin Numbers

Pin numbers are used only for labeling purposes and have no particular connectivity significance to DesignWorks. Pin numbers are not checked for duplicates or other invalid usage. Pin numbers placed on a diagram will be used in creating a netlist (see the Report Generator manual), and will appear when the circuit is printed. If a pin is unnumbered, it will appear in a netlist with a "?" unless the AUTONUMBER option is used in the report form file.

Default Pin Numbers

A device type may have default pin numbers associated with it which will appear when the device is first created. These pin numbers are not permanent and can be edited or removed by the techniques discussed in this section. Devices in a library can have default pin numbers assigned to them using the DevEditor tool.

See Chapter X - Device Symbol Editing.

Text and Picture Objects

Text and picture objects are used only to enhance the graphical appearance of a schematic diagram. They have no logical significance in the design.

Random text items are not associated with any particular device or signal on the screen and should not be used to set a name or attributes for devices or signals. The text in these boxes is not accessible in net or component lists. Use the naming and attribute features to attach text to devices and signals.

Creating a Text Block

If you click the pencil cursor on the diagram not near a device or signal line a blinking cursor will appear at that point and you will be able to type any desired text on the diagram. Text entry is terminated by clicking the mouse outside of the text entry rect.

Editing Random Text

If you click the pencil cursor inside an existing text item the insertion point will be positioned at the click point. You can then use normal text editing techniques to modify the text. Note that text on the clipboard can be pasted into an existing text box using the Ctrl-V key equivalent for the Paste function. Similarly, the key equivalents for Cut (Ctrl-X) and Copy (Ctrl-C) can also be used while editing a text box.

Text boxes can be Zapped, Duplicated, Cut, Copied, Pasted and dragged just like any other item on the screen. See the descriptions of these commands for more information.

See Chapter V - Advanced Schematic Editing for more information on creating title blocks and sheet borders.

Sheet Sizes

DesignWorks 3.0 allows sheet size and border layouts to be set by one of two methods:

1) Importing an existing layout using the Import Sheet Info command. This is the normal method of setting the sheet size for a new design. DesignWorks is shipped with a number of templates for standard sheet sizes. These are actually just empty designs in which the sheet info has been preset. You can create your own custom sheet layouts which can then also be imported by this method.

2) Creating a custom layout using the Custom Sheet Info command. This command, and the associated text and picture capabilities, allows all aspects of the sheet size, drawing grid, border text style, title blocks and graphics to be set up to meet your drawing standards.

Custom Sheet Sizes

DesignWorks supports very flexible use of plotters and printers for output on a variety of media. For this reason, there can be complex interactions between the drawing area allocated for each page and the way the schematic is presented when it is output on various kinds of output devices.

The following information is considered to be part of the complete sheet definition:

- the overall height and width of the drawing area.
- the origin and spacing of the location grid.
- the format of text used in the border.
- whether or not the border size should be linked to the current printer page setup.
- whether or not the page should auto-expand to multiple sheets.
- any text and picture objects marked as "border" items.

When the Import Sheet Info command is used, all the above information is imported from page 1 of the selected design.

Default Sheet Layout

In this manual the term "page" is used to refer to a circuit page, which is the single, contiguous area that is viewed in one circuit window. The term "sheet" is used to refer to the physical sheet of paper that the diagram is printed or plotted on. Depending on the sheet size settings, a page may correspond to one or more sheets.

The default sheet size options "Use Page Setup" and "Single Sheet" give the simplest situation. With these settings, each circuit page will be printed on a single sheet of paper and any change in the printer Page Setup will be immediately reflected by a change in the drawing border.

When you first create an empty circuit using the New command, the drawing area defaults to a single sheet using the current printer Page Setup. Any area outside this sheet will appear in gray on the screen.

General Editing Concepts

The Clipboard

The standard clipboard commands Cut, Copy and Paste can be used to move or copy circuit fragments, graphical and text information within a single circuit window and between windows.

Cut and Copy work on the currently selected group of objects and will be disabled if nothing is selected. See the earlier section in this chapter on "Selecting Circuit Objects". When items are copied onto the clipboard, their names are copied with them, which may result in duplicate names. If duplicate signal names are pasted back into the circuit page they were copied from, then logical connections will be made between the like-named segments.

See Chapter IX - Schematic Menu Reference for information on the specific clipboard commands.

Selection

Many DesignWorks commands, such as Get Info, Cut, Copy, etc., operate on the currently selected objects. For a detailed list of the techniques available for selecting objects, see the Editing Procedures section at the end of this chapter.

Multi-page Selection

The DesignWorks commands that operate on selected objects normally only affect objects on the current page. However, some external tools, such as the Browser, Find and ErrorFind, operate across pages and can select objects on non-current pages. This is done to allow global operations to be done without having to go to each page and perform separate editing operations. For example, Find can be used to locate all devices of a particular type and select them, then Browser can be used to display and edit their attributes.

Note the following rules for multi-page selection:

- Clicking in an empty part of a schematic window deselects all objects everywhere in the design.
- Switching from one schematic window to another does not change the status of any objects that were selected on either page.
- You can select objects on one page, switch to another page, and (using the normal SHIFT key technique) add items on the new page to the selected group.
- Editing commands like Cut, Copy and Clear will only affect selected items on the current page, even if objects are selected on other pages.

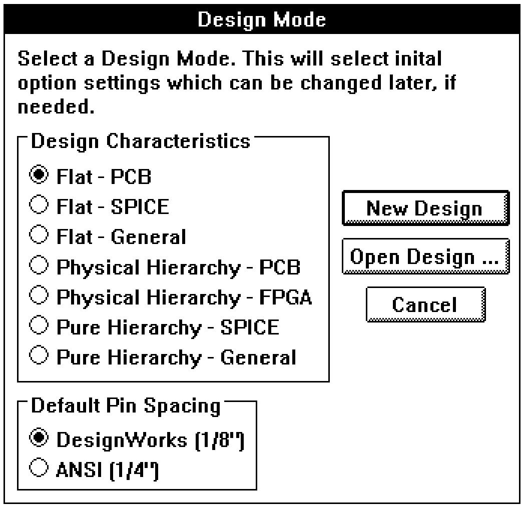
Schematic Editing Procedures

The purpose of this section is to give you a quick outline of commonly-used schematic editing techniques in DesignWorks. Background on each of these techniques is provided earlier in this chapter and in other reference sections of the manual.

Creating a New Design

A new design is created by selecting the New Design... command in the File menu. The new design will consist of an empty master circuit with a single page. This page will appear in a circuit window entitled “Design1:1” (the ":1" portion indicates that this is page 1). The New Design... command does not create a disk file. The design exists only in memory until you do a Save Design As command.

Each time a new design is created, when the New Design... command is selected and when the program starts up, a design mode dialog box is displayed. This dialog does not actually implement any new options, it just groups them by design type and encourages you to select the appropriate mode in advance. All options can be set individually later in the design if desired, although this dialog can only be displayed when creating a new design.



Select a Design Mode. This will select inital option settings which can be changed later, if needed.

Design Characteristics

● Flat - PCB
○ Flat - SPICE
○ Flat - General
○ Physical Hierarchy - PCB
○ Physical Hierarchy - FPGA
○ Pure Hierarchy - SPICE
○ Pure Hierarchy - General

Default Pin Spacing

● DesignWorks (1/8")
○ ANSI (1/4")

Users are encouraged to use one of the Flat modes until they are familiar with the ramifications of the other design modes.

Note: If you always use the same options and want to override this dialog, the Ini file options NEWDESIGN and NOUNTITLED allow you to control the creation of untitled designs. See more information on these options in the Appendix.

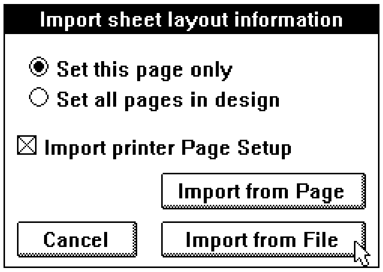
Your circuit diagram is created by first placing one or more devices in the circuit window (as described below), and then interconnecting the device pins with signal connections.

Disposing of a Design

DesignWorks has no built-in command to dispose of a design file. All information about a design is stored in a single file. Window’s "File Manager" may be used to delete a complete design.

Selecting a Standard Sheet Size

Select the Import Sheet Info command in the Drawing menu. The following box will appear:



Import sheet layout information

● Set this page only
○ Set all pages in design

☒ Import printer Page Setup

Click on the Import from File button. Locate the Sheet Templates folder inside the DesignWorks folder. Select the desired sheet template and click the Open button.

Navigating Around a Schematic Page

In addition to the standard scroll bars and Reduce/Enlarge menu items, DesignWorks has a number of convenient features for moving around a diagram.

Auto-Scrolling

Whenever the mouse button is depressed and moves close to the edge of a schematic window, the window automatically scrolls to expose more area on that side.

Zoom (Magnifying Glass) Tool

The 🔍 item in the tool palette is a powerful tool for moving around in a schematic diagram. It can be used to zoom both in and out and can control the exact area displayed on the screen.

1) Clicking and releasing the mouse button on a point on the diagram will zoom in to that point by one step.

2) Clicking and dragging the mouse down and to the right zooms in on the selected area. The point at which you press the mouse button will become the top left corner of the new viewing area. The point at which you release the button will become approximately the lower right corner of the displayed area. The circuit position and scaling will be adjusted to display the indicated area.

3) Clicking and dragging the mouse upward and to the left zooms out to view more of the schematic in the window. The degree of change in the scale factor is determined by how far the mouse is moved. Moving a small distance zooms out by one step (equivalent to using the Reduce command). Moving most of the way across the window is equivalent to doing a Reduce to Fit.

Find Tool

The find tool allows you to locate any device, signal or pin by name or any other attribute value. To use it:

- Select Find in the Tools menu.
- Select the type of object you wish to search for, device, signal or pin.
- Select the scope of the search, i.e. current page, current circuit level, entire design.
- Enter the name of the object to search for, or other search criteria. If working in physical hierarchy mode it may be more appropriate to search by the InstName attribute.
- Click the Find button.

The Find tool automatically displays the first item found. The Next button can be used to cycle through all objects matching the given search criteria.

> *More detailed information on the Find tool is provided in Chapter XII - Find and ErrorFind Tools.*

Placing and Editing Device Instances

Selecting a Device From a Library

To select a device from a library for placement in the schematic:

- Select the desired library using the pop-up menu at the top of the Parts palette.
- If necessary, use the scroll bar to scroll the list until the desired part name is in view.
- Double-click on the part name in the list.
- Move the cursor to the current schematic window.

The cursor will be replaced by a flickering image of the selected device. While moving this flickering image around, you can use the arrow keys on the keyboard or the orientation tool on the tool palette to rotate the symbol.

Clicking anywhere in the circuit diagram will make a permanent copy of the flickering device at that point.

NOTE: Holding the Control key while clicking will inhibit checking for pin connections. This allows you to select the device again and drag it to a new position without affecting any existing connections.

Duplicating an Existing Device

Either:

- select a similar device anywhere on the current circuit page and use the Duplicate command (either in the Edit menu or in the device pop-up menu).
- select a similar device in any other page of any open design and use the Copy command. Return to the destination circuit and select the Paste command.

After either of these operations, the cursor will be replaced by a flickering image of the selected device which can be placed by clicking in the schematic, as discussed above.

Deleting a Device

Devices can be removed by either of two methods:

- select the device by clicking on it (holding the SHIFT key if it is a switch or other input device) and then hit the "delete" or "backspace" key on the keyboard, or select the Clear command from the Edit menu.
- enter Zap mode by selecting the Zap command or clicking on the Zap icon in the tool palette, then click on the device in question.

Moving a Device

Devices can be moved by clicking and dragging to the desired new position. If more than one device is selected, all the devices, and all signals connecting between them (whether or not selected) will be moved. Signal lines will be adjusted to maintain right angles at points where moving signal lines intersect with non-moving ones.

Entering Device Attributes

To enter device attribute, either:

• Clicking the right mouse button on the device, then select the Attributes command in the pop-up menu, OR,

• Select the device by clicking on it, then choose the Get Info command in the Options menu, then click the Attributes button.

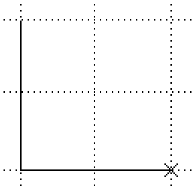
> *See more information on attributes in Chapter VI - Attributes.*

Signal Line Editing Procedures

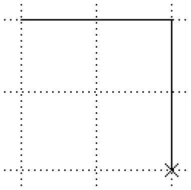
Drawing From an Existing Line or a Device Pin

A line can be extended from the end of an existing line or device pin using the normal (↖) cursor. Click and hold on the end of the pin and drag away from the pin. A pair of right-angle lines will follow the cursor away from the pin and as long as the mouse button is pressed. Releasing the mouse button makes these lines permanent. If the end of the line (i.e. the point where the mouse button was released) touches another signal line, a connection will be made at that point.

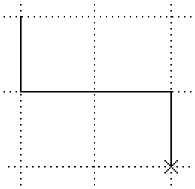
Alternate line routing methods can be activated by pressing the Tab and Control keys, as follows:



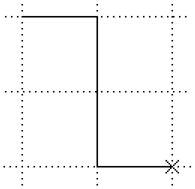
No keys pressed



Control key pressed



Tab key pressed



Tab and Control keys pressed

The Control key inverts the order of line drawing, and the Tab key switches to three line segments with a center break.

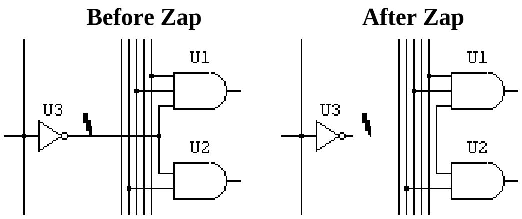
Creating an Unconnected Signal Line

The Draw Sig (+) cursor can be used to create an unattached signal line, or can be used to extend an existing signal. Simply click anywhere in the schematic and drag away in the desired direction. Unlike the Point mode drawing method, above, the mouse button does not have to be held while creating signals in this mode. Double-clicking terminates the signal line.

Editing a Signal Line

The following features are available to edit signal lines:

- Zap mode (entered by selecting the Zap command in the Edit menu or the Zap item in the tool palette) allows you to remove any single line segment from a signal connection. Zapping on a signal line removes only the line segment being pointed at, up to the nearest intersection, device pin or segment join point.



Before Zap **After Zap**

- selecting a signal line (by clicking anywhere along its length) and hitting delete or selecting the Clear command removes an entire signal trace.
- drawing backwards along the length of an existing line causes the line to be shortened to end at the point where you let the button go.
- clicking and dragging the middle of a signal line segment allows you to reposition the line. Vertical lines can be moved horizontally and vice versa.

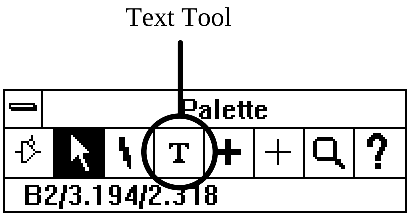
Checking Signal Interconnection

Double-clicking anywhere along a signal line will cause that signal segment and all logically connected segments on the current page to be selected.

Name Operations

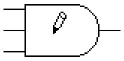
Adding a Device Name

Enter "Text" mode either by selecting the Text menu item in the Edit menu, or by clicking on the text icon in the tool palette:



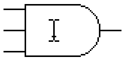
Note that once "Text" is selected, the cursor changes to a pencil icon. Press and hold the mouse button with the tip of the pencil positioned inside a device symbol. As long as you hold the mouse button down an I-beam cursor will track the mouse movements. The device name text will start at the position where you release the button. Type the desired name and press Enter, or click the mouse button outside of the text entry rectangle. This sequence of steps is illustrated below.

1



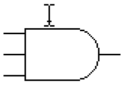
Position the pencil cursor anywhere inside the device symbol.

2



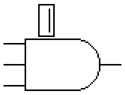
Click and hold the mouse button. The cursor changes to an "I-beam".

3



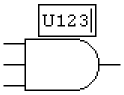
Still holding the mouse button pressed, position the cursor where you want the name to start.

4



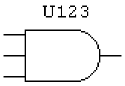
Release the mouse button. A blinking insertion marker and entry rect will appear.

5



Type the desired name, up to 16 characters.

6



Click the mouse button outside the entry rect to make the name permanent.

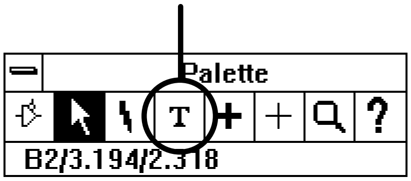
The pencil tool can be requested to automatically locate the device name in a default position by clicking the pencil tool over the device and releasing it without moving the mouse. This will place the name position in a default location.

Once a name is placed, it can be repositioned by dragging it using the Point cursor or removed using the Zap cursor. The device name will be removed automatically if the device is removed.

Device names are stored in an attribute associated with the device. Depending on the design mode, Flat, Physical, or Pure the name assigned will be stored in either the Name attribute or the InstName attribute.

Adding a Signal Name

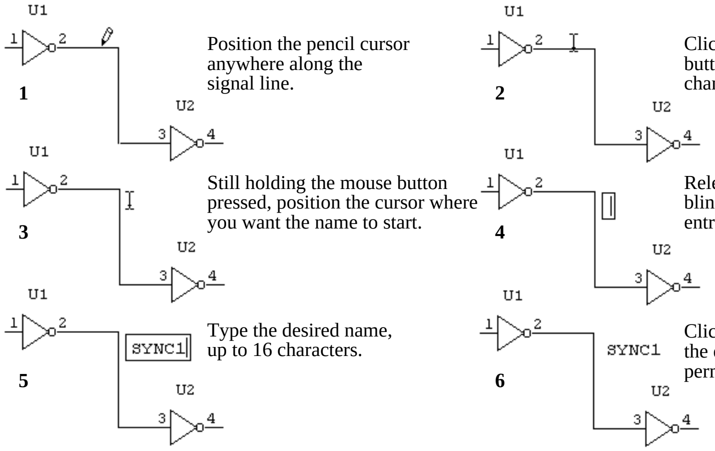
Enter "Text" mode either by selecting the Text command in the Edit menu, or by clicking on the text icon in the tool palette:



Text Tool

Note that once "Text" is selected, the cursor changes to a pencil icon.

Press and hold the mouse button with the tip of the pencil positioned anywhere along a signal line except within 5 screen dots of a device. As long as you hold the mouse button down an I-beam cursor will track the mouse movements. The signal name text will start at the position where you release the button. Type the desired name and press Enter or click the mouse button outside of the text entry rectangle. This is illustrated below.



1 Position the pencil cursor anywhere along the signal line.

2 Click and hold the mouse button. The cursor changes to an "I-beam".

3 Still holding the mouse button pressed, position the cursor where you want the name to start.

4 Release the mouse button. A blinking insertion marker and entry rect will appear.

5 Type the desired name, up to 16 characters.

6 Click the mouse outside of the entry rect to make the name permanent.

Adding an Invisible Name

An invisible name for either a device or signal is created by either:

1) Clicking the right mouse button on the device or signal, then selecting the Name command from the pop-up menu. Or

2) Selecting the desired device or signal, then selecting the Get Info command in the Options menu (Ctrl-I), then clicking on the Attributes button in this box, then selecting the Name field in the attributes box.

In either case, if the name is already visible on the diagram, changing it here will change all displayed occurrences of it.

Making an Invisible Name Visible

An invisible name can be made visible by simply clicking the Text cursor anywhere on the signal or device. When the mouse button is released, the name will be positioned at that point, as shown in the general naming instructions above.

Removing a Name

A device or signal name can be removed by using the Zap cursor, described above. If the signal has been named in multiple locations then Zap removes the name only at the location zapped. Only the name’s visible frame is removed the signal or device still retains its name.

Editing a Name

The name can be changed by simply clicking the Name cursor on the signal name and editing using the keyboard as desired. Alternatively, a name can be edited by doing a Get Info on the signal. Changing the name in the Get Info box or at any one location on the diagram will change all visible occurrences of it.

Moving a name

A device or signal name can be moved by selecting the Point (normal) cursor, clicking and holding the mouse button on the name, and dragging it to the desired new position. Pin numbers cannot be repositioned.

Multiple Naming of Signals

A signal name can appear in up to 100 positions along the length of a signal line. To add a new position, simply use the normal naming procedure given in the section on signal naming i.e.:

- select Name mode
- click and drag anywhere along the signal line
- with the mouse button pressed, move to the desired position for the name
- release the mouse button

A new copy of the signal's name will appear at this point followed by a flashing cursor. To accept the name, simply click the mouse button outside of the text entry rectangle or press the Enter key. If you edit any occurrence of a name along a signal segment, all other occurrences will be updated to reflect the new name.

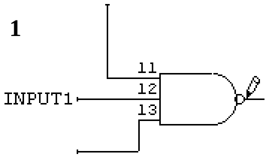
Any occurrence of a signal name can be removed using the Zap cursor. If you remove the last visible name from a signal segment then the logical connectivity to other like-named signals is removed.

Setting and Editing Pin Numbers

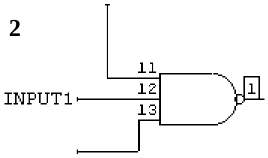
Editing Pin Numbers On the Schematic

If the mouse button is pressed with the tip of the pencil positioned on a signal line within 5 screen dots of a device, a blinking insertion bar will appear immediately where the signal joins the device. You cannot set the text position for pin numbers. Type the desired 1 to 4 character number and press Enter, or click the mouse button anywhere, to make the number permanent. See the sequential numbering option discussed below under "Auto-naming Features".

1



2



Editing Pin Numbers Using Get Info

To edit pin numbers using the Get Info dialog box:

1) Click on the device in question to select it and enter (Ctrl-I).

2) Click on the Pin Info box. This will display the pin information for the first pin.

3) Edit the pin number as desired.

4) Click the Next Pin button to see the next pin in the list.

Selecting Circuit Objects

To select circuit objects, the cursor must be in the normal pointer (↖) mode.

Selecting a Device

A single device is selected by clicking the mouse button with the pointer positioned anywhere inside the device symbol or in any displayed attribute value associated with the symbol.

Simulated input devices such as switches and keyboards can only be selected by holding the SHIFT key while clicking. This is necessary because a normal click is used to change the state of these devices when a simulator tool is active.

Selecting a Signal

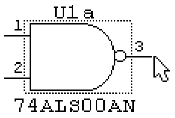
A single signal is selected by clicking anywhere along the signal line. This selects only the part of the signal directly attached to the clicked line. Double-clicking the signal selects all parts of the signal on the given page including logical connections by name or bus.

Selecting a Pin

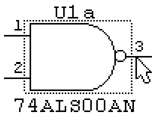
A pin is selected by clicking with the left mouse button on the pin line close to the device. NOTE: Since a device pin is both a pin and a signal, you select the pin or signal as follows:

- • clicking on the pin close to the device symbol, with the left mouse button selects the pin, with the right mouse button displays the pin’s popup menu.
- • click on the last 1/4 of the pin away from the device, with the left mouse button selects the signal, with the right mouse button displays the signal’s popup menu.

Selecting the Signal



Selecting the Pin



Selecting a Picture or Text Object

A single, non-background picture or text item is selected by clicking the mouse button with the pointer positioned anywhere inside the item.

Selecting a Background Object

To select a picture or text item marked as "background" you must hold the Shift and Control keys pressed while clicking on it.

Selecting Groups of Objects

Several methods are available for selecting multiple objects:

1) Any group of adjacent items can be selected by clicking and dragging across the group. A flickering rectangle will follow the mouse movement. Any object that intersects this rectangle when the button is released will be selected (except background objects).

2) A group of interconnected devices and signals is selected by double-clicking on any device in the group while holding the Control key pressed. If a circuit is completely interconnected, this will select the entire circuit.

3) The Select All command in the Edit menu selects all items on the current page except background items.

4) The SHIFT key can be used in combination with any of the above methods to select multiple items. When the SHIFT key is held, the previously selected items are not deselected when a new item is selected. Thus you can add to the selected group until the desired collection of items is selected.

Changing Search Order

Holding the Control key pressed while clicking the pointer causes object types to be searched in the opposite order from normal. This can be used, for example to select a signal name that has accidentally moved under a device, etc.

Deselecting a Selected Object

All currently selected objects are deselected by clicking in an empty area of the schematic window. A single item can be deselected by holding the SHIFT key while clicking on it.

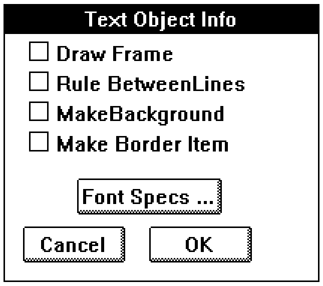
Creating a Text Notation

If you click the text cursor on the diagram **not** near a device or signal line, a blinking cursor will appear at that point and you will be able to type any desired text on the diagram. The Return key can be used to enter multiple lines in a single text block. Text entry is terminated by clicking outside of the text entry rectangle. Text entry may be aborted by pressing the Esc key.

NOTE: Random text items are not associated with any particular device or signal on the screen and should not be used to name or set attributes for devices or signals. The text in these boxes is not accessible in net or component lists. Use the naming and attribute features to attach text to devices and signals.

Text Style and Display Options

To set text display options and text style, select the text block by clicking on it with the Point cursor, then select the Get Info command in the Options menu. This will display the following box:



The following table summarizes the options available in this box.

Draw Frame	Turning this switch on causes a frame to be drawn around the text item on the schematic.
Rule Between Lines	Turning this switch on causes a line to be drawn after each row of characters.
Make Background	Turning this switch on makes the selected text block a background item, i.e. it cannot be affected by normal editing operations. In order to select the text, hold the Shift and Control keys pressed while clicking on it.
Make Border	Turning this switch on marks this item as a border object. This means that it will be treated as part of the sheet border. I.e. it will be updated by the Import Sheet Info command, copied to new pages as they are created, and unaffected by the Center in Page command.
Font Specs	Clicking this button displays the standard text style dialog. Any changes made in text style affect only the selected item, but they also become the default for future text blocks.

See more information on using text notations in title blocks and sheet borders in Chapter V - Advanced Schematic Editing.

Creating a Picture

Pictures can not be created on the schematic diagram. They can be created in the DevEditor tool. To do this:

- enter the DevEditor tool and draw the desired picture.
- select the objects in the DevEditor which form the picture.
- select the Copy command in the Edit menu. This places a copy of the graphic items on the clipboard.
- select your DesignWorks circuit.

- select the Paste command from the Edit menu. A flickering image of the graphics object will follow the mouse movements until you click once in the desired position.

Chapter V - Advanced Schematic Editing

This chapter provides information on the more advanced schematic editing features of DesignWorks.

Device Packaging

DesignWorks 3.0 provides features for automatic or manual assignment of devices to physical packages. This mechanism is intended for use when creating designs that are destined for production in printed circuit board (PCB) or other discrete package format.

The Packager tool has the following features:

- Automatically generates a unique package assignment for each device placed on the schematic.
- Uses a name prefix stored with the device symbol, or a default value stored with the design. This allows appropriate names to be generated for discrete parts, e.g. "D1", "D2", "D3", etc. for diodes, and "R1", "R2", "R3", etc. for resistors.
- Automatically handles device types having multiple units per package, e.g. gates. Special attributes are stored with the symbol to tell the packager how many units are available and the pin assignments for each.

Packaging vs. Auto-Name Assignment

DesignWorks has two related features which can both assign names to devices: Packaging and Auto-Name Assignment.

- Packaging assigns a name, unit and pin numbers to each device based on physical packaging considerations. Packaging is controlled by the commands in the Packaging sub-menu in the Options menu.
- Auto-naming is more restricted and simply ensures that each device symbol in a circuit has a unique name. It is controlled by buttons in the Design Preferences box.

In flat hierarchy mode (the default), packaging takes precedence, i.e. if auto-packaging is enabled, auto-name assignment will have no effect. In physical hierarchy mode, both features can be used together, since the package assignment is not normally used as a device name, but instead is the instance name. Packaging is not available in pure hierarchy mode.

Auto Packaging

When Auto Packaging is enabled (i.e. the Auto Packaging Enabled menu item is checked), the Packager assigns each device to a package as soon as it is placed on the diagram. When a device is deleted, its package assignment becomes available for the next device of the same type that is created. In this mode, package assignments will be in chronological order of placement on the diagram.

It is normally convenient to leave Auto Packaging turned on while initially creating a design. This way, all devices get reasonable initial name assignments. The design can be repackaged later (using the Repackage Design command) once the physical layout of the design becomes more clear.

Re-enabling Auto-Packaging After Manual Edits

If Auto-Packaging has been disabled , by the user or because of a packaging name conflict, then the design’s package table will be out of date. It may be desirable to select the Rescan Design command from time to time to recheck the validity of the current package assignments. This command updates the information in the package table based on name and unit assignments currently in the design. This is done without changing any values in the design. In any case, Auto-Packaging cannot be re-enabled until a successful Rescan is completed.

Auto-Packaging Limitations

The Packager can be confused by some conditions when package assignments are changed manually. Such conditions include:

- two devices given the same packaging assignment.
- a name or unit attribute field set to an invalid value.
- Packager attributes changed.
- a package contains more then 32 units.

In these cases, the Packager will display a warning box and disable auto-packaging. The error must be corrected or the offending device marked with the "Ignore" option (see below) before auto-packaging can be re-enabled.

NOTE: The Packager looks only at the name or instName and unit, unit.all, unit.list attribute fields, not at the pin numbers themselves. If you manually change any pin numbers on devices without making corresponding changes to the Unit field, invalid package assignments could result.

Batch Packaging

The Repackage Design command in the Options menu allows you to request a complete reassignment of device packages. This command performs the following steps:

- the design package table is cleared.
- all devices marked as Locked (see below) are noted in the package table and checked for errors.
- all unlocked devices are sorted and assigned to packages.

For packaging, devices are sorted by page number, then by position on the page. Names are assigned in vertical strips corresponding to the lettered grid references on the page. I.e. all devices in column A are assigned, starting at the top of the page and working down. Next, all devices in column B are assigned, etc.

Manual Packaging

All aspects of device packaging can be controlled manually, if desired. This can be cone on two levels.

Manual/Automatic Packaging

The higher-level approach takes advantage of the packaging information in the device libraries and the error checking capabilities of the Packager. The following packaging features can be used:

- Each library entry contains information about available gate package assignments. The list of available units appears in the Unit sub-menu of the device pop-up menu. Selecting one of these items automatically updates the Unit field and the pin number assignments on the device.
- The Rescan Design command can be used to check for packaging errors without affecting any existing settings. Any duplicate assignments or invalid unit settings will be announced.

The following cautions should also be noted:

- The Packager does not check pin numbers when checking package assignments. It uses only the contents of the Unit attribute field. If you manually change any pin numbers on devices without making corresponding changes to the Unit field, invalid package assignments could result.
- It is best to set the Packaging Level to Locked (using the Get Info command) for devices that you wish to have a fixed package assignment,

even if not using auto-packaging. This will avoid loss of data if auto-packaging is inadvertently enabled or a Repackage Design is done.

Fully Manual Packaging

The lower-level approach completely ignores all packaging features. This can be used if unusual pin assignments are needed, or libraries without packaging information are being used. Note the following points when doing manual packaging:

- Pin numbers can be edited directly on the diagram or assigned using the Pin Info box or the Browser tool.
- If you do not wish to display a unit assignment (e.g. "a", "b", "c", etc.), you can use the Define Attribute Fields command to turn off the Visible by Default option for the Unit field.
- If you wish to manually assign a unit value, it is best to create a user-defined attribute field (e.g. "GateUnit") and make all entries in that field. This is necessary to circumvent the automatic updating of pin numbers when the Unit field is changed.
- When a netlist or Bill of Materials is created using the Report tool, all devices with the same name are normally treated as a single device. This means that they are, in effect, assigned to the same package. The Report tool does not check for duplicate pin numbers or name assignments, although it is possible to customize a report format that will assist in locating these problems.

Package Name Format

Device names are generated by appending a numeric suffix to a prefix provided. If the prefix field (i.e. the field whose name is in the design's PrefixField field) is non-empty in the device, then it is taken as the prefix. This is normally set using the DevEditor tool in the device library entry, although it can be set individually for any device on the schematic. If the prefix field is empty, then the contents of the design's "PkgPrefix" field are used.

NOTES:

1) The default prefix field is Name.Prefix. This can be changed by entering the name of any other field in the design's PrefixField field.

2) Most library parts provided with DesignWorks that represent logic devices or integrated circuits have no Name.Prefix value. They will therefore use the value of the design's PkgPrefix field, "U" by default. Discrete parts or others that normally have specific standard prefixes are set to the common values.

The numeric suffix is assigned sequentially for each different prefix found in the design. E.g. the packager will assign names "U1", "U2", "U3", etc. to devices with the value "U" in their prefix field, and names "R1", "R2", "R3", etc. for devices having prefix "R".

Device Packaging Restrictions

DesignWorks allows you to mark individual devices for special handling during packaging operations. Using the Get Info command, you can select one of the following packaging levels for each device in the design:

Normal (the default value)	This device will get a package name and unit number assigned automatically by the Packager. The assignment can be changed by future repackaging operations if necessary.
Lock and Check	This device’s package assignment will not be changed, but it is recorded in the package table so no other device will be assigned the same name. The device will be checked for packaging errors.
Lock and Don’t Check	The same as Lock and Check except that the package assignment is not checked for errors. This could be used in special cases where the same name is purposely assigned to two different symbols.
Ignore	This device is completely ignored by packaging operations. I.e. its name and unit will not be changed, no error checking is done, and some other device might be given the same name.

Back Annotation and Packaging

Back annotation refers to any process that reads a list of name, unit and pin numbering changes from an external system and automatically updates the design file. Obviously, any such process can potentially bypass the Packager and create invalid package assignments. It is recommended that the Rescan Design command be used after any such process to ensure that the name and unit assignments are still valid.

WARNING: The Rescan Design command does not guarantee the correctness of your design. Pin numbers are not checked. Check your design carefully after any Back Annotation process.

Packaging in Hierarchical Designs

Hierarchy Mode

The usage of attribute fields by the Packager is different between flat and physical hierarchy modes. See the discussion below under "Packaging Attribute Fields". Packaging is not available in pure hierarchy mode.

Name vs. InstName

In hierarchical designs, a device symbol in a sub-circuit can represent multiple physical devices. This happens if the parent device has been instantiated multiple times in the design. For this reason, the device "name" is split into two different functions in a hierarchical design.

The "Name" attribute field is associated with the definition of the sub-circuit and will be the same for all instances of the sub-circuit. For this reason, it can be thought of as the "logical name" or "definition name" of the device and cannot be used to hold the package assignment.

The "InstName" attribute field is associated with each instance of the sub-circuit. It can take on a different value for each physical device it represents and is used to store the package assignment in hierarchical designs. The user can choose to display either or both of these names on the schematic.

See Chapter VII - Hierarchical Design for more information.

Restricting Packaging Depth

In a hierarchical design, a device symbol that represents a physical device may have a sub-circuit for simulation or analysis purposes. Normally, the parent device rather than the contents of this sub-circuit should be packaged. The packaging options in the device Get Info command can be used to control packaging of internal circuits. Standard library devices provided with DesignWorks will default to packaging the parent device even if an internal circuit is added.

Name Assignment Order

In hierarchical designs, all packagable devices in a given circuit level are assigned before the internals of any sub-circuit devices in that circuit. This ensures that a given circuit level contains sequential device names and units.

Using Device Libraries Without Packaging Information

If the Packager encounters a device without any packaging attribute fields define, it takes the following minimum action:

- the device is assumed to have only one unit per package, i.e. each device will be given a unique name.
- the default package name prefix for the design will be used to assign a name.
- no unit assignment will be displayed on the schematic.
- no Unit selection menu will appear.
- default pin numbering defined in the library will be used.

Attribute Fields Used During Packaging

Fields Set By the Packager

The Packager stores the package name and unit assignments in attribute fields associated with each device symbol. The fields that are used depend upon hierarchy mode:

Flat Mode	The package name is stored in the "Name" field and the gate unit (e.g. "a", "b", "c", etc.) is stored in the "Unit" field. This reflects the common usage that the device name used on the schematic is the name of the package on the final PCB.
Physical Mode	The package name is stored in the "InstName" field and the gate unit is stored in the "Unit" field. The "Name" field cannot be used because a single device in a sub-circuit may actually represent multiple devices on the physical PCB. Each one of these devices requires a separate package assignment. The "InstName" field can take on a different value for each instance.
Pure Mode	Packaging is not supported in Pure mode. This is because there is no instance data and therefore no place to store different package assignments for multiple instances of the same device.

Fields Read By the Packager

The following information is used by the Packager in determining what units are available in each package and what pin assignments belong with each unit. Devices with only one unit per package do not need any special attribute fields.

Part	The "Part" field is used to determine which devices can share a package. In some devices, several different types of elements exist in one package. In this case, a different symbol is used for each different logic type, but each must have the same "Part" value.
Name.Prefix	This device attribute field contains the prefix used to generate the device name/instName. If it is empty, then the design’s "PkgPrefix" field is used. This field is normally defined in the device library entry, but can be overridden in each device for special requirements. NOTE: "Name.Prefix" is the default field used for this purpose, but this can be overridden. See the "PrefixField" description below.
PkgLevel	This field contains the package level setting for this device instance. If the field is empty, or contains the character "0", Normal mode is assumed. "1" means Lock and Check, "2" means Lock and Don’t Check, "3" means ignore. Any other value will be interpreted as Lock and Check. This value is not normally defined in the library entry for the device. The package level is usually set using the buttons in the Device Get Info box, but the attribute field can be set directly if desired.
Unit.List	This device attribute field contains a list of the units available for this device symbol. The unit names are separated by commas, e.g. "a,b,c,d". In device types having different types of logical units in one package, this is the list of only those units which can be used with this symbol. This field is normally defined in the device library entry, but can be overridden in each device for special requirements.

Unit.All	This device attribute field contains a list of all units available in this package, whether or not they apply to this symbol. Normally this is the same as "Unit.List", but will be different for devices having more than one type of logical unit per package. Format is the same as for "Unit.List". This field is normally defined in the device library entry, but can be overridden in each device for special requirements.
PkgPrefix	This design attribute field contains the default package name prefix used if no prefix field appears in the device.
VisPin.List	This **pin attribute** field is used to set the pin number corresponding to each gate unit. It consists of a list of pin numbers separated by commas, e.g. "1,4,10,13". The order of the pin numbers in this list must match the order of the units in "Unit.List". This value is normally defined for each pin in the device library entry, but can be overridden on the schematic if desired.
PrefixField	This design attribute field contains the name of the field to use for a device name prefix. The default is "Name.Prefix", but this can be overridden to use specialized prefix values.

PCB Footprint and Package Type Information

If the netlist output from DesignWorks is to be used to generate a printed circuit board, then it will be necessary to associate package type information or PCB footprint information with each device. This is done by specifying a field called "Package" with a keyword for each package or footprint type. Package type fields are included in the standard type libraries, but may be overridden if necessary by specifying a value for the "Package" field in the device attributes.

The keywords associated with the common package types are described in a technical note supplied with the package.

Device and Signal Auto-Naming

When Auto-Naming is enabled, all devices (except pseudo-devices) and signals will be given a default name when they are created. This ensures that every object in a design has a unique name and can be identified in a netlist.

Device Auto-Naming vs. Packaging

If the design is in flat hierarchy mode (the default) and auto-packaging is enabled (the default), then Device Auto-Naming will have no effect. In "flat" mode, both the Packager and the Device Auto-Naming feature set the "Name" attribute field. In this case, the Packager takes precedence.

If the design is in physical hierarchy mode, then the Packager sets the "InstName" field and the Device Auto-Naming sets the "Name" field.

> *See the comments on Name vs. InstName in hierarchical designs earlier in this chapter and in Chapter VI - Attributes.*

Enabling Auto-Naming

Auto-Naming is enabled by selecting the Design Preferences command in the Drawing menu. Checking the corresponding box causes all devices placed in the design subsequently to be given a default name if they are not already named.

If the "Visible" option is selected, the default name will be displayed on the diagram adjacent to the device or signal when it is created.

Auto-Naming Name Format

The auto-generated name consists of two parts, the fixed prefix and the numeric suffix. The prefix portion is derived from the device's prefix attribute field, if any, or from the design's DevPrefix field. The latter is normally set using the Design Preferences command. The standard prefix field, "Name.Prefix", is empty for most digital parts in the DesignWorks libraries. For discrete components, a commonly-used prefix letter is specified. For signals, the design's SigPrefix attribute field value is used to generate the name.

The numeric portion of the name is generated from the device or signal's "token" value. This value is guaranteed to be unique within a circuit level, but no attempt is made to fill in unused values. Thus, in a design that has been edited, sequential numbering is not guaranteed.

Selecting an Alternate Prefix Field

The prefix used in generating device names is derived from another device field known as the "prefix field". The default prefix field is "Name.Prefix". DesignWorks allows the prefix field to be selected for a given design so that multiple prefixes for different naming conventions can be stored with the same part. E.g. the standard Discretes library included with the package has two prefix fields:

- "Name.Prefix" is the normal one used by default for generating names.
- "Name.Spice" contains prefix values specific to SPICE-based analog simulators.

Its possible to use any attribute field as a device prefix. For example, the standard field "Function" contains a short function code for the part that can also be used as a descriptive prefix: Nand1, Nand2.

The selected prefix field is changed by using the Set Design Attributes command to place the name of the desired prefix field in the "PrefixField" attribute field. See the procedures section later in this chapter for specific steps.

Device and Signal Token Values

Every time a device or signal is created in a DesignWorks circuit, it is assigned an integer value known as its "token". The token number stays with the device or signal for its lifetime and numbers are not re-used. This ensures that a given device or signal can always be recognized despite duplicate names or name changes. The token is used for a number of internal operations in DesignWorks, but can also be seen by the user in the following circumstances:

- The token number is used to generate default names for devices and signals, as described elsewhere is this chapter.
- The token number can be written out in netlists or bills of materials whenever a guaranteed-unique identifier is needed.

Note the following characteristics of tokens:

- Tokens are assigned independently for each circuit in a hierarchical design and are thus only unique within a circuit, not across the entire design.
- When two signals are joined, one of the two tokens is discarded at random.
- Each logical symbol on the diagram (including pseudo-devices) has its own token. In a netlist, several symbols may be combined into a single package, so there is not necessarily a one-to-one correspondence between tokens and physical packages.

Multipage Schematics

DesignWorks allows a circuit to be represented on multiple pages with logical connections between pages. When a new circuit is created, it is assumed to have a single page. A new page is added to the circuit by selecting the Page command in the Drawing menu and clicking on the New Page button.

Sheet Border Setup for Multi-Page Designs

When a new page is added to a circuit, all border information and text and picture objects marked as "border" items are copied automatically from the preceding page (i.e. what was formerly the last page). For this reason, it is most convenient to set up the first page of the circuit with the desired border arrangement before adding other pages.

Borders on all pages in a design can be updated later if needed using the Import Sheet Info command.

Inter-page Connections

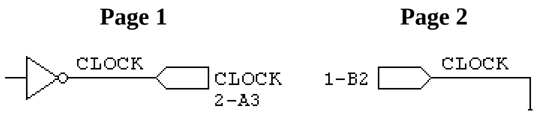
Using the Page Connector Device

Signals drawn on different pages will initially have no logical connection, even if the names are the same. Inter-page connections are made using the Page Connector pseudo-device which is in the Primitives and Connectors libraries. When attached to a signal line, the Page Connector makes the name of that signal known across all pages, i.e. any like-named signal on another page which also has a page connector will be logically connected to this one. The Page Connector can be placed anywhere along the signal line although it is normally convenient to place it at the edge of the page. The Page Connector itself can be named (in order to take advantage of the automatic page references mechanism described below), but it must have the same name as the attached signal. Changing the name of the Page Connector (using the naming procedure for devices) will also change the name of the attached signal. Likewise, changing the name of the signal will rename any attached Page Connectors.

Automatic Display of Page References

DesignWorks has an Automatic Page Reference facility which displays the page number and position of any Page Connectors associated with a given signal. These page references are automatically displayed immediately adjacent to any Page Connector as soon as multiple Page Connectors are attached to the same signal.

For example, in the following case both pages have signals named CLOCK with Page Connectors attached, so the two CLOCK signals are logically connected. Note that both Page Connectors are notated with the page number and grid reference:


Page 1 **Page 2**

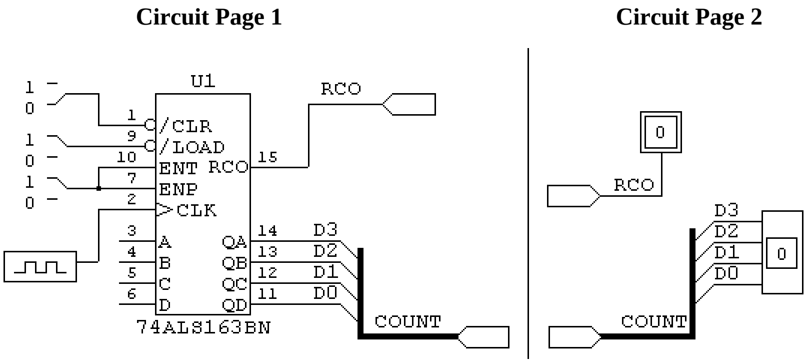
When a name is applied to the Page Connector itself, the page reference will appear under the name.

As long as automatic updating is enabled, these references will be updated automatically when any Page Connectors or attached signals are added, deleted, moved or renamed.

See the Design Preferences command in Chapter IX - Menu Reference for page reference format options and update control.

Connecting Busses Across Pages

Busses can be connected between pages by using the Bus Page Connector. This is illustrated in the following simple circuit:

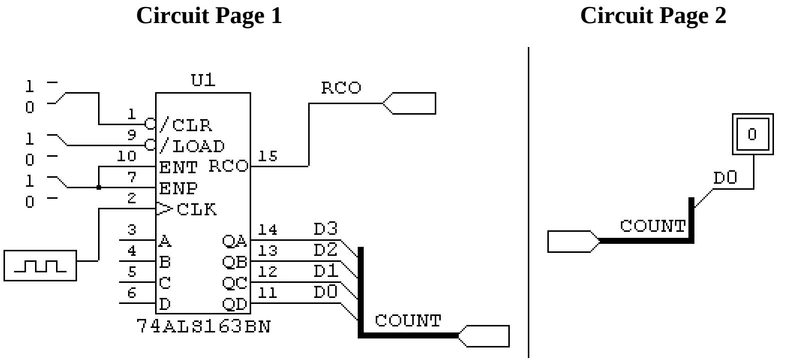

Circuit Page 1

Circuit Page 2

Page Connectors on Internal Bus Signals

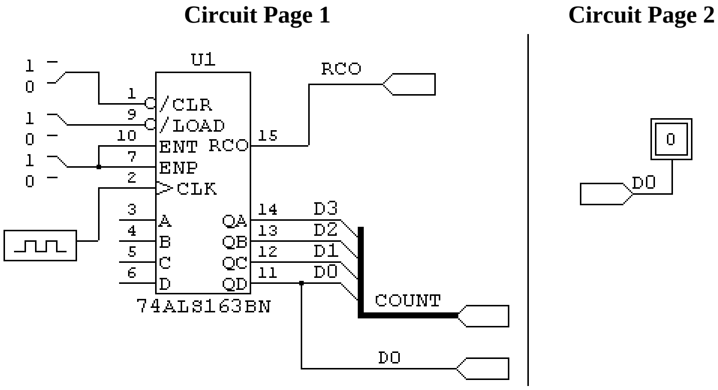
Note that a Bus Page Connector does not make the names of the internal signals known globally, only the bus itself. To bring an individual internal bus signal across to another page, one of the following methods can be used:

1) Make the entire bus global using a bus page connector, then use a breakout to access the desired signal, as in the following:


Circuit Page 1

Circuit Page 2

2) Place signal page connectors on the individual signal on both pages, as in the following:



Changing the Page Connector Symbol

Note that the symbol for the Page Connector device can be changed using the DevEditor tool just as with any other device.

See Chapter X - Device Symbol Editing for specific information on creating Page Connector symbols.

Bussing

The bussing facility allows any combination of named signals to be represented by a single line and any subset of these to be brought out through a "breakout" at any point along the bus line.

Properties of Busses

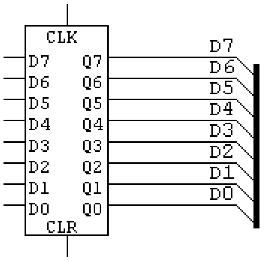
A bus is treated by DesignWorks as a signal with special properties. Thus, bus lines can be drawn and modified on the screen using all the same editing features available for signals. Note the following properties of busses:

- Only bus pins on devices can be connected directly to a bus. All other connections must be made by using a breakout to access the desired internal signals. A breakout is normally created using the New Breakout command in the Options menu. In special cases a user may want to create a non-standard breakout symbol, this may be done using the DevEditor

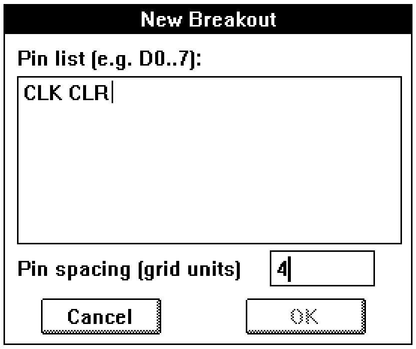
- You do not need to specify in advance what signals will be contained in a given bus. Any signals that are present in a breakout or bus pin attached to a bus will become part of that bus and can be brought out through another breakout anywhere along the bus.
- Any two busses can be joined together, regardless of their internal signals. When two different busses are merged, any signal in either bus becomes available anywhere along the combined bus.
- If you select a bus line, then do a Get Info command from the Options menu the displayed info box will show a list of the signals currently contained in the bus.
- A given signal can be present only in one bus. If you attempt to connect together two signals in different busses, a warning box will be displayed and the connection will be canceled.
- A bus can be created by drawing the bus lines first, then creating the breakouts to attach, or by creating a breakout and extending the bus line starting at the bus pin. Bus lines are drawn or extended using exactly the same techniques as for signals, except that the Draw Bus command or cursor is used instead of Draw Signal.

Properties of Breakouts

Signals are attached to a bus via a special type of device symbol called a "breakout". It is not legal to attach a signal line directly to a bus line and any attempt to do so will elicit a warning box. In DesignWorks, a breakout is treated as a device with certain special properties. This means that it can be placed in any desired orientation, moved, duplicated, etc. using any of the device editing features available. A typical breakout appears as follows:



Any breakout can always be attached to any bus. When a breakout is attached that contains signals unknown in that bus, the signals are implicitly added to the bus. For example, suppose we want to add control signals to the above circuit, we could create a breakout containing only the new signals, as follows:

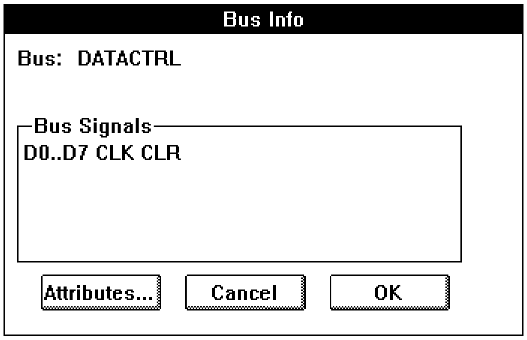

New Breakout

Pin list (e.g. D0..7):

CLK CLR

Pin spacing (grid units) 4

Once such a breakout has been added to the bus, all signals in all attached breakouts are considered part of that bus. A list of internal signals can be seen by selecting the bus and using the Get Info command:

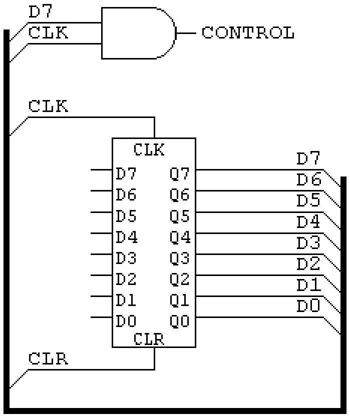

Bus Info

Bus: DATACTRL

Bus Signals

D0..D7 CLK CLR

Any combination of the internal signals can now be brought out of the bus at any point, as in the following addition to the above circuit:



Bus Pins

DesignWorks supports user created bus pins on devices. A bus pin can be defined to have any collection of named internal pins. Note the following properties of bus pins:

- The bus pin itself does not represent a physical device pin. It is only a graphical place-holder on the schematic representing a group of internal pins. The bus pin itself never appears in a netlist.
- The internal pins represent physical device pins. Even though they do not appear on the schematic, they can have all the same parameters as normal devices pins, including pin numbers and attributes. These parameters can be accessed using the Bus Pin Info command in the pin pop-up menu.
- When a device with a bus pin is placed, it has a pre-created bus attached to it by default. This bus will contain one signal for each internal pin, with the initial name of the signal being the same as the name as the pin.
- A "splicing" box can be displayed using the Bus Pin Info command in the pin pop-up menu. This box allows any internal pin to be connected to any signal in the attached bus.
- Bus pins may also display a pin attribute enumerating the bus internal signals that are connected to the bus pin’s internal pins. This feature is enabled by using the Bus Pin Info command in the pin pop-up menu.

For more information on creating device symbols with bus pins, see Chapter X - Device Symbol Editing.

Background and Border Objects

Either text or picture blocks can be set to be "border" or "background" objects. These two options will normally be used together, but can be enabled separately if desired. These two options are enabled by selecting the object in question and using the Get Info command.

Border Objects

If a text or picture is marked as a "border" item, then it is considered to be part of the sheet border information for the page it is on. This has the following effects:

- if a new page is added to the circuit using the Page command, this item will automatically be copied to the same position on the new page.
- if the page containing this item is used as the source for an Import Sheet Info operation, this item will be copied to the destination.
- if a Center in Page command is used, this object will not be repositioned since it is assumed to be part of the border.

Background Objects

Text or picture items marked as "background" objects cannot be selected, modified or deleted using normal circuit editing techniques. This is used to prevent custom sheet backgrounds or border graphics from interfering with circuit editing operations. This is normally used in combination with the "border" status discussed above.

To select a background object, hold the Shift and Control keys pressed while clicking on it. The Get Info command can then be used to disable the background status.

Using Text Variables

The text blocks may contain special entries called variables. When the text is displayed or printed, the appropriate system value or attribute field value is substituted for the variable name. Any variable name that is not recognized will be displayed verbatim. This facility is intended to assist in creating title blocks and similar additions to the schematic with information such as the date, page number and title, revision level, engineer’s name, etc. Using variables will allow this information to be entered in one place so that changes are reflected automatically on all schematic pages.

System Variables

System variables start with a "$" and must be one of the following:

$DATENOW	This will be replaced by the system date written mm/dd/yy. This is the current date as maintained by the hardware clock.
$TIMENOW	This will be replaced by the system time written hh:mm. This is the current time as maintained by the hardware clock.
$DATECREATED	This will be replaced by the date the file was created, as maintained by the file system.
$TIMECREATED	This will be replaced by the time the file was created, as maintained by the file system.
$DATEMODIFIED	This will be replaced by the date the file was last modified, as maintained by the file system.
$TIMEMODIFIED	This will be replaced by the time the file was last modified, as maintained by the file system.
$PAGENUM	This will be replaced by the number of the page (within the current circuit level) the text is on.
$NUMPAGES	This will be replaced by the total number of pages in the circuit level.
$PAGETITLE	This will be replaced by the title of the page the text is on, as entered using the Page command.
$PRINTPAGENUMBER	This will be replaced by the page number in the printed sequence within the current Print Design command. This is used to number pages within a hierarchical design. NOTE: This is only valid for printed output. When it is drawn on the screen, it will show the same value as $PAGENUM.
$PRINTNUMPAGES	This will be replaced by the number of pages that will be printed by the current Print command. This is used to number pages within a hierarchical design. NOTE: This is only valid for printed output. When it is drawn on the screen, it will show the same value as $NUMPAGES.

$FILENAME	This will be replaced by the name of the design file.

Attribute Variables

Attribute variables start with a "&" mark and are used to refer to fields stored in the attributes for the design. These can be used to place information at multiple points on a diagram which will be updated automatically when the design attributes are changed. For example, if the field REV was defined for the design with the following contents:

```
2.1A Mar 18, 1993
```

then the variable &REV would appear as "2.1A Mar 18, 1993" on the diagram.

See the Set Design Attributes command in Chapter IX - Menu Reference.

Editing Text with Variables

Text items on the schematic will normally be displayed with variables replaced by their values. When a text item is selected or clicked on with the Name cursor it is redisplayed in its raw format with the variable names shown without interpretation. This allows the items to be edited with the text in its actual stored position.

Text Frame Size with Variables

The framing rectangle for a text item is calculated after the variable substitution is done. This may cause the item to be highlighted or deleted incorrectly if the variable values are shorter than the names. These errors are not serious or permanent and will disappear when the screen is updated. The text box can be expanded as necessary by adding blanks at the end of any one line in the item.

Power and Ground Connections

Device Power and Ground Connections

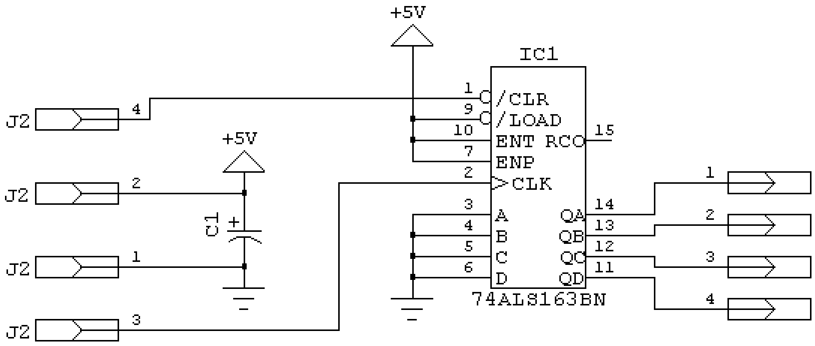
When a schematic diagram is created, the power and ground connections for devices are normally not drawn as signal connections on the diagram. These connections would clutter the diagram and are not necessary for an understanding of the logical function of the circuit. Obviously, these connections must be included at some level in order to form a complete netlist. Three methods are available to do this:

- The power and ground pin connections can be specified as attributes for each device. The Schematic tool itself attaches no significance to the attribute fields, but the report generator reads them and adds the specified pins to the netlist. The libraries of standard types supplied with DesignWorks include this information for each type. It is only necessary for you to enter this data if you are making non-standard connections or using a device that has no default power connections. The pre-defined attribute fields "Power" and "Ground" are intended for this purpose.
- Power and ground pins can be added to each device symbol (using the DevEditor tool) and the connections shown as normal signal lines on the diagram. The standard DesignWorks libraries are not set up in this fashion.
- A separate symbol can be created for each device showing only power and ground connections. This allows all power wiring to be shown in one part of the diagram or on a separate page.

In fact, these methods are not mutually exclusive and can be used in combination as needed. The Report tool implements these methods by combining connections described in different ways into a single net in the netlist:

- Any number of special names can be specified as "signal sources" using the $SIGSOURCE keyword in the report form file. The Report Generator searches the attributes attached to each device for fields with these names. The pin number specified in the field will be attached to a list for the signal of the same name. E.g. the value "14" in the Power field will cause pin 14 on this device to be attached to the signal named as the power signal. Multiple pins can be specified using commas, e.g. "1,2,14".
- A Signal Connector device (such as the Ground and +5V symbols) can be used to create a logical connection between pins. See more information on Signal Connectors below.

These features will be illustrated by way of a simple example:



The attributes for the device IC1 were specified in the library entry for type "163" and are as follows:

```
Power     14
Ground    8
```

In the above diagram, the signal named "+5V", the symbols labeled "+5V" and the attribute entries "Power" will all be merged into a single list of connections in the netlist, as follows:

```
S00005    IC1-14    J1-1
S00007    IC1-2     J2-3
S00008    IC1-1     J2-4
S00009    IC1-11    J1-4
S00010    IC1-12    J1-3
S00011    IC1-13    J1-2
Ground    C1-2      IC1-3,4,5,6,8    J2-1
+5V       C1-1      IC1-10,7,9,16    J2-2
```

NOTE: The standard report form files add the entries in the Power field to a net named "+5V".

The demonstration files provided with DesignWorks provides more complex examples of this usage.

Power and Ground Naming Convention

The devices in the standard libraries provided with DesignWorks include the attribute fields Power and Ground for the normal power and ground pins. Since field names cannot contain special characters, we recommend using the words "Plus" and "Minus" to create attribute fields for other supplies. For example, pins to be attached to a -12V supply would be listed in field "Minus12V". The Report Generator allows the name of the signal net to be different from the field name, if desired.

See Chapter XI - Report Generation for instructions on adding other power nets.

Signal Connector (Power and Ground) Symbols

DesignWorks uses a type of pseudo-device symbol called a "signal connector device" to maintain connectivity between like-named power and ground symbols that are used on circuit diagrams.

As soon as a Ground symbol is placed on the diagram, the attached signal will be named "Ground" (the name will be invisible initially). This will cause it to be connected by name to any other signals having Ground symbols, or explicitly named "Ground".

Connectivity can be checked at any time by double-clicking on any ground or power line. This will highlight all other like-named lines on the diagram.

IMPORTANT: Signal connectors do not cause a logical connection to be made between hierarchy levels in a hierarchical design. These signals can be merged into a single net for netlisting purposes by specifying a $SIGSOURCE statement in the report form file. However, this processing is done as the report is generated and does not create a logical connection in the design, e.g. for interactive simulation.

See Chapter XI - Report Generation for more information.

Using Signal Connector Devices

Signal Connector Devices are placed on the diagram just like any other DesignWorks device. A set of standard power-supply symbols are included with DesignWorks in the Connectors library. If you connect two different signal connector devices together, you will be prompted to provide a name for the resulting signal. This should normally be interpreted as a error and the operation should be canceled.

Creating Signal Connectors in a Library

Signal Connector devices (like Page Connectors) are special primitive "pseudo-devices" in DesignWorks and can be created using the Set Primitive Type command in the DevEditor tool to select the SIGCONN primitive type.

NOTE: The signal attached to a signal connector device is actually named to match the pin name of the signal connector pin specified in DevEditor, not the type name. In most of the power and ground symbols provided with DesignWorks, these two names are the same. However it is possible to create a symbol called, for example, "Ground" in the library which actually names the attached signal "GND". The Ground symbol in the SPICE Devices library provided is an example of this in that it names the attached signal "0" to match the SPICE ground naming convention.

See Chapter X - Device Symbol Editing for more detailed information on this procedure.

Connectors and Discretes

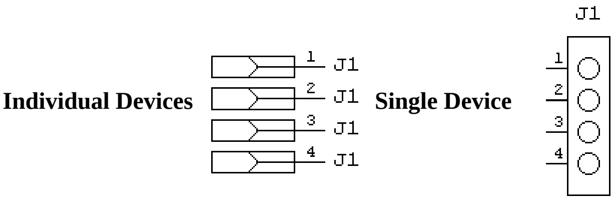
In DesignWorks, each symbol is considered to be a separate device and each device is normally assumed to be one IC package with a standard pin numbering scheme. Thus connectors and discrete components will require special consideration.

Handling Connectors

Connectors can be handled in one of two ways:

- A special symbol can be created for the connector with the appropriate number of pins and pin numbering specified for each pin. This can be done using the DevEditor tool to create a device symbol using your own picture.
- Each connector pin can be created as a separate, 1-pin device using the symbols from the Connectors library or a custom symbol. This option is preferable only if you need to spread the connector pins over different parts of the diagram. In this case, each "device" must be given the name of the connector and the pin number associated with that pin. We are relying on the fact that the report generator will normally merge all devices with the same name into a single component entry. Attributes (if any) are taken from the first device encountered with the given name.

Following is an example of these two methods:



NOTE: When the single-pin devices are used, every device must carry exactly the same name, although the names can be invisible if desired.

Handling Discrete Components

Discrete components such as capacitors, transistors, etc. can be handled just like any other device except for the following special considerations.

Pin Numbering on Discrete Components

Pin numbers are not normally placed on discrete component pins on a diagram. If pin numbers are omitted from a device, DesignWorks will normally put a question mark in the netlist item for that device. Two methods are available to provide pin numbers for netlisting purposes:

• To provide automatic numbering of discrete devices pins, the Report Generator provides the AUTONUMBER option, which can be specified in the form file. This option causes any device with less than or equal to the given number of pins to be numbered automatically if no pin numbers are present on the diagram. In the standard form file, this value is set to 3 to cover the common discrete components.

WARNING: This option assumes that the pin number order of the discrete components is not significant. If a specific order is important, do not use this method.

• Pin numbers can be assigned but left invisible. This is done using the Get Info command for either the pin or the device.

Package Types for Discrete Components

Discrete components do not normally have standard package types. If a package type is required for PCB layout purposes, then the "Package" field in the attributes must be set manually for each device.

Advanced Editing Procedures

The purpose of this section is to give you a quick outline of commonly-used schematic editing techniques in DesignWorks. Background on each of these techniques is provided earlier in this chapter and in other reference sections of the manual.

Adding a New Page to a Circuit

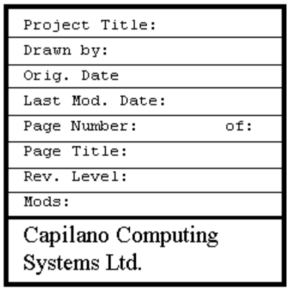
To add additional circuit pages to a circuit, select the Page command in the Drawing menu. The box that appears has controls allowing you to add, delete, re-order and add titles to pages. Clicking on the New Page button will add a new, empty page to the current circuit.

See the Page command in Chapter IX - Schematic Menu Reference.

Creating a Title Block

DesignWorks has a number of features which can assist in creating title blocks which will contain information that is automatically updated. A simple title block can be created using the random text feature.. Click the pencil cursor in the desired location on the diagram and enter the desired text. After the text has been typed, click on the text block to select it, then use the Get Info command to set the text size appropriately and create a frame.

A more striking title block can be created using a graphic item imported from the DevEditor, such as the following:



Project Title:

Drawn by:

Orig. Date

Last Mod. Date:

Page Number: of:

Page Title:

Rev. Level:

Mods:

Capilano Computing
Systems Ltd.

First, paste the graphic on the diagram using the Paste command then select it and set it to be a background and border object using the Get Info command. Once it has been converted to a background object, it will not be selected again and you can use text notations to draw on top of it. Text variables can be used to add dates and titles to the block. See the section on variables earlier in this chapter.

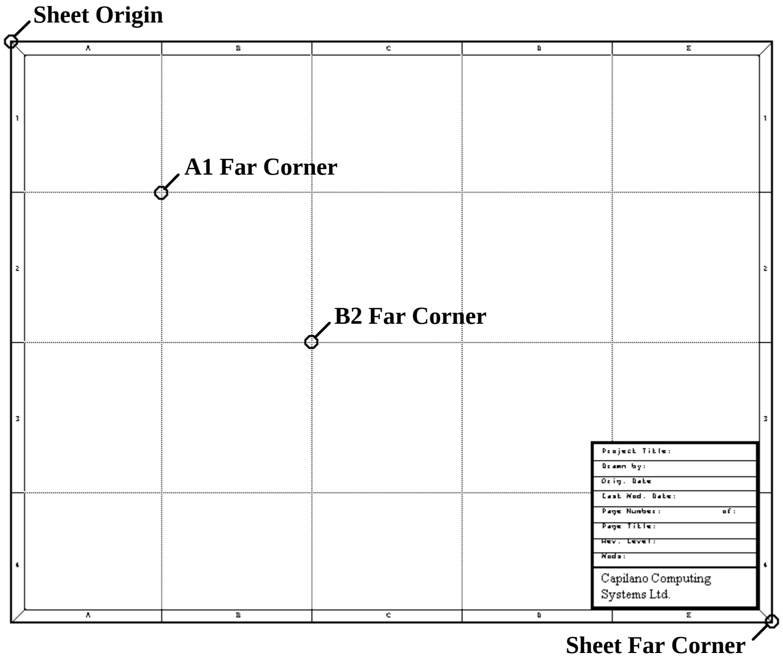
NOTE: Background objects can be selected by holding the Shift and Control keys while clicking on them.

Creating a Custom Sheet Size

A custom fixed sheet size is one that does not vary with the current Page Setup printer settings. In order to have a complete description of a sheet, you need to indicate to the program the following points:

Point #1 - The sheet origin. This is the corner of the sheet where the A1 grid box is located. The origin point will be on the outside line of the border. Drawing will not be allowed outside this point.

Point #2 - The opposite corner of the A1 grid rectangle, i.e. where it meets the B2 grid rectangle. This point is used to determine the start of the regular drawing grid.

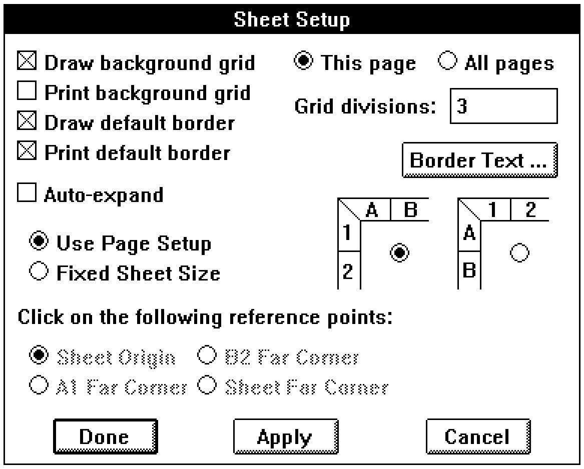


Point #3 - The opposite corner of the B2 grid rectangle. The space between Points 2 & 3 determines the grid spacing for the rest of the sheet.

Point #4 - The far corner of the sheet. The point indicated will be the outside line of the border. Drawing will not be allowed outside this point.

> NOTE: The sheet origin can be in any corner. It is the corner that will be the origin (i.e. the "A1" box) for the reference grid.

Select the Custom Sheet Info command in the Drawing menu. The Sheet Info palette will appear as follows:



Sheet Setup

☒ Draw background grid
□ Print background grid
☒ Draw default border
☒ Print default border
□ Auto-expand

● This page ○ All pages

Grid divisions: 3

Border Text ...

● Use Page Setup
○ Fixed Sheet Size

Click on the following reference points:

● Sheet Origin ○ B2 Far Corner
○ A1 Far Corner ○ Sheet Far Corner

Click on the Fixed Sheet Size option. You will see the Sheet Origin button has become enabled. This indicates that you should scroll to the sheet origin position and click accurately on the origin point.

You will now notice that the A1 Far Corner button has been automatically selected. Click now at the opposite corner of the A1 grid rectangle.

Next, locate and click on the B2 far corner and sheet far corner points.

Select any other sheet option boxes desired, then click on the Done button. The sheet info palette will disappear and the displayed drawing area will be updated.

Creating Custom Sheet Border Graphics

Any picture or text objects can be made part of the sheet border. To do this:

- select the object.
- choose the Get Info command in the Options menu.
- enable the "Make Background" and "Make Border" switches.
- click OK.

The object will now be treated as part of the sheet background. To select it for deletion or editing, hold the Command and Option keys while clicking on it.

Controlling Device Packaging

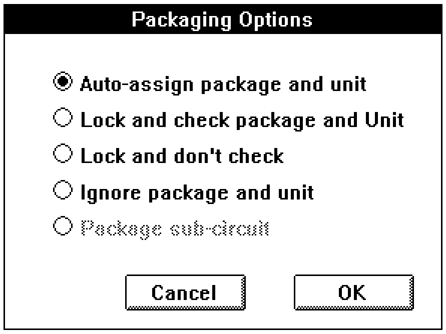
A device is packaged depending on the hierarchy mode and the packaging flags for the device. In flat mode, packaging only occurs on the top level of the design. In pure mode the design is not packaged. In physical hierarchy mode a device is packaged if it doesn’t have a sub-circuit or flag indicating not to package it.

To control a device’s packaging options, bring up the Device Info box by either one of the following methods:

- Select the device by clicking on it, then select the Get Info command in the Options menu.
- Click the right mouse button on the device, then select the Device Info command in the pop-up menu.

Next, click the Packaging Options button in the Device Info box:

This displays the following box:



Packaging Options

● Auto-assign package and unit
○ Lock and check package and Unit
○ Lock and don't check
○ Ignore package and unit
○ Package sub-circuit

Following is a quick summary of the meanings of these options. For complete information see Device Packaging earlier in this chapter.

Normal (Auto-assign package)	Name and unit number will be assigned automatically by the Packager.
Lock and Check	Name and unit will not be assigned, but the present values will not be assigned to any other device. The device will be checked for packaging errors.
Lock and Don’t Check	The same as Lock and Check except that the package assignment is not checked for errors.

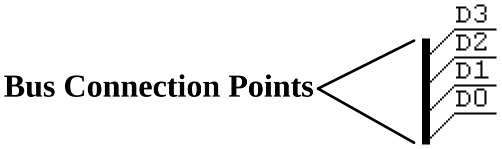
Ignore	Name and unit are not changed or checked and may be assigned to other devices.
Package Sub-circuit	Ignore this device symbol and package devices in its internal circuit. This option will be disabled if the symbol has no internal circuit.

Bus Operations

Creating a Bus

A bus can be created by any one of the following methods:

- Select the Draw Bus tool (✚) in the tool palette. Draw any desired contiguous set of lines on the diagram using the usual signal drawing techniques. This bus will have no internal signals initially. Signals will be added implicitly when it is connected to any breakout or bus pin.
- Create a breakout symbol using the New Breakout command (see below). The bus pin (backbone) of the breakout can now be extended using the normal pointer (↖) or the Draw Bus cursor. The bus will contain all signals specified in the breakout.



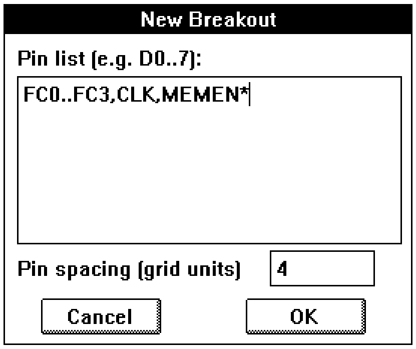
- Extend a line out from an existing bus pin on a device (see below) using the normal pointer or the Draw Bus cursor. The bus will contain all signals specified in the bus pin on the device. Connections between bus internal pins and bus internal signals can be changed using the Bus Pin Info command on the bus pin’s pop-up menu.

Adding Signals to a Bus

There is no explicit command to add signals to a bus. Signals are added to a bus each time a breakout or device bus pin is connected to the bus. Any signals in the breakout or bus pin are implicitly added to the bus if they don’t exist already.

Creating a Breakout

To create a breakout, select New Breakout command in the Options menu. If the new breakout is to be similar to an existing one, first select the similar breakout or the bus to which the new breakout is to be connected. The following dialog box will appear:



Pin list (e.g. D0..7):

FC0..FC3,CLK,MEMEN*

Pin spacing (grid units) 4

If a bus or breakout was selected on the circuit diagram then the breakout info box will display a list of the signals in that bus or breakout, otherwise it will be empty. If this list already matches the signals you want in the new breakout, then just click the "OK" button or hit Enter on the keyboard. Otherwise, edit the signal list, noting the following options:

- blanks or commas can be used to separate individual names in this list, therefore bussed signals cannot have names containing a blank or comma.
- a range of numbered signals can be specified using the following formats:

  D0..7 or D0..D7

  is equivalent to

  D0 D1 D2 D3 D4 D5 D6 D7

  D15..0

  is equivalent to

  D15 D14 D13 D12 D11 D10 D9 D8 _ D0

  D15..D00

  is equivalent to

  D15 D14 D13 D12 D11 D10 D09 D08 D07 _ D00

  Note that the ".." format implies that bussed signal names cannot contain periods.

- the signals specified will always appear in the order given in this list from top to bottom in standard orientation. We recommend always specifying numbered signals from lowest numbered to highest, as in the first example above, since this matches the standard library symbols.
- there is no fixed limit on the number of signals in a bus, but we recommend dividing busses up by function (i.e. address, data, control, etc.) for ease of editing.
- any combination of randomly-named signals can be included in the list, as in the following examples:

  D0..15 AS* UDS* LDS*

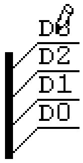
  CLK FC0..3 MEMOP BRQ0..2

Once the list has been entered, click on the OK button or hit the Enter key. A flickering image of the breakout will now follow your mouse movements and can be placed and connected just like any other type of device.

Editing Pins on a Breakout

The signal name notation that appears on a breakout pin is actually a pin attribute. It can therefore be edited by the usual attribute editing mechanisms, i.e. either:

- Select the pin and choose the Get Info command in the Options menu, then click the Attributes button, OR,
- Click the text cursor directly in the text on the schematic, as follows:



- Type the desired new name.
- Press the Enter key. The breakout pin and the attached signal will be renamed as entered.

IMPORTANT: The notation on the breakout pin is always the same as the name of the attached signal name. Changing the breakout pin renames the attached signal and will detach it from any like-named signals already in the bus.

Changing Bus Pin Connections

When a bus is connected to a bus pin on a device or hierarchical block, the bus internal pins will by default connect to signals with the same name in the bus. To change these default connections, use the Bus Pin Info command in the pin pop-up menu.

See Chapter IX - Menu Reference for more information.

Controlling Text Styles

Pin Numbers

The text style for pin numbers is set globally for the entire design. It cannot be set individually for pins.

To set pin number text style:

1) Select the Design Preferences command in the Drawing menu.
2) Click on the Pin Text button.
3) Select the desired text font, size and style in the style box.
4) Click OK on the text box, then OK on the Design Preferences box.

Depending on the size of the design, there may be a short delay at this point while sizes and positions of text items are recalculated.

Attributes

The text style for attributes is set globally for the entire design. It cannot be set individually for each item or for each field.

To set attribute text style:

1) Select the Design Preferences command in the Drawing menu.
2) Click on the Attr Text button.
3) Select the desired text font, size and style in the style box.
4) Click OK on the text box, then OK on the Design Preferences box.

Depending on the size of the design, there may be a short delay at this point while sizes and positions of text items are recalculated.

Border Text

Text style can be set for the reference letters and numbers appearing in an auto-generated sheet border. To do this:

1) Select the Custom Sheet Info command in the Drawing menu.
2) Click on the Border Text button.
3) Select the desired text style.
4) Click OK on the text style box, then Done on the border info box.

Text Blocks

Text style can be set individually for each text block. To do this:

1) Select the text block by clicking on it. (If it is a background item, then hold the Shift and Control keys while clicking on it.)
2) Select the Get Info command in the Options menu.
3) Click on the Text Style button.
4) Select the desired text style.
5) Click OK on the text style box, then OK on the text info box.

Controlling the Page Reference Format

To change the page reference format, select the Design Preferences command in the Drawing menu. Two aspects of the page reference format can be controlled:

- the "Format" text item controls the appearance of each page reference in the reference list.
- the "Max. Width" item controls the number of references that will appear on each line before creating a new line. For signals with many connections, this prevents unwieldy page reference strings.

Three characters are special in the format string:

P	will be replaced by the page number
X	will be replaced by the X grid position
Y	will be replaced by the Y grid position

The default format is "P-XY" which generates pages references like "2-C4" (i.e. page 2, grid position C4). All characters other than the special ones above will be placed in the page reference list verbatim.

The maximum length of a format string is 16 characters.

Controlling Prefix Selection for Device Naming/Packaging

The design attribute "PrefixField" is used to determine which device attribute field is used to generate default names for devices as they are placed. The standard prefix field is "Name.Prefix". Two other fields are included in the standard DesignWorks libraries which can be used as prefix fields:

- "Name.Spice" is defined in the Discretes library and contains the prefix commonly used by SPICE-based analog simulators to identify part type.
- "Function" contains a short function code for each discrete and digital part. It can be used as an alternate prefix field to give descriptive names to parts as they are placed.

To set the prefix field:

1) Select the Set Design Attributes command in the Options menu.
2) Select the PrefixField item in the field list.
3) Type the name of the desired prefix field in the data box. IMPORTANT: The name must match an existing field name exactly, including case, or it will be ignored.

If a device doesn’t have the referenced prefix field then the default prefix is used. For device auto-naming this is the value in the design attribute "DevPrefix". For device packaging, ie: the packager is enabled, the default name is stored in "PkgPrefix".

Chapter VI - Attributes

DesignWorks allows arbitrary blocks of text to be associated with any device, signal or pin in a design, or with the design itself. The blocks of text are called *attributes.* Attributes have a wide variety of uses, including:

- displaying device name, package assignment, part type, value, etc.
- storing manufacturing information such as catalog numbers, manufacturer’s name, price, part description, etc.
- storing data for use by external systems such as simulators, PCB layout, analysis tools, etc.

In addition, many DesignWorks features use attributes as temporary or permanent storage locations for data relating to a device or signal. For example, each device in a design has the name and location of the library it was read from stored in an attribute field. This allows the library to be located easily for later updating.

Attribute Organization

Attribute Definition Table

Attributes are divided into named *fields*. The list of allowable field names is stored with each design in the *attribute definition table.* For each field, the table contains:

- the field name (up to 16 characters)
- what types of objects the field can be used with (devices, signals, pins or design)
- data entry constraints (maximum length, carriage returns, etc.)
- default settings, such as whether the field should be made visible on the diagram.

Once a field is defined in the table, its name is never typed again. When attribute data is entered for a selected object, the field name is picked from a list of allowable fields. This eliminates any chance of accidentally misspelling the field name in one object.

Predefined Fields

When a new design is created, a default attribute definition table is created with the standard, predefined fields. Predefined fields cannot be deleted from the design since many of them have specific internal functions in DesignWorks.

> *See Appendix C - Predefined Attribute Fields for a complete listing of predefined fields.*

User-defined Fields

You can add more fields to a design at any time using the Define Attribute Fields command in the Options menu. Commonly used fields may also be added to the design by modifying the "DW.ini" file.

Primary vs. Secondary Fields

DesignWorks has 40 or more predefined fields that are used for various program features and more can be added by the user. Most of these fields never need to be viewed or set directly by the user. To reduce clutter, attribute fields can be defined as being *primary* or *secondary.* When entering attribute data, you can choose whether you wish to see "primary fields only" or "all fields". The Attribute Probe tool (?) in the tool palette shows only primary fields.

The primary/secondary setting is for convenience only and has no effect on the meaning or internal usage of the field. This setting can be changed at any time using the Define Attribute Fields command in the Options menu.

Definition vs. Instance Fields

In a hierarchical design, a single device, pin or signal in a sub-circuit may actually represent several physical objects. This happens if the parent device containing the sub-circuit is used more than once in the design.

For this reason a distinction is made between fields that are associated with the definition of a sub-circuit and fields associated with the instance.

- Fields associated with the definition will be the same in all instances of the sub-circuit. Changing the value in one instance will affect all instances equally.
- Fields associated with the instance can take on a different value in each physical instance of the device, pin or signal. Changing the value in one instance has no effect on the others.

When defining a new attribute field for use in a hierarchical design, careful consideration should be given to whether to consider it a definition or instance field. If the value of the field has a significant impact on the logical functioning of the sub-circuit, then the field should probably be a definition field. Thus, if two sub-circuits require different values in this field, then separate definition circuits will have to be created. This makes it clearer to other users of the design that a distinction exists.

See Chapter VII - Hierarchical Design for more information.

Name and InstName Fields

In DesignWorks, the single most important attribute field is the "Name" field. It can be associated with devices or signals and is used to uniquely identify an object within its circuit. The Name is a *definition* field.

Note that the Name field is unique within a circuit, not necessarily within the entire design. For this reason, the InstName field is used in hierarchical designs. InstName is an *instance* field, meaning that it can have a different value for every physical device in a hierarchical design. See the section "Names in Hierarchical Designs" below. The InstName field is not normally used in Flat mode designs.

Names may contain any letters, numbers, "_" (underscore) or "." (period) characters. They are restricted in length to at most 15 characters. The name associated with an object can be placed anywhere on the diagram and will be removed if the object is removed.

The Name or the InstName fields can both be set using the pencil tool. To control which field is modified by the pencil please refer to the Design Preferences dialog.

Invisible Names/InstNames

Like other attribute fields, the Name/InstName field can be associated with a device or signal without displaying it on the diagram. The visibility of the name can be set using the Get Info or Attributes command associated with the device or signal in question.

NOTE: When signals are connected by name, only visible names affect connectivity. Two signals with the same invisible name will not be connected. See Chapter IV - Schematic Editing for more details.

Device Names/InstNames

Typical device names might be "U23", "C4", "XTAL1", etc. The Name field for devices is equivalent to the "reference designator" used in other systems. The InstName is equivalent to a name associated with a physical instance of the device. Many other systems have no direct way of representing this.

The Name/InstName are distinct from the "type name" that is used to distinguish the type definition that is read from a device library. The "type name" is not an attribute, but a fixed name associated with a device symbol when it is saved in a library. Typical type names are "74LS138", "MC68000L8", "SPDT Switch", etc.

Normally only one of Name or InstName appears on the diagram as a text notation, the other is invisible. In addition to being a text notation, the device Name and InstName fields are used by the following DesignWorks functions:

- if the design is in flat hierarchy mode, then the Packager tool places the assigned package name in the Name field. In other modes, the "InstName" field is used.
- the device name is used in Report Generator output, such as netlists and bill of materials reports.
- the device can be located by name using the Find tool.
- the device name associated with a Page or Port Connector is used to make logical connections.

Signal Names

Typical signal names are "CLOCK", "ADDR12", etc. Signal name is referenced by the following DesignWorks functions:

- the signal name is used in Report Generator output, such as netlists.
- the signal can be located by name using the Find tool.
- signals can be logically interconnected by name.

See more information on signal names and logical connections in Chapter IV - Schematic Editing.

Names/InstNames in Hierarchical Designs

In a hierarchical design, a single device in a sub-circuit may actually represent several physical devices. This happens if the parent device containing the sub-circuit is used more than once in the design.

See Chapter VII - Hierarchical Design for an example of this situation.

For this reason another attribute field is needed that will be unique for each and every physical instance of a device in the design. The "InstName" field serves this purpose.

If the design is in "physical" hierarchy mode, the InstName field is used by the Packager to store the package assignment. If the Packager is not being used, the InstName field can be used to store notations about the physical placement of the device.

NOTE: In a hierarchical design, the Name field can be thought of as a "logical" name and the InstName field as a "physical" name. The "logical" name should be assigned by the designer based on the device's function in the circuit, not on its physical location.

The visibility of the Name and InstName fields on the diagram can be set separately. It may be desirable to only show the Name field on the diagram during the early stages of design development. The physical package assignments can be shown later when needed for manufacturing.

.List Fields - Value List Fields

For attribute fields that can only take on one of a known list of values a *value list field* can be created. This mechanism is used to create the "Unit" and "Part" fields used in all standard DesignWorks libraries. These standard fields can be used as an example.

A value list field has the same name as the associated field but with ".List" appended. When a field named "xxx.List" is seen by the program, it does the following:

- when the attribute edit box is displayed for a given object and the field "xxx" is selected, a pop-up list is displayed containing all the possible values for the field. The user may then select the appropriate value from the list.
- when a pop-up menu is displayed by the user clicking the right mouse button, a sub-menu with the field name "xxx" as its title will appear. The list of possible field values will be displayed in the sub-menu. The user can select one of these items to directly change the value of field "xxx".

NOTE: DesignWorks does not prevent the user from entering a field value that is not in the list.

Value List Data Format

The value list field must contain a sequence of textual values following these rules:

- items must be separated by commas. A comma will always be taken as a separator and therefore cannot be part of the value.
- the value string can contain any characters other than commas, but leading and trailing blanks will be removed.
- the length of a value string is limited to 16 characters.
- the maximum number of values is limited to 256.

Normally, a default value for the "xxx.List" field should be defined with the symbol in the library using the DevEditor tool. In this way, the value list will always be available whenever this device type is selected from the library. If the field is defined in only one device on the diagram, the value list will only be available for that device.

.Pt Fields - Default Position

For attribute fields that are to be displayed on the schematic, an associated field called a "default position field" can be created. This mechanism is used to provide default positions for the "Name" and "Part" fields used in all standard DesignWorks libraries. These standard fields can be used as an example.

A default position field has the same name as the associated field but with ".Pt" (for Point) appended. When a field named "xxx.Pt" is seen by the program, it uses the values in this field to generate a default position for the "xxx" field.

Default Position Data Format

The default position field must contain a sequence of textual values following these rules:

- two items, an X and a Y position, must be specified, separated by a comma. Leading an trailing non-numeric characters are ignored.
- the X and Y values are each decimal numbers representing an offset from the top-left corner of the device’s bounding box measured in 1/1000". Negative values are allowed (negative is up and left).

- the X coordinate can be optionally followed by a letter indicating horizontal justification, as follows: "L" for left, "M" for middle, "R" for right. I.e. the given number specifies the offset to the left, middle or right position of the displayed text. If no letter appears, middle is assumed, i.e. the text will be centered at the given position.
- the Y coordinate can be optionally followed by a letter indicating vertical justification, as follows: "T" for top, "M" for middle, "B" for bottom. I.e. the given number specifies the offset to the top, middle or bottom position of the displayed text. If no letter appears, middle is assumed, i.e. the text will be centered at the given position.

Some examples of value points are:

0R,0T — The value will extend right and down from the top-left corner of the device.

150,-75 — The value will be centered at a position above and to the right of the top-left corner.

(400L,100) — The value will extend left from a point to the right of the top-left corner of the device. (Leading and trailing characters are ignored.)

Normally, a default value for the "xxx.Pt" field should be defined with the symbol in the library using the DevEditor tool. In this way, the value will be available for setting the default position of the associated field when the device is placed on the schematic.

Default Values

A device symbol can have associated with it predefined default values for any number of fields. Values can be specified for the device itself, and independently for each pin on the device.

When the standard attribute data box is displayed for a device, you will see a button labeled "Use Default Value". If this button is grayed out (disabled), then there is no default value or the value shown is already the default.

See Chapter X - Device Symbol Editing for more information on creating default attribute values.

Attribute Field Options

When a new field is created using the Define Attribute Fields command a number of options are available which determine how that field will behave in your design. These options are described in the following sections.

IMPORTANT: Changing any of the following options in a field that has already been used in the design has no effect on items already displayed on the schematic.

TIP: To make a temporary change (e.g. to display a field that is not normally displayed) you can use the Duplicate button in the Define Attribute Fields box to create a new field containing the same data but with new display options. This field can be deleted again when no longer needed.

Keep with Instance

This option applies only in Physical Hierarchy mode. When this switch is checked, the selected attribute field may take on a different value for each instance of a device, pin or signal.

See Chapter VII - Hierarchical Design for more information on instance data.

Group with Name

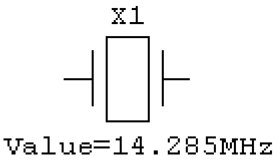
When this switch is checked, any value entered for this field will be displayed beneath the Name/InstName field by default. When the Name/InstName is moved, this field will also be moved. The attribute is grouped with the Name field in Flat and Pure hierarchy modes and with InstName field in Physical hierarchy mode.

This option can be used for fields that are typically shown with the name on a schematic. E.g. the Unit field (gate unit) has this setting enabled by default.

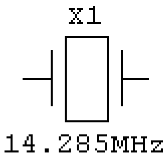
Show Field Name

When this switch is checked, the field name will be shown with the value on the schematic.

With
Show Field Name



Without
Show Field Name



The Show Field Name option can also be controlled for each individual item on the schematic using the Show Field Name command in the attribute pop-up menu.

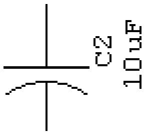
Allow Carriage Returns

If this switch is checked, then the Enter key may be used to enter Carriage Return characters into this field.

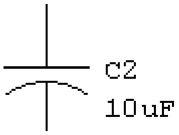
Rotate Text With Object

If this box is checked, displayed values will be rotated if the associated object is rotated on the schematic. Otherwise, when the object is rotated, the text will move to an appropriate new position but be drawn in the normal orientation.

Rotate
With Object



No Rotate
With Object



Visible by Default

If this box is checked, whenever a value is entered for this field it will be displayed on the schematic in a default position. If the field is found in the default attributes for a device, the field will be displayed automatically on the diagram when the device is placed.

> NOTE: Changing this setting has no effect on values already in the circuit. See the note at the front of this section.
>
> NOTE: The Name and InstName fields may ignore the settings of the Visible by Default option because if auto-naming is enable its settings take precedents.

In Primary List

If this box is checked, the field will be considered a "primary" field and will be displayed in all attribute entry boxes when "Primary Only" is selected. It will also appear in the window that is displayed by the Attribute Probe (?) cursor.

Allowed in Object Types

This group of check boxes allows you to select which types of schematic objects may have this field associated with them. You can check any one or more boxes.

Maximum Length

You can specify a maximum length (in characters) for the field data.

Attribute Limitations

Attribute fields have the following specific limitations:

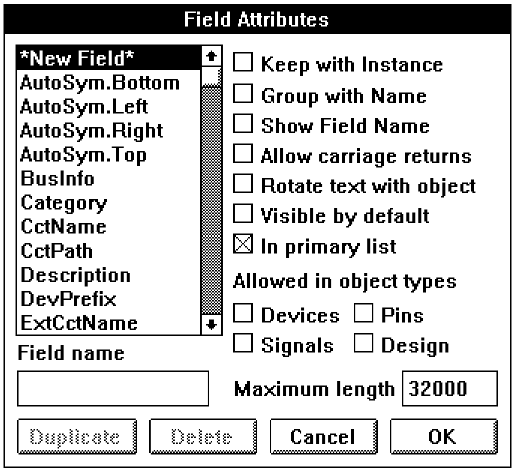
- Maximum length of field name: 15 characters
- Maximum length of field data item: 32,000 characters. Note: In principle, binary data could be pasted into an attribute field although this is not recommended since DesignWorks provides no mechanism for displaying or editing this data.
- Maximum number of user-defined fields: 900
- Maximum number of displayed positions of a single attribute item: 100

Like all other circuit data, the amount of attribute data that can be associated with a design is limited by available memory.

Attribute Procedures

Defining a New Field

All changes in field definitions are done using the Define Attribute Fields command in the Options menu. Selecting this command displays the following dialog box:



Field Attributes

New Field
AutoSym.Bottom
AutoSym.Left
AutoSym.Right
AutoSym.Top
BusInfo
Category
CctName
CctPath
Description
DevPrefix
ExtCctName

Field name:

☐ Keep with Instance
☐ Group with Name
☐ Show Field Name
☐ Allow carriage returns
☐ Rotate text with object
☐ Visible by default
☒ In primary list

Allowed in object types
☐ Devices ☐ Pins
☐ Signals ☐ Design

Maximum length: 32000

To create a new field:

1) Select the •New Field• item in the list, if it is not already.

2) Type the name of the field in the Field Name box.

3) Select one or more of the boxes under "Allowed in object types".

4) Optionally enter a maximum length.

5) Optionally enable any of the other field options. NOTE: These options affect only new data as it is entered, not values already in the design.

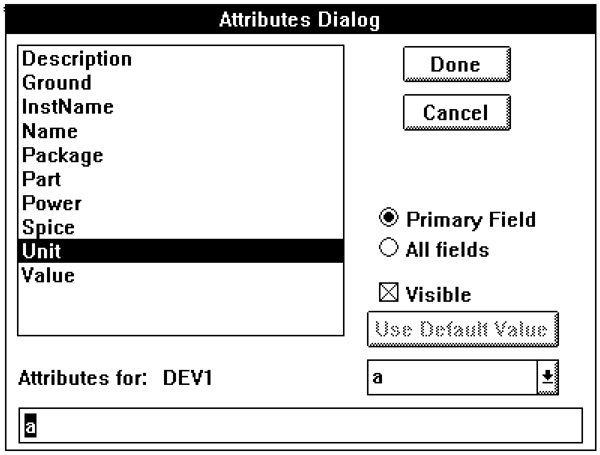
6) Click OK. The new field will now appear in the attribute data entry box for the selected object types.

See the description of the Define Attribute Fields command in Chapter IX - Menu Reference for details on the field options.

Editing Attribute Data (General)

NOTE: The same attribute edit box is used to enter data for all object types. This section discusses the general operation of this box. The following sections discuss each object type.

The following edit box is used to enter or edit attribute data:



● Primary Field
○ All fields
☒ Visible

Attributes for: DEV1

a

a

Basic Procedure

To edit the contents of a field, simply select the field name in the list. The current contents of the field will be displayed in the editable text box. Edit this value using the usual text editing techniques. Press the Done button or select another field when done. If the data you typed exceeded the maximum length for the field, or if it contained invalid characters for the field, then you will be asked to correct the data.

You can view or edit as many fields as desired while in this box. No changes are made to the actual design data until you click the Done button. Clicking Cancel will abandon all changes made while in this box.

Default Value

Clicking the Use Default Value button sets the value for the selected field to the default value stored with the symbol’s definition. If this button is inactive (grayed out) then the value is already the default value, or no default value is present. Only devices and pins can have default values.

Value List Pop-up List

If the selected field has an associated value list field (.List), then a pop-up value list will appear just below the "Use Default Value" button. An item selected out of this menu will cause the value text box to be updated accordingly. This does not prevent the text data from being edited directly.

Editing Attribute Data (Devices or Signals)

The attribute edit box can be entered in one of two ways:

- click on the device/signal to select it, then select the Get Info command in the Options menu, then click on the Attributes... button.
- pop up the device/signal menu by clicking the right mouse button on the device/signal. Select the Attributes command.

The standard attribute edit box will appear.

Select the desired field by clicking on it in the list. The current contents of the field will be displayed in the edit box.

Displaying an Attribute on the Schematic

To display attribute text associated with devices, signals, or pins on the schematic:

1) Click the right mouse button on the device, signal or pin you wish to attach the attribute to. This will display the pop-up menu.

2) Select the Attributes command. This will display the attribute data box.

3) Select the desired field by clicking on its name in the field list. If the desired field is not a Primary field, you will have to click on the "All Fields" button in order to see it in the list.

4) Edit the attribute value as desired.

5) Turn on the Visible switch.

6) Click OK.

The attribute text will now be displayed in a default position near the device or signal. It can be dragged to the desired location using the pointer cursor.

To display attribute text associated with the design a miscellaneous text block needs to be used which contains an attribute variable.

1) Click the pencil tool over an open array of the schematic..

2) Enter the attribute field name preceded by an ’&’, eg: &Designer, using the same spelling and case as the attribute field.

3) Exit text entry mode by clicking outside of the text entry box. The attribute variable should be redisplayed as its value. If the text remains as typed then there was not a like named attribute found.

Rotating Attribute Text

To rotate an attribute text item that is already displayed on the schematic:

1) Click the right mouse button on the text item. This will display the attribute pop-up menu.

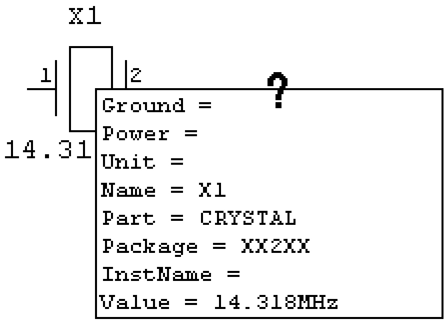
2) Select the Rotate Right or Rotate Left command.

Probing Attributes of Devices, Signals, or Pins

The contents of all primary fields in an object can be viewed with a single click using the Attribute Probe tool.

1) Select the Attribute Probe command in the Edit menu or click on the (?) tool in the tool palette.

2) Click and hold on the device, signal or pin in question. The dot at the base of the question mark is the "hot spot" that determines which item is selected. The popup information box will remain while the mouse is down inside of the popup area.



```
Ground =
Power =
Unit =
Name = X1
Part = CRYSTAL
Package = XX2XX
InstName =
Value = 14.318MHz
```

Setting Attribute Text Style

Attribute text style is set globally for the entire design. There is no way to set text style for an individual attribute item.

> WARNING: Changing the attribute text style affects all visible attributes throughout the design. This may alter text alignment and position to accommodate a different text size.

To set the global text style:

- choose the Design Preferences command in the Drawing menu.
- click on the "Attr Text..." button.
- select the desired font, size and style, then click OK.
- click OK on the Design Preferences box.

Depending on the size of the design, there may be some delay at this point. The program must check all visible attribute items to see if there position and framing is affected by the text change.

Creating a .List Field - Value List

The creation of value list fields will be illustrated by way of an example.

Suppose you wish to create a symbol for a device which will be simulated using an external simulator. This symbol can be associated with one of three models which have the same pinouts, but different timing parameters. The model name is to be included in the netlist.

1) Using the Define Attribute Fields command, create two new fields, called "Model" and "Model.List". For both fields, check the "Devices" box indicating that these will be associated with devices.

2) Using the DevEditor tool, create the desired device symbol, or open an existing one for editing. Using the DevEditor's Part Attributes command, create a field called "Model.List" containing the desired list of values. For example:

```
    MODEL1, MODEL2, MODEL3
```

3) Save the edited symbol to a library. Select the part from the library in the usual way and place it on a circuit diagram.

4 Clicking the right mouse button on the device. You will now see at the bottom of the pop-up menu a new item called "Model". Moving down to this item will display a new sub-menu containing the values entered in the "Model.List" field.

Chapter VII - Hierarchical Design

This chapter provides background and detailed procedures for the hierarchical design features of DesignWorks.

General Concepts

What is Hierarchy?

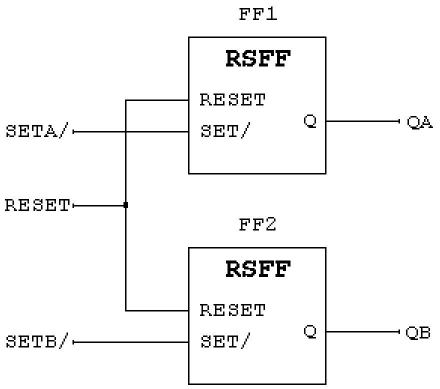
"Hierarchy" refers to the ability to have a "device" symbol in a schematic actually represent an arbitrary circuit block. The "Pins" on the device symbol represent connections to specific input-output points on the internal circuit. For clarity, a device symbol that represents an internal circuit will be called "hierarchical block".

Hierarchical design provides a powerful way of representing complex designs in compact and readable form. A top-level diagram of your system can show only major functional blocks. These blocks can then be opened to show more and more design detail.

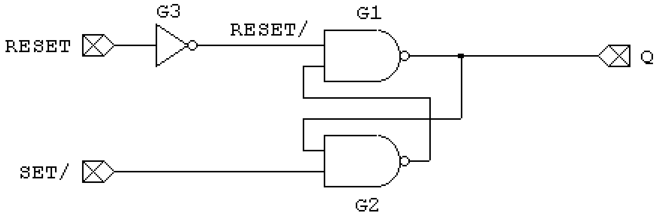
Hierarchical design in effect adds a "third dimension" to a schematic diagram. It also raises some complex issues that should be understood before embarking on a major design. Please review this chapter carefully before making extensive use of the hierarchical features of DesignWorks.

A Simple Hierarchy Example

The following diagram is the master circuit, or top level, of our design example:



Note that it contains two symbols, both representing hierarchical blocks. Both symbols are of the same type, "RSFF", and therefore share the same internal circuit definition. The two blocks are named "FF1" and "FF2". Opening either one of these blocks reveals the following internal circuit definition:



This circuit consists of three device symbols, "G1", "G2", and "G3", representing physical devices, and a number of "port connector" symbols. The port connectors define the interface between the internal circuit and the pins on the symbol representing it.

Note the following characteristics of this simple design:

- The block "RSFF" has been used twice, so there are actually two G1s, one inside FF1 and one inside FF2. We say that there are two *instances* of G1. Similarly for G2 and G3.
- The signals "SET/", "RESET" and "Q" in the internal circuit will actually get absorbed into the attached signals in the parent circuit because they are attached to port connectors. They do not exist independently in the physical circuit.
- Signal "RESET/" does not connect to a port connector, so it represents a separate signal in the internal circuit. Like the devices G1, etc., each signal in RSFF actually represents two physical signals.

These characteristics of a hierarchical design raise the following issues:

- Device and signal names inside hierarchical blocks are not unique if the block has been used more than once. Therefore, a mechanism has to be found for generating a unique name for use in a netlist or bill of materials.
- Signals attached to port connectors get absorbed into the signal attached to the parent symbol’s pin. If this too is attached to a port connector, the process repeats upward until a unique signal is found. The name that will appear in the netlist will be the name of the highest-level signal.
- A single symbol in the definition of an internal circuit can actually represent two or more physical devices. A mechanism must be provided to store separate information for each of these physical devices, e.g. for PCB placement or layout information.

- When transferring data to PCB or simulation systems, it is usually necessary to produce a "flattened" netlist. I.e. a netlist representing the same circuit as if it had been expanded out on a single circuit level.

These issues are addressed in the following sections.

Definition vs. Instance

In DesignWorks, circuit data for hierarchical blocks is separated into two groups:

- Definition data is all information about the structure and connectivity of a circuit, plus the contents of all attributes <u>not</u> marked as "instance" fields. Definition data is associated with the parent symbol and will therefore be identical for all usages of the parent symbol. The "Name" field is a definition attribute and as such a device's name is the same for all its instances. For example, in the simple design above, gate G1 will represent two physical devices yet it only has one name.
- Instance data refers to data kept with each physical instance of a device. In the simple design used in the previous section, gate G1 will have two sets of instance data associated with it, one for each physical device that it represents. Instance data consists of pin numbers, interactive simulation data and any attribute fields marked as "instance" fields. For example, the attribute field "InstName" for the gate named G1 may have the values U1 for its first instance and U2 for its second. This allows the physical name associated with an instance of a device to be kept.

In a hierarchical design, only one set of definition data is kept for each hierarchical block type, whereas separate instance data may be kept for each instance.

Hierarchy Modes

DesignWorks 3.0 implements a true hierarchical design structure. This means that "device" symbols appearing in a circuit can represent another, nested circuit. This sub-circuit can be opened and edited at any time in a separate circuit window. The term "design" is used in this manual to refer to a complete logical system, including the top level circuit and all sub-circuits it contains.

DesignWorks 3.0 has three hierarchy modes to address different design situations.

Flat Hierarchy Mode

"Flat" Mode is the simplest mode and is intended for small designs or those imported from older versions of DesignWorks. Despite the name, it is still possible to create and edit internal circuits, but they are assumed to be for simulation or analysis only. Devices in sub-circuits will not be assigned to physical packages by the Packager tool and will not appear in any netlist output from the Report tool.

Physical Hierarchy Mode

Physical mode allows full use of hierarchical design, while allowing data associated with individual device and signal instances to be stored with the design. It is intended for larger board-level or FPGA designs where interactive simulation is used or PCB layout information is to be kept with the design.

Pure Hierarchy Mode

Pure mode stores only definition data for each device in a design. Attempting to modify instance attribute fields results in a common definition being changed. The "Keep With Instance" setting in attribute fields will be ignored. This allows very large designs to be represented in a small amount of memory, but does not allow interactive simulation or physical layout data to be associated with the design. It is intended for VLSI or other large-scale design applications.

Initial Hierarchy Mode

The following table provides detail on the DesignWorks options enabled for each of the design modes in the New Design... dialog. These options can be individually changed at any time during the design process, although it is usually best to select the desired options before a large amount of schematic entry is completed. This avoids making manual changes to name assignments, etc. later in the process.

Button	**Hier. Mode**	**Dev Auto Vis**	**Packaging**	**Text Tool Sets**	**Name Prefix Field**	**Unit and Part Vis**	**Spice Vis**	**Value Vis**
Flat - PCB	Flat	Off*	On	Name	Name.Prefix	On	Off	On
Flat - SPICE	Flat	On	Off	Name	Name.Spice	Off	On	Off
Flat - General	Flat	Off	Off	Name	Name.Prefix	Off	Off	On

Physical Hierarchy - PCB	Physical	Off	On	InstName	Name.Prefix	On	Off	On
Physical Hierarchy - FPGA	Physical	Off	Off	Name	Name.Prefix	Off	Off	On
Pure Hierarchy - SPICE	Pure	On	Off	Name	Name.Spice	Off	On	Off
Pure Hierarchy - General	Pure	Off	Off	Name	Name.Prefix	Off	Off	On

* Note: In Flat hierarchy mode, the Packaging option overrides the auto-naming feature in any case, so this setting is irrelevant. The Packager will place visible names on devices as they are placed.

In this table:

- Hier. Mode - refers to the Hierarchy Mode setting in the Design Preferences menu command.
- Dev. Auto Vis - refers to the Device Auto Naming - Visible option in the Design Preferences command. If this is On, the auto-assigned Name value will be displayed on the schematic.
- Packaging - refers to the Auto-Packaging Enabled menu command. If this is On, the device name and unit will be assigned automatically as each device is placed.
- Text Tool Sets - refers to the Text Tool Options item in Design Preferences. This determines which attribute field is set when the text tool is used to edit a device name on the schematic.
- Name Prefix Field - refers to the contents of the PrefixField design attribute field, set using the Set Design Attributes command. This determines where DesignWorks looks for a prefix to use in generating a device name.
- Unit, Part, Spice, Value Vis. - refer to the Visible By Default settings of the Unit, Part, Spice and Value attribute fields. These can be set using the Define Attribute Fields command. If this is On, the field value will be displayed on the schematic when the device is placed.

Effect of Changing Hierarchy Mode

The choice of hierarchy mode is not cast in stone once made since you can switch modes at any time. "Flat" and "Physical" modes are very similar and switching between them will normally only require that gate packaging be redone. "Pure" mode is substantially different, however, and should only be used after considering what the loss of physical instance data will mean to the design and to interfacing to target systems.

WARNING: Switching to Pure hierarchy mode from any other mode will cause all instance data to be lost. THIS CANNOT BE UNDONE!!! See the following section.

The following table summarizes the effects of switching between hierarchy modes.

Flat -> Physical	• No data is lost • Attribute field usage for device packaging changes. Packaging will have to be redone
Flat -> Pure	• Any instance data associated with internal circuits will be lost
Physical -> Flat	• No data is lost • Attribute field usage for device packaging changes. Packaging will have to be redone
Physical -> Pure	• Any instance data associated with internal circuits will be lost
Pure -> Flat	• No data is lost
Pure -> Physical	• No data is lost

Netlists from Hierarchical Designs

When producing a netlist for use by an external system, it is necessary to determine whether a "hierarchical" or "flattened" netlist is required.

Hierarchical Netlists

A hierarchical netlist retains the hierarchical structure of the original design. It contains only a single description of each type of hierarchical block used in the design. Normally, the lowest-level blocks are written out first, followed by higher blocks that refer to them, followed finally by the master circuit for the design. By its nature, a hierarchical netlist can contain only definition data.

This format is used by many simulators and FPGA packages, e.g. SPICE.

Generally, any package that will accept a hierarchical netlist can also accept a flattened one. It may be useful to use a flattened netlist in cases where unique instance data was stored with devices or signals in internal circuits.

Flattened Netlists

A flattened netlist has had all information about hierarchical structure removed from it. Every device and signal instance is listed separately with its instance data. Flattened netlists are normally required for PCB formats since name, unit and pin number assignments must be different for each device instance.

See more information on hierarchical netlists in Chapter XI - Report Generation.

Hierarchical Names

When producing a flattened netlist from a hierarchical design, it is usually necessary to generate a unique name for each device and signal in the design. This can be done in one of two ways:

- Manually or automatically assign a unique identifier to each item and store it in an "instance" attribute field. For example, in physical hierarchy mode, the Packager assigns a unique name to each device instance and stores it in the "InstName" field. This option is not available in Pure hierarchy mode.

- Generate a unique identifier consisting of the definition name (i.e. "Name" attribute field) of the device or signal, prefixed by the names of all parent devices. For example, in the simple design example given earlier in this chapter, the six physical devices would be called:

```
FF1/G1
FF1/G2
FF1/G3
FF2/G1
FF2/G2
FF2/G3
```

Assuming the Name field is unique within each circuit level, this is guaranteed to produce a unique identifier for each instance.

The "/" slash character is the standard name separator used in DesignWorks but can be changed by the user if needed. Refer to the Procedures section of this chapter.

Hierarchical Block Primitive Type

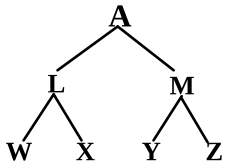
Hierarchical block symbols are simply device symbols which have the primitive type "SUBCCT", or sub-circuit. Device symbols with any other primitive type cannot be used as hierarchical blocks. SUBCCT is the default primitive type when creating symbols with DevEditor, so it is normally not necessary to change this setting.

Printing Hierarchical Designs

A hierarchical design has an extra dimension which must be taken into account in determining the order of page printing.

Print Page Order

When a design is printed, the master circuit of the design is printed first in its entirety. Then, each hierarchical block is printed, followed by all hierarchical blocks it contains. For example, given the following structure of nested hierarchical blocks:



The print order would be: A, L, W, X, M, Y, Z. The order of printing of hierarchical blocks inside a single circuit is determined only by internal storage order and cannot be controlled by the user. E.g. W and X could be reversed above.

Scope of Printing

The Print Range command in the File menu determines how much of a design is printed using the Print command.

See more information in Chapter IX - Schematic Menu Reference.

Page Numbering

When a hierarchical design is printed, the $PRINTPAGENUMBER text variable can be used to apply sequential page numbers to the printed sheets.

NOTE: This text variable only applies during a Print operation. When it is drawn on the screen, it is interpreted the same way as the PAGENUMBER variable, i.e. it gives the page number within the circuit.

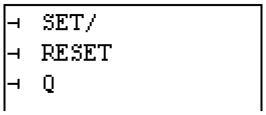
See more information on text variables in Chapter V - Advanced Schematic Editing.

Connecting Signals Across A Hierarchy Level

Signal connections between adjacent levels of a hierarchical design are made using port connector symbols. With the exception of power and ground nets, all connections between levels must pass through a port connector.

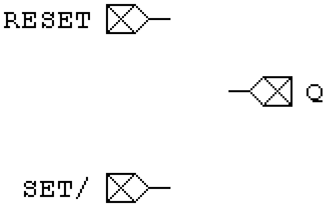
Port/Pin Naming

The relationship between the port connector in the sub-circuit and the pin on the parent device symbol is established by matching the pin name on the parent device with the "Name" attribute of the port connector. E.g. if we were to open the RSFF device, used in the example above, using the DevEditor tool, we would see the following pins listed:



```
⊣ SET/
⊣ RESET
⊣ Q
```

For a complete port interface, a port connector must exist in the internal circuit named to match each one of these pins. In this case, the following port connectors would be required (ignoring all other internal circuitry):



The port interface is rechecked whenever any change is made. Thus, as soon as a port connector is added or removed, or its "Name" attribute is changed, the port interface has been updated to reflect the new logical connections. However, to avoid excessive warning messages, error checking is performed only when an internal circuit is opened or closed. A warning box will be displayed if any error is found. This checking cannot be disabled.

NOTE: When naming ports, connections are only made by examining the "Name" field. Make sure you assign the port name to the "Name" attribute field and not to the "InstName" attribute field.

NOTE: The name of the port connector's pin and the name of the signal attached to the port connector are not significant in making the port association. Only the contents of the port connector's Name attribute field is used.

NOTE: The rules are different for bus ports. See below.

Port Pin Type

If pin type (i.e. input/output/bidirectional, etc.) is important for your design, then the following rules should be noted when creating and using Port Connectors:

- The pin on the Port Connector symbol is normally of the opposite type to the corresponding pin on the parent device symbol. E.g. A signal coming in to the hierarchical block is actually an output from the port connector pin.
- DesignWorks does not check correctness of pin types on port connectors.
- When an internal circuit is auto-generated, port connectors of a type opposite the parent pin are placed automatically.

Bus Ports

Connections can be made between busses across hierarchy levels using Bus Port Connectors. Bus pins on a parent device symbol must be matched with a Bus Port Connector having identical internal pins. For this reason, Bus Port Connectors must always be custom made, either using the auto-generate function described below, or using the DevEditor tool.

Bus Pin Name Matching

Note the following rules for name matching in bus ports:

- As with other Port Connectors, a Bus Port Connector must be given a Name attribute exactly matching the pin name of the bus pin on the parent device.

- The internal pins in the parent bus pin must exactly match the internal pins on the Bus Port Connectors bus pin.

- The pin name of the bus pin itself on the Bus Port Connector is not significant.

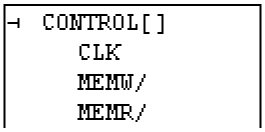
- As with normal ports, the names of the signals attached to the Bus Port Connector's pin are not significant.

Bus Pin Example

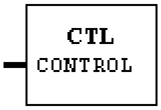
For example, the following simple device has a bus pin called CONTROL containing internal pins CLK, MEMW/, and MEMR/.

Pin List (in DevEditor)

Hierarchical Block Symbol



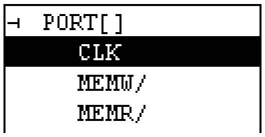
```
⊣ CONTROL[]
    CLK
    MEMW/
    MEMR/
```



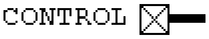
The corresponding Bus Port Connector to be used inside this device would look as follows:

Pin List (in DevEditor)

Port Connector Symbol



```
⊣ PORT[]
    CLK
    MEMW/
    MEMR/
```



The comments above under "Port Pin Type" apply to each internal pin in a bus pin. Remember that the name of the bus pin in the port connector is not significant.

Power and Ground Connections

Power and ground symbols (i.e. Signal Connector devices) do not make an immediate logical connection across hierarchy levels. For this reason, signal connectors should not be used to make active signal connections for interactive simulation purposes.

However, it is possible to merge power and ground nets across levels in netlist output using the Report Generator's SIGSOURCE function.

See Chapter XI - Report Generation for more information.

Auto-Generated Internal Circuit Template

If an empty hierarchical block symbol is opened (either by double-clicking on it or using the Push Into command), Design Works will create default "template" for the internal circuit using the following rules:

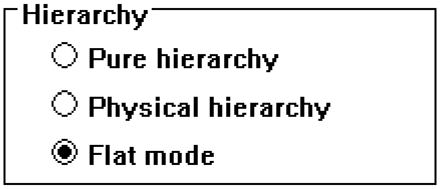
- A default port connector symbol is placed for each pin on the parent block, placed according to their position on the symbol.

- Each pin on each port connector has its type set according to the rules outlines earlier in this chapter.

Hierarchy Procedures

Changing the Hierarchy Mode

To change the hierarchy mode for a design, select the Design Preferences command in the Drawing menu. The following controls allow selection of hierarchy mode:



Hierarchy

○ Pure hierarchy

○ Physical hierarchy

● Flat mode

Click on the desired button to select a new hierarchy mode, then click on the OK button. A warning box will ask you to confirm the mode change.

WARNING: See the reference section earlier in the chapter for warnings about potential loss of data when switching hierarchy mode.

Creating a Hierarchical Block - Top Down

To create a hierarchical block from the top down, the following steps must be followed:

1) Create the block symbol using DevEditor.
2) Place the symbol on the schematic.
3) Create the internal circuit by pushing into the new symbol.

These steps are described in the following sections.

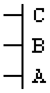
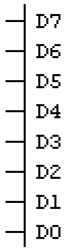
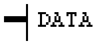
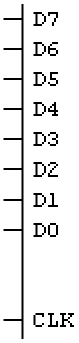
<u>Creating a Block Symbol</u>

Symbols for hierarchical blocks are created using the DevEditor tool. DevEditor provides numerous tools and functions for symbol editing. We will provide only a quick summary of one technique.

1) Invoke the DevEditor by selecting its name in the Tools menu.

2) Select the Auto Create Symbol command in the DevEditor menu.

3) In the "Left Pin Names" text box enter a list of pins to appear on the left-hand side of the symbol. Following are some examples of pin lists:

Pin List	**Pin Description**	**Symbol**
A B C	Three signal pins A, B and C	
D0..7	Eight signal pins D0, D1, D2, ... D7	
DATA[D0..7]	A bus pin called DATA containing 8 internal pins D0, D1, ... D7	
CLK,,,D0..7	Nine pins in all with extra space between CLK and the others	

4) In the "Right Pin Names" text box enter a list of pins to appear on the right-hand side of the symbol, using the same format as above.

5) Click the "Generate" button.

6) Make any desired graphical edits to the auto-generated symbol.

7) If you do not already have a suitable temporary library file open, create one now using the New Lib command (Using the Parts palette pop-up menu).

8) Save the new part into the temporary library.

See Chapter X - Device Symbol Editing for more symbol editing techniques.

Placing the Block Symbol

Hierarchical block symbols are placed just like any other device symbol:

1) In the Parts palette, use the library selection menu to select the library that you saved the block symbol in.

2) Double-click on the name of the new symbol.

3) Place the symbol where desired on the schematic.

Creating the Internal Circuit

In this section we will take advantage of the fact that DesignWorks creates a default internal circuit template when you open an empty block symbol. It is also possible to create your own internal circuit from scratch. This is discussed below in the section on "bottom-up" block creation.

1) Click the right mouse button on the hierarchical block symbol.

2) Select the Push Into command.

3) A box will appear asking you to confirm that you wish to create a new internal circuit. Click the OK button.

A new circuit window will now open with the default port connectors laid out according to their position on the parent symbol.

Creating a Hierarchical Block - Bottom Up

Any open design can be made into the sub-circuit of a hierarchical block symbol. To create the internal circuit first and work upwards, follow these steps:

> 1) Create or open the design file that is to become the internal circuit. Make sure that its hierarchy mode is the same as the design its being linked to.
>
> 2) Create the hierarchical block symbol.
>
> 3) Link the sub-circuit to the symbol using the Attach Internal command.

These three steps will be covered in more detail in the following sections.

Creating a Sub-Circuit

Creating a circuit that is to be attached to a parent symbol is exactly the same as creating an independent design, except for the necessity of defining the port interface. Note the following issues when creating the sub-circuit:

- Each signal that is to be an input/output point for the sub-circuit (i.e. a pin on the parent symbol) must have an attached port connector.

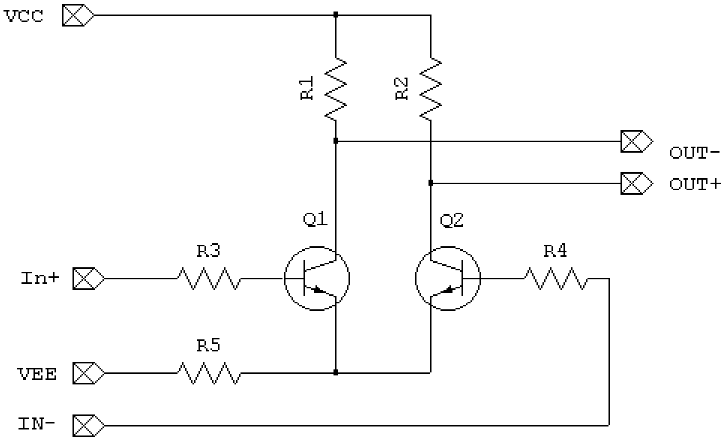
- The name of the port connector device must exactly match the name of the associated pin on the parent symbol.

- For each port, choose a port connector type (input, output or bidirectional) appropriate to the signal flow in your circuit. This allows for more efficient simulation, improved error checking and design clarity.

- Bus ports require special care in creation and naming of port connector symbols. Review the rules given earlier in this chapter carefully.

See the section earlier in this chapter on Port Interface for complete details.

The following sample shows how port connectors are attached to the connection points in a simple circuit:



Creating a Symbol for an Existing Sub-Circuit

To create a symbol for a sub-circuit that has already been created and is saved as a file:

1) Select the DevEditor tool in the Tools menu.

2) Select the Subcircuit & Part Type command in the DevEdit menu.

3) Click on the Imported Port Definition Only button. This will display a list of saved designs.

NOTE: Choosing the External Subcircuit option instead will cause a reference to the circuit file to be saved in the library with the symbol.

4) Select the desired external circuit in the list.

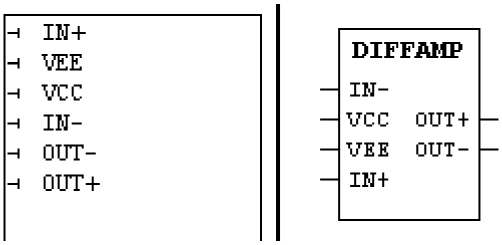
DevEditor now examines the selected circuit and creates a list of pins matching the port connectors in the circuit.

5) Select the Auto Create Symbol command in the DevEdit menu.

6) Click on the Extract Pin List button.

This will place input and output pins in default positions in the Left and Right Pin Names boxes. You may move names between these boxes as desired to determine placement of pins. Be careful not to add or delete any names. You can also change the part name that will appear in the symbol, if desired.

7) Click on the Generate button. A default symbol will now appear in the DevEditor window.



This symbol can be edited manually using the drawing tools at the left.

8) Create a temporary library to save the symbol in. This is done using the New Lib command in the Libraries sub-menu of the File menu, or in the Parts palette pop-up menu.

9) Select the Save Part As command in the File menu. Select a suitable part name and double-click on the new library that was created in the last step.

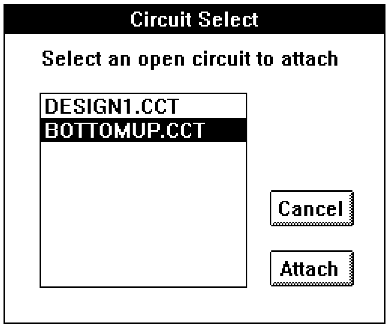
10) Close the DevEditor window.

Linking a Symbol to an Existing Sub-Circuit

1) Place the new symbol in the master circuit for the design, by double-clicking on it in the Parts palette and placing it in a suitable position in the schematic.

2) Click on the new symbol to select it.

3) Choose the Attach Internal command in the Options menu. A box will appear listing all open designs except the current one (a sub-circuit cannot be attached inside itself!):



Circuit Select

Select an open circuit to attach

DESIGN1.CCT
BOTTOMUP.CCT

Select the desired internal circuit (presumably the same one that was selected while in the DevEditor), and click on the Attach button.

The internal circuit will now be linked to the selected symbol and its window brought to the front.

NOTE: The Attach Internal command in effect changes the definition for the selected device by adding an internal circuit. If there were other devices of the same type on the schematic, you will be asked if you want to update them when the internal circuit is closed.

Changing the Hierarchical Name Separator

The separator character used to generate hierarchical names is stored in the design attribute field "HierNameSep".

To change it:

1) Select the Set Design Attributes command in the Options menu.
2) Select the HierNameSep field in the field list.
3) Enter a new value in the data box.
4) Click OK.

Chapter VIII - Before Starting a Major Design

This chapter provides information that should be considered before starting any major design. Correct choices made in advance concerning usage of hierarchy mode, attribute format, sheet layout etc. will save time and frustration later. Issues such as hardcopy format, drawing standards, interfacing with other systems, and reports required are addressed.

This chapter is not intended as a specific reference for any DesignWorks features. It focuses on your desired end-product and how best to make use of this package to achieve that goal. Its primary aim is to raise issues that you should consider and alternatives that you may wish to test before proceeding.

In addition, this chapter makes only general reference to third-party products and specific issues on interfacing with them. Capilano Computing has a number of Technical Notes available on issues concerning specific third-party packages. Some of these notes were provided with this manual and others can be obtained by calling our Technical Support department at the number listed in the front of this manual.

Hierarchical Design

Hierarchy Mode

DesignWorks offers three hierarchy modes to suit a variety of applications. Following are a number of points to consider before settling on which to use.

Hierarchical design is a powerful technique, but does add complexity to the design process. If you are one of the following:

- a new user of schematic capture tools
- working with small-to-medium designs (say, less than 200 ICs)
- interfacing to a third-party Printed Circuit Board layout package

...then we recommend sticking with "Flat" mode and <u>not</u> using hierarchical blocks in your designs. Once you have gained some experience with creating schematics in DesignWorks and moving design data between systems, you can review the hierarchy issues discussed in this manual.

Pure Hierarchy Mode Restrictions

Pure hierarchy mode is intended for large designs that:

- lend themselves to a very structured hierarchical definition.
- do not require device instance data.
- do not require flat netlist output.

Unless you are sure your design meets these requirements, do not use Pure mode. Pure mode is not recommended for printed circuit board-level designs.

Hardcopy Output Format

The size and format of the schematic sheets you use will likely be determined by your organization's drawing standards and by what form of hardcopy device you have available. It is best to do some test sheets in your chosen format before proceeding with a large design. Reformatting a design to a different sheet size is an unpleasant task.

Printing

Most laser or dot-matrix printers have a number of options for print area size and scaling. For this reason, it is best to either:

- use the "Use Page Setup" option and default borders in the Custom Sheet Info command. This way, the schematic drawing area will automatically adjust to changes in printer page setup.

OR

- choose a format and scale factor that looks good on your printer and create a fixed sheet template for it.

Interfacing with PCB Layout Systems

If you plan to use DesignWorks in conjunction with any other package for Printed Circuit Board layout, it is best to review any restrictions that may be imposed on data imported to that system. A number of possible considerations are mentioned below.

IMPORTANT: We strongly recommend that you create a small test case and go through the procedure of creating the output files and passing them to the desired target system. This will help you correct interface problems before they have a major impact on your design.

Naming restrictions

DesignWorks allows devices and signals to have any name up to 16 characters. Except for carriage returns, anything you can type on the keyboard can be part of a name. Many systems place some restrictions on names, for example:

- length restrictions
- usage of special characters
- case sensitivity

If you intend to use DesignWorks’ hierarchical design features, you will need to determine whether hierarchical names (e.g. "BLOCK1/INBUF/REG1") of arbitrary length are acceptable. If not, it will be necessary to ensure that each device and signal has a unique identifier stored in an instance attribute.

Part Names and Package Codes

Most netlist formats for PCB systems contain either or both of the following for each device in the design:

- a part name, e.g. "74ALS244"
- a package code, e.g. "DIP20"

There is no clear industry standard for the format of either of these codes. It is therefore important to verify that the type names and package codes used in the DesignWorks libraries will be acceptable to the target system. If not, it will be necessary to either update the library entries for the parts you will be using or make appropriate attribute entries for each device as you use it on the diagram.

Interfacing with SPICE

DesignWorks 3.0 has a number of features specifically intended to assist in creating designs for SPICE-based analog simulators.

> *See also the SPICE report examples in Chapter XI - Report Generation.*

SPICE Data Field

An attribute field called "Spice" is predefined for devices and the design itself. If the standard Spice Report forms are used, any text data entered in this field is included in the output file, as follows:

• for devices, the data is appended to the corresponding device line in the file. E.g. entering the value "SIN(0.0 0.01V)" in the Spice field for a voltage source would create the following line in the SPICE output file:

```
V1 0 VPLUS SIN(0.0 0.01V)
```

• for the design, the contents of the field is inserted verbatim at the front of the file. This can be used to enter parameter lines, model definitions, etc. Multiple lines can be entered if desired.

SPICE Name Prefix

Library parts in the SPICE Devices standard library have a field called "Name.Spice" defined. This field contains the standard SPICE name prefix letter for that type. To use this prefix, use the Design Attributes command to set the PrefixField field to "Name.Spice".

Hierarchical Designs for SPICE

Hierarchical designs created in Pure or Physical hierarchy mode will be written to the SPICE file using the SUBCKT statement. See the SPICE sample designs provided with the package.

Pin Ordering for SPICE

In the SPICE netlist format, the functions of device pins are determined by the order of appearance in the file. Therefore, the ordering of pins on a device symbol is extremely important for this usage. Pins are normally placed in the order they appear when the symbol is opened using the DevEditor, although this can be modified using the Report tool's DEVPINSEQUENCEfacility and the PinSequence attribute field, described below.

Discrete Component Symbols

Discrete devices in the standard SPICE Devices library provided with DesignWorks have their pin order preset for SPICE usage. If you are using any other discrete component symbols, or creating your own, check the pin order carefully.

Digital Component Symbols

Unfortunately, there is no established standard pin ordering for digital simulation extensions to SPICE. The standard DesignWorks digital libraries <u>are not set up</u> to work with SPICE-based digital simulators, although most gate-level parts are created using the natural order of inputs first, then outputs.

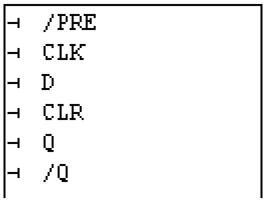
In order to use digital components with a SPICE format, it will be necessary to <u>either</u>:

- enter the pins to be listed in the net-list file in the device's PinSequence attribute field, OR
- create a special symbol with the pins in the correct order, OR
- obtain libraries of symbols specially created for use with your simulator.

Using the PinSequence Field

The standard SPICE report forms provided with DesignWorks make use of the PinSequence attribute field to specify pin ordering for digital parts. To specify pin ordering, it is first necessary to know the names of all the pins on the device in question. This can be obtained using the Get Info command, the Pin Info command or by opening the part type using the DevEditor.

For example, the 74ALS74 part has the following pin list when viewed in the DevEditor:



```
⊣ /PRE
⊣ CLK
⊣ D
⊣ CLR
⊣ Q
⊣ /Q
```

To specify an alternate order for output, you could enter the following string into the PinSequence attribute field for the device:

```
/CLR,D,CLK,/PRE,Q,/Q
```

Only the items entered in this list will appear, i.e. not all pins need be listed and some can be listed more than once, if desired. The pin names must match exactly the list as seen in the DevEditor, including case. Unrecognized names are ignored by the Report tool.

NOTE: Entries in the PinSequence field are effective only if specifically enabled by the selected Report form file. The SPICE form files provided with DesignWorks include this entry. If you are creating your own format, refer to the description of the $DEVPINSEQUENCE format command.

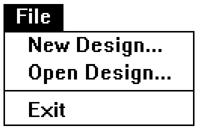
Chapter IX - Schematic Menu Reference

This chapter provides detailed information on the use of individual menu commands in DesignWorks. In order to give you rapid access to commands, the program has two different sets of menus:

- • Standard Menus - these are the normal File, Edit and other menus that appear in the menu bar.
- • Pop-up Menus - these menus will appear anywhere on the schematic diagram when the right mouse is clicked. Most of the commands appearing in these menus also appear in the standard menus, so they can be considered a short-cut. The type of menu that appears depends on where the mouse is clicked. Different menus appear for Circuit, Signal, Device, Pin and Attribute functions.

The commands in the DevEditor menu are covered in Chapter X - Device Symbol Editing.

DesignWorks File Menu



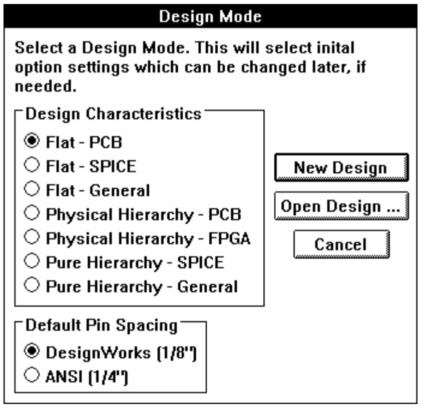
New Design...

This command will create a new, empty design window. This can be used to create an entirely new design, to create a sub-circuit that will later be associated with a device, or as a temporary area to edit a circuit scrap.

There is no fixed limit on the number of designs that can be open at once, although the complete contents of all open designs must fit into memory.

The design will use the current printer Page Setup for its sheet size. More information on these settings is provided under the Custom Sheet Info and Design Preferences commands. New Design... does not create a disk file and has no effect on any files on your disk.

Each time a new design is created a design mode dialog box is displayed. This dialog does not actually implement any new options, it just groups them by design type and encourages you to select the appropriate mode in advance. All options can be set individually later in the design if desired, although this dialog can only be displayed when creating a new design.



Select a Design Mode. This will select inital option settings which can be changed later, if needed.

Design Characteristics

- ● Flat - PCB
- ○ Flat - SPICE
- ○ Flat - General
- ○ Physical Hierarchy - PCB
- ○ Physical Hierarchy - FPGA
- ○ Pure Hierarchy - SPICE
- ○ Pure Hierarchy - General

Default Pin Spacing

- ● DesignWorks (1/8")
- ○ ANSI (1/4")

Users are encouraged to use one of the Flat modes until they are familiar with the ramifications of the other design modes.

Note: If you always use the same options and want to override this dialog, the DW.ini file options NEWDESIGN and NOUNTITLED allow you to control the creation of untitled designs. See more information on these options in the Appendix.

Design Mode

The upper group of buttons allows you to choose the initial option settings for your design based on the type of application. The following table summarizes the usage of these modes.

Mode	**Description**
Flat - PCB	This mode is intended for creating small to medium designs intended to be implemented as a Printed Circuit Board and where hierarchical design is not required. This is the default mode and the best one to choose if you’re not sure.
Flat - SPICE	This mode is intended for creating small to medium designs for input to a SPICE-based analog simulator. SPICE-format names will be assigned automatically.
Flat - General	This mode enables a minimum of options. It is used for block diagramming, simulation or analysis where device packaging is not needed.
Physical Hierarchy - PCB	This mode is intended for creating larger PCB designs with hierarchy.
Physical Hierarchy - FPGA	This mode is for hierarchical designs which require physical device data (i.e. for simulation or physical layout information), but do not need automatic gate assignment or name display.
Pure Hierarchy - SPICE	This mode is intended for creating hierarchical circuits (i.e. with SUBCKT definitions) for use with a SPICE-based simulator.
Pure Hierarchy - General	This mode is intended for general hierarchical design. NOTE: Pure hierarchy mode cannot be used with the DesignWorks Digital Simulator and is not recommended for PCB designs.

NOTE: It is strongly recommended that you read Chapter VII - Hierarchical Design in the DesignWorks manual before using any of the non-Flat modes.

Default Pin Spacing

The lower group of buttons, in the New Design... dialog, allows you to select from one of two commonly-used symbol styles supported by DesignWorks.

NOTE: This setting has no effect on the contents or availability of symbol libraries. It is up to you to choose the appropriate setting for the symbols you intend to use. The default DesignWorks installation installs only the DesignWorks-format symbols (1/8" pin spacing).

The following table summarizes the effect of these buttons.

DesignWorks (1/8")	This mode sets the default text sizes to 9-point for attribute and pin number text and the default pin spacing (i.e. for breakout symbols and DevEditor operations) to 2 grid units (approx. 1/8"). These settings will give best results with the DesignWorks-format symbol libraries provided with the package.
ANSI (1/4")	This mode sets the default text sizes to 12-point for attribute and pin number text and the default pin spacing to 4 grid units.* These settings will give correct text and line space proportions for ANSI-standard drawings when used with the ANSI or IEEE-format symbols provided. NOTE: The ANSI and IEEE symbols are not installed by the Easy Install option. You must select the Customize option to load them onto your disk.

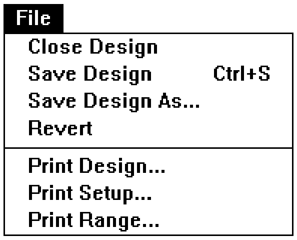
* NOTE: The ANSI setting actually uses the setting specified in the PINSPACE keyword in the DesignWorks Ini file. If no PINSPACE item appears in the Ini file, the default value of 4 is used.

Open Design...

This command allows a design to be opened from a disk file.

When you open a file, the circuit data is read into memory in its entirety and no more access to the disk file is required. DesignWorks will let you open multiple copies of the same file and makes no attempt to restrict you from writing any of them back to the same file. If you do this, it is up to you to keep track of which windows have been updated and what file you want to save them into.

Schematic File Menu



Close Design...

When **Close Design** is selected from the file menu on a schematic circuit window, this indicates to close all of the windows associated with the same circuit file. If the design has been modified then the user will be given an opporunity to save the design.

Save Design and Save Design As...

Save Design and **Save Design As** save the current design in a circuit file. **Save** saves the circuit back into the file that was most-recently opened. It will be disabled if no file has been opened. If you select **Save Design As**, a dialog box will be displayed requesting the name the new file. The default name will be the current title of the circuit window, i.e. the name of the most recently opened or saved file.

Revert

This command rereads the current design from the disk file it was last saved to or read from. If any changes have been made since the last Save, you will be prompted to confirm the choice before they are discarded.

Print Design...

Print Design allows you to print all or part of your circuit diagram. If the diagram will not fit on a single page, it will be broken into as many parts as are needed, based upon paper size specified in Page Setup. You can preview the page breaks by using the Show Page Breaks option in the Design Preferences command. For purposes of specifying a range to print, pages are numbered from top to bottom, then left to right, starting on circuit page 1. In multi-page circuits the printed pages are numbered sequentially from the top left corner of circuit page 1 to the bottom right corner of the last circuit page. Page numbers do not appear in the printed output unless explicitly added with text variables.

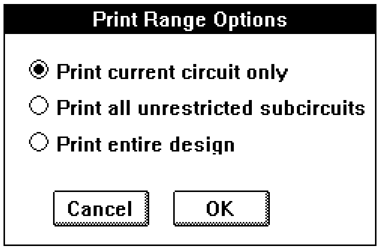
In hierarchical designs, pages are numbered separately for each circuit level. The Print Range command is used to determine printing extent.

Page Setup

The page setup dialog will be presented allowing you to choose the size and orientation of printer paper you wish to use. Once chosen, this information will be stored with your design file and affect the page outlines shown in the command and the Show Printed Page Breaks option in the Design Preferences command

Print Range...

This command is used to set the scope of printing in a hierarchical design.



Print Range Options

● Print current circuit only

○ Print all unrestricted subcircuits

○ Print entire design

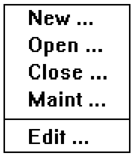
Three options are available:

1) Print current circuit only: With this setting, the Print Design command will print only pages belonging to the circuit level in the current window. With this setting, the page numbers shown by the "Show Printed Page Breaks" option can be used to select a subset of pages to print.

2) Print all unrestricted circuits: With this setting, the design’s master circuit and all unlocked internal circuits will be printed. (The "locked" setting is controlled by a check box in the Get Info command for the parent device.) With this

setting, the page numbers shown by the "Show Printed Page Breaks" option will not be relevant since they are numbered separately for each internal circuit.

Libraries Pop-up Menu



This menu can be reached only by clicking the right mouse button anywhere in the Parts palette.

New...

The New Lib command allows you to create a new, empty symbol library file on your disk. It will be opened automatically and appear in the Parts palette.

Open...

The Open Lib command allows you to select an existing symbol library file to be opened. The name of the library will appear in the library selection list at the top of the Parts palette. A small amount of memory is occupied by each open library file.

Note: Libraries can be opened automatically when the program starts by placing them in the default library folder or by using the LIBRARY and LIBRARYFOLDER setup file keywords.

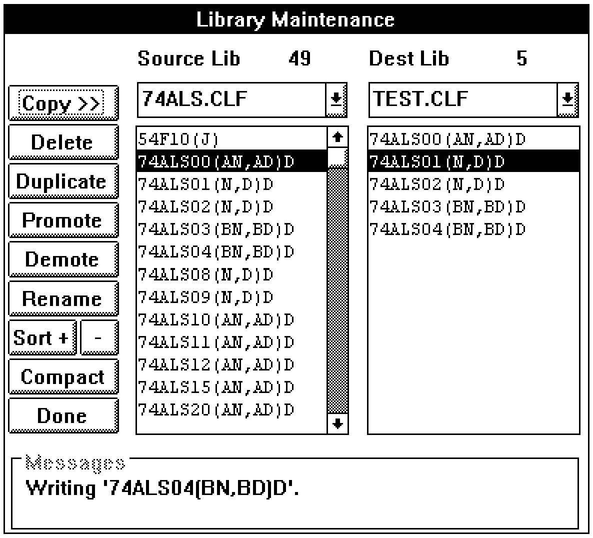
See Chapter II - Getting Started and the Appendix for more information.

Close...

Close Lib allows you to close an open library and remove it from the Parts palette. Any information required for parts used in any open designs will be automatically retained in memory. When you select the Close Lib command, a box will appear listing the open libraries. Simply pick one by clicking on it and then press the Close Lib button, or simply double-click on the name of the library.

Lib Maint...

This command invokes a variety of library maintenance functions. The following box will be displayed:



Source Lib 49

74ALS.CLF

54F10(J)
74ALS00(AN,AD)D
74ALS01(N,D)D
74ALS02(N,D)D
74ALS03(BN,BD)D
74ALS04(BN,BD)D
74ALS08(N,D)D
74ALS09(N,D)D
74ALS10(AN,AD)D
74ALS11(AN,AD)D
74ALS12(AN,AD)D
74ALS15(AN,AD)D
74ALS20(AN,AD)D

Dest Lib 5

TEST.CLF

74ALS00(AN,AD)D
74ALS01(N,D)D
74ALS02(N,D)D
74ALS03(BN,BD)D
74ALS04(BN,BD)D

Writing '74ALS04(BN,BD)D'.

The following table summarizes the options available.

Source Lib	The Source Lib is the primary library operated on by all the command buttons. The Source Lib can be any one of the currently open libraries, selected using the pop-up list box at the head of the list. Any single item in the Source Lib can be selected by clicking on it. A contiguous set of items can be selected by holding the Shift key to add to the selection.
Dest Lib	The Dest Lib is used only as the destination of the Copy and Compact command. No items can be selected in this list.
Copy	This button causes the selected parts in the Source Lib list to be copied to the Dest Lib.
Delete	This button causes the selected parts in the Source Lib to be deleted. THIS CANNOT BE UNDONE!!!

Duplicate	This causes the selected parts in the Source Lib to be duplicated, i.e. a copy of the selected items is made in the Source Lib. The Dest Lib is not affected.
Promote/Demote	The Promote and Demote buttons cause the selected items in the Source Lib to be moved up or down the list, respectively.
Rename	Rename displays a box allowing a new name to be entered for the selected part.
Sort	This button sorts the entire list alphabetically. The adjacent button sorts in the opposite direction.
Compact	This button causes the Source Lib to be compacted to the destination lib (which must be empty), i.e. any free space due to deletions is removed. See more information below.
Done	The Done button closes the Lib Maint dialog box.

Library Compaction

When parts are deleted from a library the free space in the file is not automatically recovered. In most cases this is not a significant overhead. However, if a large percentage of the parts in a library have been deleted then you may wish to compact the file. To do this:

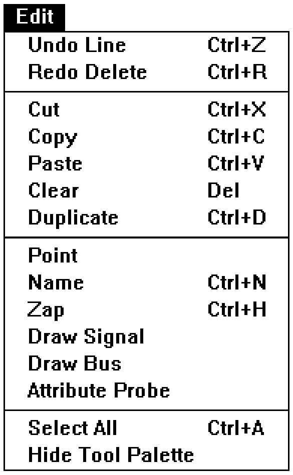
1) Create a new, empty library which will become the target for the Compact operation.
2) Select the Lib Maint command.
3) Select the library to be compacted as the Source Lib.
4) Select the new, empty library as the Dest Lib.
5) Click on the Compact button.

IMPORTANT: Verify that the new destination library is correct before discarding the old copy.

Edit ...

This command starts up the DevEditor tool and opens the part currently selected in the Parts palette. This item will be disabled if nothing is selected in the Parts palette.

Edit Menu



Undo

This command undoes the last editing operation that was performed. The text of this menu item will change based on the type of operation. Generally, only schematic editing operations can be undone. Major structural changes, such as Define Attribute Fields, or any menu commands involving a dialog box are usually not undoable.

Undo never changes the contents of the Clipboard. E.g. after a Cut operation, Undo will restore the schematic, but leave the Cut objects on the Clipboard.

Redo

This command redoes the last Undo command. It will only be enabled immediately after an Undo operation. Any other editing operation will disable this item.

Using the Clipboard

The standard clipboard commands Cut, Copy and Paste can be used to move or copy circuit fragments, graphical and text information within a single circuit window, between multiple windows, and between different MEDA modules, E.g. Drawing and DevEditor.

Using Clipboard Data From Other Programs

The current version of DesignWorks for Windows does not support importing or exporting data via the clipboard from other applications.

Using Clipboard Data From DesignWorks

When a Cut or Copy is done, two types of data are placed on the clipboard:

- A picture ("PICT" format) of the selected items, which can be pasted into other MEDA modules.
- The DesignWorks circuit info for the selected items. This data is in a format that only DesignWorks can understand and is discarded when you Quit.

WARNING: Circuit structural information on the clipboard is discarded when you quit the program. Only picture and text data is retained. You cannot Copy and Paste circuit data between DesignWorks sessions.

Cut and Copy work on the currently selected group of circuit objects and will be disabled if no objects are selected. When items are copied onto the clipboard, their names are copied with them, which may result in duplicate names. If duplicate signal names are pasted back into the circuit page they were copied from, then logical connections will be made between the like-named segments.

Cut

Cut removes the currently selected objects from the circuit and transfers them to the clipboard. It is equivalent to doing a Copy and then a Clear. Cut will be disabled if no objects are selected.

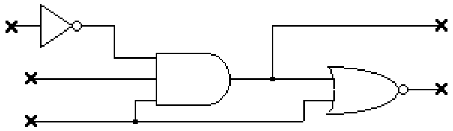
Copy

Copies the currently selected objects onto the clipboard without removing them. This can be used to duplicate a circuit group, copy it from one file to another, or to copy a picture of the circuit group to a drawing program. See the notes on clipboard data above. Copy will be disabled if no objects are currently selected.

Paste

Paste replaces the cursor with a flickering image of the contents of the Clipboard in the circuit window. As noted above, this data may be a circuit group copied from within DesignWorks, or it may be text or picture information created by another module. The image of the clipboard data can be dragged around and positioned as desired. The item will be made a permanent part of your diagram when the mouse button is pressed.

DesignWorks checks for signal connections only at "loose ends" in the signal lines being pasted, i.e. ends of line segments which do not touch devices or other line segments. E.g. if the following circuit scrap was pasted, the points marked X would be checked for connection to the existing circuit.



NOTE: Connection "hit testing" can be disabled by holding the Control key on the keyboard depressed while clicking the mouse button (this also applies to single device placing). In this case, the circuit scrap is placed, but no connections will be made to adjacent items. This allows the group to be selected again (by Control-double-clicking on any device in the group) and moved without interactions with other objects in the circuit.

Paste will be disabled if there is no information on the clipboard of a recognized type.

Rotation on Paste and Duplicate

Any group of objects being Pasted or Duplicated can be rotated using the same controls as when placing a device:

- the Orientation pop-up menu in the tool palette.
- the arrow keys on the keyboard.

Note that these controls are only effective while actually moving the flickering image of the object being pasted. Each Paste or Duplicate always starts in the same orientation as the source.

NOTE: The Orientation command cannot be used for Paste or Duplicate since selecting this menu command will abort the paste operation.

Clear

Removes the currently selected objects from the circuit. Typing the backspace or delete key on the keyboard is equivalent to Clear. This will be disabled if no devices or signals are selected.

Duplicate

Makes a copy of the selected circuit group which can be dragged and positioned as desired. This is equivalent to selecting Copy and then Paste, except that the selected circuit scrap is not placed on the clipboard for future use. See the notes under Paste, above, on how connections are made when a group is placed in the circuit. Note that the duplicated objects can be rotated using the orientation pop-up menu in the tool palette or the arrow keys on the keyboard.

Point

This selects the normal operating mode for DesignWorks, indicated by the arrow cursor. Selecting this command is equivalent to clicking on the ↖ icon in the tool palette. The following functions are accessible in Point mode:

- by clicking on an object you can select the object for operations using the Edit menu commands. To select an I/O device, or to select multiple objects, hold the SHIFT key on the keyboard while you click. To select a Background object, hold the Shift and Control keys while clicking on it.
- by clicking and dragging near the end of a device pin or signal line you can extend that line in any direction.
- by clicking and dragging a signal line anywhere but near the end then you can change its perpendicular position.
- by clicking and dragging any other object you can reposition the object.

All of the above functions are described in more detail in Chapter IV - Schematic Editing.

<u>Shortcuts to Point</u>

Since it is frequently desired to return to Point mode, a shortcut has been provided:

- pressing any key on the keyboard.

Text

The Text command changes the current cursor to Text mode and is equivalent to clicking on the **T** icon on the tool palette. In this mode the following functions are available:

- a name can be associated with a device by clicking and holding on the device in question, then moving the cursor to the desired position for the text then releasing the button, then typing the desired name (up to 16 characters) followed by Enter.

  A name entry can be located in a default position for the device by clicking and releasing the mouse button on the device without moving the mouse. The name can then be entered as described above.

  The text tool sets either the Name attribute or the InstName attribute as directed by the settings in the Design Preferences dialog.

- a name can be associated with a signal by clicking and holding anywhere along a signal line, then proceeding as for devices above. Signal names can appear at multiple locations along the length of the signal line, up to a maximum of 100 positions. Additional name positions are added by simply repeating the naming procedure as many times as required. If the name at any position is altered then all positions are updated.

- any existing attribute item displayed on the schematic (including Name or InstName) can be editing by clicking in it.

- a pin number can be placed on a device pin by clicking on the pin within 5 screen dots of the device. A blinking insertion point will appear, and you will be able to type up to 4 characters. Press the Enter key to terminate the pin number.

- random text (i.e. not associated with a specific device or signal) can be placed anywhere on the diagram by clicking on the diagram other than on a device or signal. This text can contain carriage returns or any other characters.

- any of the above items can be edited by clicking anywhere in the existing object. The blinking insertion point will appear in the text at the position of the click.

To abort a text editing operation without making the changes press the ESC key.

All of the above functions are described in more detail in Chapter IV - Schematic Editing.

Zap

The Zap command changes the current cursor to Zap mode and is equivalent to clicking on the ↯ icon in the tool palette. When the tip of this cursor is clicked on any object in a circuit then that object is hidden or removed. The Zap cursor is more selective than using the Clear command on selected objects:

- Signal or bus lines: only the line segment under the cursor is removed.

- Pin numbers: Removes the pin number.

- Attribute items: Removes the visible attribute text from the diagram, but leaves the value associated with the object.

See more information on this command and other editing features in Chapter IV - Schematic Editing.

Draw Sig

The Draw Sig command is equivalent to clicking on the + icon in the tool palette and places the program in signal drawing mode. In this mode you can draw or extend signal lines as follows:

- clicking anywhere along an existing signal line extends the signal, starting at that point.
- a new signal can be created by clicking anywhere on the diagram.

When you click again, the lines on the screen become permanent and a new set of lines are drawn starting at that point. A number of line routing options are selected by pressing the Shift, and Control keys while drawing. To terminate signal drawing mode, double-click or press the space-bar on the keyboard.

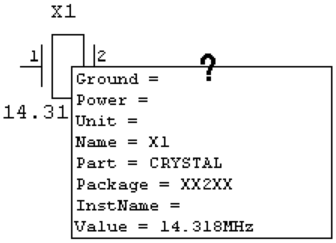
See more information on signal drawing modes in Chapter IV - Schematic Editing.

Draw Bus

The Draw Bus command is equivalent to clicking on the ✚ icon in the tool palette. Bus drawing mode behaves exactly like signal drawing mode except that a bus line is created instead of a signal line.

Attribute Probe

The Attribute Probe command displays the ? cursor and is equivalent to clicking on the ? tool in the tool palette. In this mode, clicking on an object displays a small window containing the contents of attribute fields associated with the object, as in the following example:



```
Ground =
Power =
Unit =
Name = X1
Part = CRYSTAL
Package = XX2XX
InstName =
Value = 14.318MHz
```

NOTE: The dot at the base of the ? icon is the "hot point" that determines which object is selected.

For compactness, only "primary" fields are shown and only the first 32 characters of the field value will appear. Whether or not a field is considered "primary" is determined by the setting of the "Include in Primary List" switch in the Define Attribute Fields command.

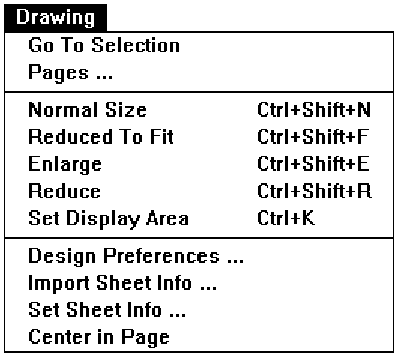
Select All

This command selects and highlights all elements in the current circuit page, except picture and text items marked as Background items. (Background objects can be selected by holding the Control and Shift keys pressed while clicking on them.) The entire page can then be operated on with the clipboard commands and others that work on selected items.

Show Tool Palette/Hide Tool Palette

This command hides or displays the floating tool palette window. It can also be hidden by selecting close from its system menu.

Drawing Menu



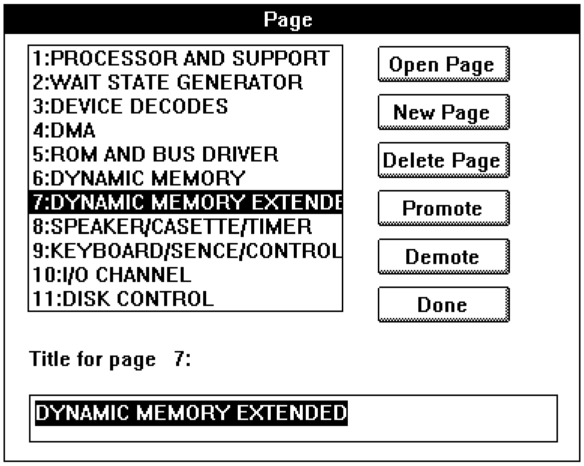
The Drawing menu contains commands for positioning and scaling the drawing on the screen, and for setting sheet size, display and printing options.

Go To Selection

This command causes the circuit position and scaling to be adjusted so that the currently selected items are centered and just fit in the circuit window. The scaling will be set to normal.

Pages...

The Page command is used to create a new page, display an existing page, change page order, or set a page title. Executing this command will produce the following box:



1:PROCESSOR AND SUPPORT
2:WAIT STATE GENERATOR
3:DEVICE DECODES
4:DMA
5:ROM AND BUS DRIVER
6:DYNAMIC MEMORY
7:DYNAMIC MEMORY EXTENDE
8:SPEAKER/CASETTE/TIMER
9:KEYBOARD/SENCE/CONTROL
10:I/O CHANNEL
11:DISK CONTROL

Title for page 7:

DYNAMIC MEMORY EXTENDED

Operations on pages are performed by selecting the pages in the scrolling list and clicking on one of the operation buttons. More than one page can be selected at a time (for all operations except setting the page title) by holding the shift keys pressed while clicking in the list.

WARNING: Operations in this box cannot be undone!!!

Open Page

The Open Page button causes the pages selected in the list to be displayed in a window. If a window is already open with the selected page, then it is brought to the front.

New Page

The New Page button causes a new, empty page to be added to the circuit and opens it in a new window.

Delete Page

The Delete Page button causes the page selected in the list to be deleted from the circuit. Note that the page must already be empty, i.e. all objects must have been removed from the page (for example using the Select All and Clear) commands before this will be allowed. Higher numbered pages will be renumbered when this page is removed.

Promote/Demote

The Promote and Demote buttons allow you to change the number order of the pages. Promote causes all pages selected in the page list to be moved one step toward the front of the list, i.e. their page number will be reduced by one. All non-selected pages will be renumbered as needed. Demote similarly moves all selected pages toward the end of the list. Promote will be disabled if the first page is selected and Demote will be disabled if the last page is selected.

Page Title

The page title text box allows a title of up to 63 characters to be entered for the selected page. This title can be accessed via the $PAGETITLE variable in text boxes.

See Chapter IV - Schematic Editing for more information.

Screen Scaling Commands

Four commands are provided which control the enlargement or reduction of the circuit diagram on the screen. These commands control screen display only and have no effect on the stored circuit information, printed output, or graphics files. The maximum enlargement allowed is 200% and the minimum level of reduction is 20% of normal size. Due to the integer calculations that are done by DesignWorks and by Windows, device symbols and text may be displayed rather crudely at scale factors other than 100%. It is best to do most editing at normal size to ensure that everything lines up as you would expect.

Normal Size

Normal Size sets the screen scale to 100%.

Reduce to Fit

Reduce to Fit sets the scale factor and centers the display so that the entire diagram fits on the screen. If the size of the diagram and the size of the window is such that this would require a scale factor of less than 20%, then the scale is set to 20% and the diagram is centered. If the diagram fits completely in the window at 100%, then the scale is set to 100% and the diagram is centered.

Enlarge

Enlarge increases the scale factor by about 20%, causing the diagram to appear larger on the screen. If this causes the setting to exceed the maximum of 200%, it is set to the maximum.

Reduce

Reduce decreases the scale factor by about 20%, causing the diagram to appear smaller on the screen. If this causes the setting to go below the minimum of 20%, it is set to 20%.

Magnify

This command provides an alternative method of zooming into and out of a selected area of the diagram. When you select Magnify, the cursor changes into the 🔍 shape.

Zooming In

Two methods of zooming in are provided:

1) Clicking and releasing the mouse button on a point on the diagram will zoom in to that point by one step.

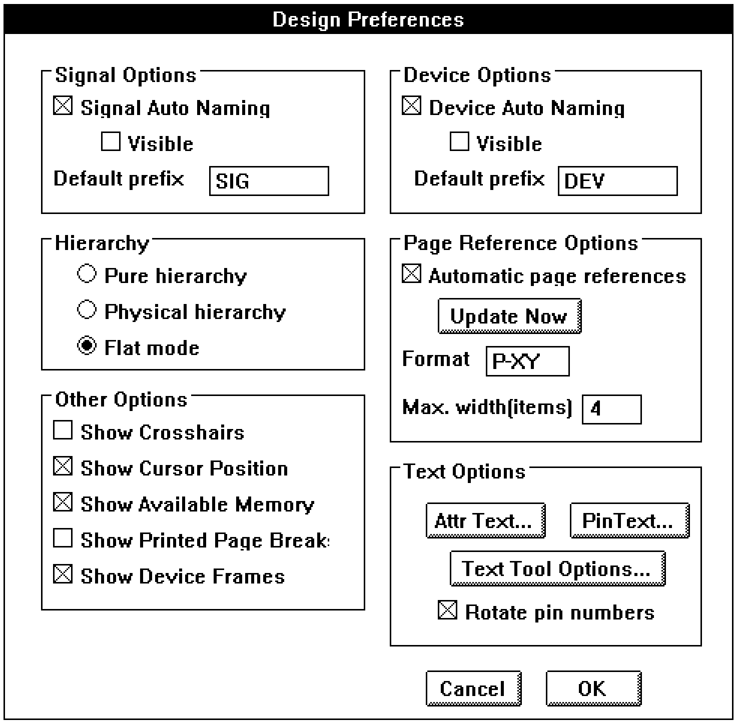
2) Clicking and dragging the mouse down and to the right zooms in on the selected area. The point at which you press the mouse button will become the top left corner of the new viewing area. The point at which you release the button will become approximately the lower right corner of the displayed area. The circuit position and scaling will be adjusted to display the indicated area.

Zooming Out

Clicking and dragging the mouse up and to the left zooms out to view more of the schematic in the window. The degree of change in the scale factor is determined by how far the mouse is moved. Moving a small distance zooms out by one step (equivalent to using the Reduce command). Moving most of the way across the window is equivalent to doing a Reduce to Fit.

Design Preferences...

The Design Preferences command is used to set a number of options which have global effect throughout a design. Selecting this command displays the following box:



Design Preferences

Signal Options
☒ Signal Auto Naming
□ Visible
Default prefix: SIG

Device Options
☒ Device Auto Naming
□ Visible
Default prefix: DEV

Hierarchy
○ Pure hierarchy
○ Physical hierarchy
● Flat mode

Page Reference Options
☒ Automatic page references
Format: P-XY
Max. width(items): 4

Other Options
□ Show Crosshairs
☒ Show Cursor Position
☒ Show Available Memory
□ Show Printed Page Break
☒ Show Device Frames

Text Options
☒ Rotate pin numbers

Signal Auto-Naming

When this option is enabled, all signals and busses will be given a default name when they are created. This ensures that every signal in a design has a unique name and can be identified in a netlist.

If the "Visible" option is selected, the default name will be displayed on the diagram adjacent to the signal when it is created. This is normally not recommended as it creates clutter on the diagram while creating and editing signals.

See more information on this topic in Chapter IV - Schematic Editing.

Device Auto-Naming

When this option is enabled, all devices (except pseudo-devices) will be given a default name when they are created. This ensures that every device in a design has a unique value in the "Name" attribute and can be identified in a netlist.

IMPORTANT: If the design is in "flat" hierarchy mode (the default) and auto-packaging is enabled (the default), then Device Auto-Naming will have no effect. In "flat" mode, both the Packager and the Device Auto-Naming feature set the Name attribute field. In this case, the Packager takes precedence.

If the "Visible" option is selected, the default name will be displayed on the diagram adjacent to the device when it is created.

See more information on this topic in Chapter V - Advanced Schematic Editing.

Hierarchy Options

The Pure Hierarchy, Physical Hierarchy, and Flat Mode buttons allow you to select the hierarchy mode for the current design.

WARNING: Changing the hierarchy mode in a non-empty design can cause some data stored in the design to be permanently deleted. Make sure you understand the impact of your choice before proceeding.

If you make a hierarchy mode change that will cause data to be discarded, you will be warned and allowed to cancel the operation.

See Chapter VII - Hierarchical Design for more information.

Page References

Automatic page reference updating is enabled whenever the Automatic Page References box is checked. This means that page reference text will automatically be placed next to any named Page Connector that is placed in the design, and will be updated whenever any page connector is moved.

If the Automatic Page References box is unchecked, page references will be updated only when the Update Now button is clicked. This can be used for large designs where page reference updating may cause delays while editing.

Page Reference Format

Two aspects of the page reference format can be controlled:

- the "Format" text item controls the appearance of each page reference in the reference list.
- the "Max. Width" item controls the number of references that will appear on each line before creating a new line. For signals with many connections, this prevents unwieldy page reference strings.

Three characters are special in the format string:

P	will be replaced by the page number
X	will be replaced by the X grid position
Y	will be replaced by the Y grid position

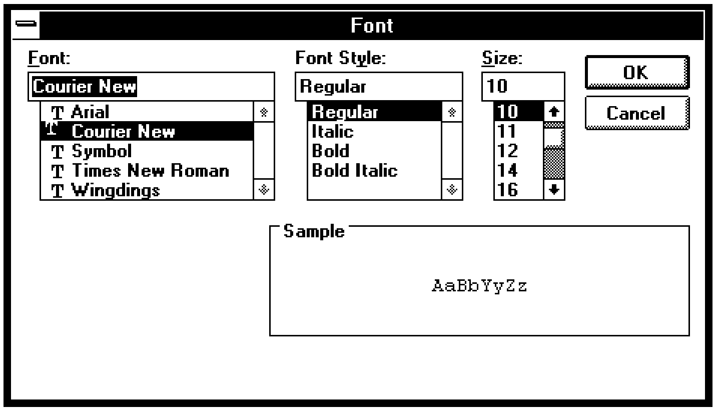
The default format is "P-XY" which generates pages references like "2-C4" (i.e. page 2, grid position C4). All characters other than the special ones above will be placed in the page reference list verbatim.

The maximum length of a format string is 16 characters.

See more information in Chapter V - Advanced Schematic Editing.

Attribute Text Options

Clicking on the "Attr Text..." button will display the following text specification dialog box:



Font

Font: Courier New

Font Style: Regular

Size: 10

Sample

AaBbYyZz

This box allows you to select the text font, size and style used for all attributes displayed in the design, including:

- Device and signal names.
- Bus breakout and bus pin labels.
- All other attributes displayed on the diagram.

but not including:

- Border text (set using the Custom Sheet Info command).
- Pin numbers (set using the Pin Text... button).

The text changes are applied when the OK button in the Design Preferences box is clicked. For larger designs, there may be a substantial delay while new positions of all displayed attribute items are calculated.

Pin Text Options

The Pin Text... button displays the same text specification box shown above for Attr Text. Any changes made to this text style will be applied to all pin numbers displayed throughout the design. The text changes are applied when the OK button in the Design Preferences box is clicked.

Allow Rotated Pin Numbers

If this item is checked, pin numbers displayed on all north- and south-facing pins will be rotated 90° to run along the length of the pin. If this item is unchecked, all pin numbers will be displayed horizontally adjacent to the pin.

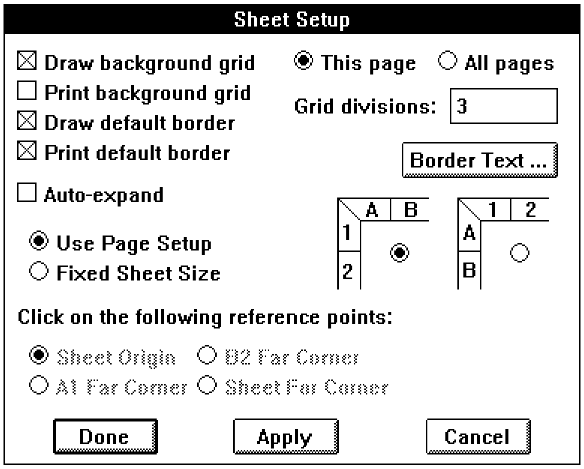
Import Sheet Info...

This command allows you to import a standard sheet template or copy the custom sheet info from any other design, or from another page in the current design.

> *See Chapter IV - Schematic Editing for more information.*

Custom Sheet Info...

This command displays the Sheet Info palette which allows custom sheet layouts to be created.



Sheet Setup

☒ Draw background grid
□ Print background grid
☒ Draw default border
☒ Print default border

● This page ○ All pages

Grid divisions: 3

Border Text ...

□ Auto-expand

● Use Page Setup
○ Fixed Sheet Size

Click on the following reference points:

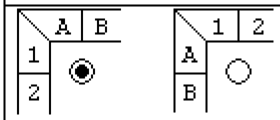
● Sheet Origin ○ B2 Far Corner
○ A1 Far Corner ○ Sheet Far Corner

Done Apply Cancel

The following table summarizes Sheet Info palette controls and options.

See Chapter V - Advanced Schematic Editing for more information on setting custom sheet information.

This Page/All Pages	These buttons determine whether the changes made will be applied to the current page only or to all pages in the current circuit level when the Apply or Done buttons are clicked.
Draw Background Grid	This switch determines whether the gray background grid is drawn in the diagram on the screen. The setting for printed output is determined independently by the "Print Background Grid" item.
Print Background Grid	This switch determines whether the background grid is included in printed or PICT file output.
Draw Default Border	This switch determines whether the default sheet border is drawn around the drawing area on the screen.

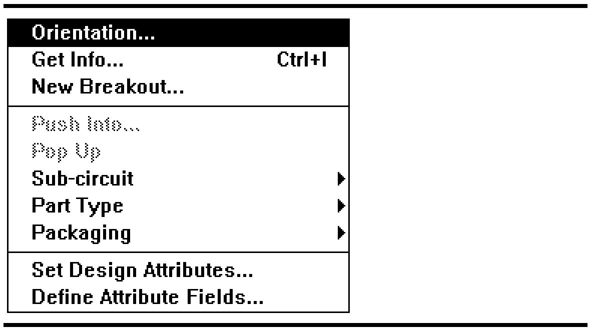
Print Default Border	This switch determines whether the default sheet border is drawn in printed or PICT file output.
Grid Divisions	This number determines how many minor grid divisions are drawn per major division. The default value is 3, entering 1 in effect disables minor grid lines. This has no effect if both Print Background Grid and Draw Background Grid are off.
Border Text	This button displays the standard text style dialog box allowing you to select the text style for the default border. This has no effect if both Print Default Border and Draw Default Border are off.
 (Grid letter/number usage)	These switches determine the usage of letters and numbers for sheet grid references. This setting will affect the position displayed on the tool palette, on Page Connectors, in the Pin List box and in report output.
Auto-Expand	If this switch is on, drawing will be allowed outside the drawing border and the drawing will be automatically expanded to multiples of the sheet size settings. NOTE: This of course means that the drawing will no longer fit on a single sheet and will have to be printed or plotted on multiple sheets.
Use Page Setup	Selecting this option causes the current printer page setup to be used to determine the sheet size. Any changes in page setup (using the Page Setup command in the File menu) will immediately affect the sheet border.
Fixed Sheet Size	Selecting this option causes fixed values to be used for the border size and position, regardless of printer page setup.
Sheet Origin	When this button is selected, clicking in the current schematic page will set the sheet origin point at that position. The sheet origin is the outer edge of the sheet in the corner containing the A1 grid position. The sheet origin can be in any of the 4 corners.

A1 Far Corner	When this button is selected, clicking in the current schematic page will set the far corner (i.e. farthest from the sheet origin) of the A1 grid rectangle. This allows the program to determine the size of the first grid space.
B2 Far Corner	When this button is selected, clicking in the current schematic page will set the far corner (i.e. farthest from the sheet origin) of the B2 grid rectangle. This is used to determine the size of all remaining grid spaces.
Sheet Far Corner	When this button is selected, clicking in the current schematic page will set the far corner of the sheet (i.e. opposite the sheet origin).
Apply	This button applies all settings currently displayed in the Sheet Info palette to the circuit. The palette remains displayed so further changes can be made. THIS CANNOT BE UNDONE!!!
Done	This button applies all settings currently displayed in the Sheet Info palette to the circuit and closes the box. These changes cannot be undone.

Center in Page

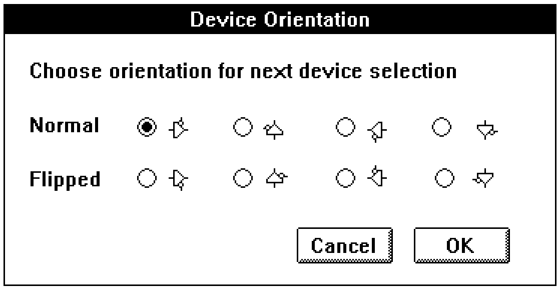
This command moves all items on the current page (except text and picture objects marked as "border" items) so that the circuit objects (taken as a group) are centered in the page. This is intended to assist with situations where a diagram has become lopsided due to modifications, or where a file imported from another system is not correctly situated when it is first read in. If the "Auto-Expand" option is turned on in the "Sheet Info" box, then this command may result in the number of pages required being reduced.

Options Menu



Orientation...

The Orientation command sets the orientation (up, down, left, right, mirrored) that will be used next time a device is created. When this command is selected, the following box is displayed:


Device Orientation

Choose orientation for next device selection

The orientation can also be changed by:

- clicking directly on the orientation icon in the tool palette, or
- using the arrow keys on the keyboard.

NOTE: Selecting any menu item cancels the device placement mode, so either of the above methods is preferred since it allows you to change orientation while placing a device or circuit scrap.

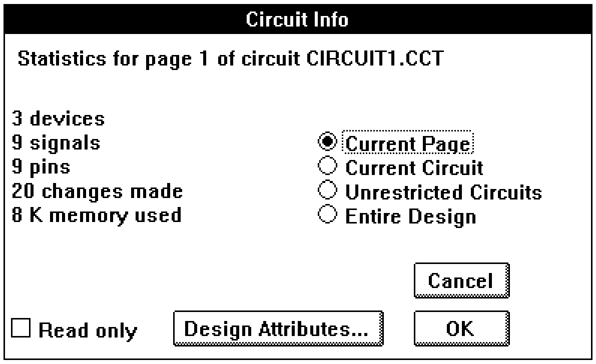
See more information about symbol rotation in Chapter IV - Schematic Editing.

Get Info...

The Get Info command is a general method of viewing and setting parameters and options that are associated with the various types of objects in DesignWorks.

Showing Design and Circuit Info

If no objects are selected in the circuit (i.e. if you have clicked in an empty portion of the diagram) then Get Info will display the following general design information box:



Circuit Info

Statistics for page 1 of circuit CIRCUIT1.CCT

3 devices
9 signals
9 pins
20 changes made
8 K memory used

● Current Page
○ Current Circuit
○ Unrestricted Circuits
○ Entire Design

□ Read only

The buttons on the right hand side of this box allow you to chose the scope of the displayed counts:

Current Page	This selection shows information only on objects displayed in the circuit page displayed in the front most window.
Current Circuit	This selection shows information on objects on all pages of the current circuit level.
Unrestricted Circuits	This selection shows information on objects in the master circuit of the design and all sub-circuits not marked as "Locked" in their respective Device Info boxes.
Entire Design	This selection shows information on all circuits in the design.

The following items of information are shown for the selected scope:

nnn devices	This is a count of devices in the selected scope. Pseudo-devices, such as page connectors and breakouts are not included. In "Current Page" or "Current Circuit" mode, this count includes devices that have sub-circuits. In "Unlocked Circuits" and "Entire Design" mode, this count includes only "bottom-level" devices, i.e. those without a sub-circuit or whose sub-circuit is not being listed.
nnn signals	This is a count of signal nets in the design, including unconnected pins. Signals that pass through more than one hierarchy level are included once for each level that they exist in.
nnn pins	This is a count of device pins, not including pseudo-devices.
nnn changes made	This is a count of editing changes made since the design was created. This is intended to allow comparisons of different versions of the same file.
nnnK memory used	This is a count of the amount of main memory occupied by the selected part of the design, in Kbytes.

Design Attributes - This button displays the standard attribute data box for the current design.

Read Only - This check box allows you to change the "Read Only" status of the current circuit level. This applies only to the current circuit level and is enabled only when the "Current Circuit" button is selected. Note that "Read Only" is not the same as "Locked" in that it allows you to view the circuit but not change it.

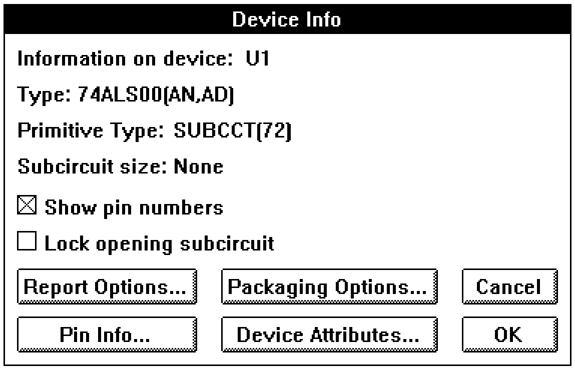
Single Object Get Info

If a single object is selected, Get Info displays a box specific to the object type. To leave any Get Info dialog box, click on the OK button.

> *More information on schematic objects is found in Chapter IV - Schematic Editing and Chapter V - Advanced Schematic Editing.*

General Device Info Box

When a normal device symbol is selected on the schematic (i.e. not a pseudo-device), then the following information box is displayed:



Device Info

Information on device: U1

Type: 74ALS00(AN,AD)

Primitive Type: SUBCCT(72)

Subcircuit size: None

☒ Show pin numbers

□ Lock opening subcircuit

The following table lists the information and options available in this box.

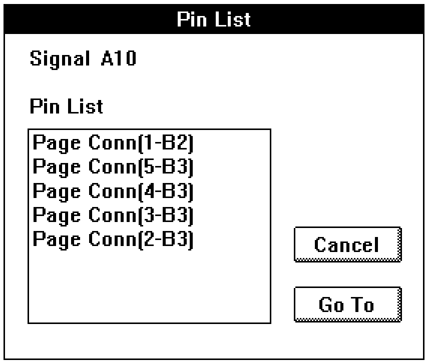
Type	This is the library type name of the device symbol, i.e. the name as it appears in the Parts palette. This is not the same as the Part attribute field, which is normally used as the part name in netlists.
Primitive Type	This is the primitive type of the symbol. For standard types, the name is shown, otherwise the name "Reserved" is shown. The ordinal number of the primitive type value is shown in parentheses.
Subcircuit Size	If the selected device has a sub-circuit, its memory size is shown in Kilobytes.
Show Pin Numbers	This switch allows you to disable the display of pin numbers for the entire device. This is intended for discrete components or others where pin numbers are not normally shown on the diagram.
Lock Opening Subcircuit	This switch allows you to prevent the sub-circuit (if any) of this device block from being opened for editing.
Report Options	This button allows you to select whether this device or its internal circuit will be listed in a netlist.

Pin Info	This button displays the Pin Info box (described below) for the first pin on the device. Buttons on the Pin Info box allow you to sequence through the other device pins.
Packaging Options	This button displays the current packaging level for this device. The meanings of these settings are described in Chapter V - Advanced Schematic Editing.
Device Attributes	This button displays the standard attribute edit box for the device.

NOTE: Clicking Cancel on the Device Info box does not cancel changes that were made in other boxes displayed using Device Info option buttons.

Page Connector Info Box

If a Page Connector device is selected in the schematic, the Get Info command displays the following box:



Signal A10

Pin List

Page Conn(1-B2)
Page Conn(5-B3)
Page Conn(4-B3)
Page Conn(3-B3)
Page Conn(2-B3)

This is essentially the same box as is displayed for the signal Pin List command, except that only Page Connectors are listed. Following on each item is the page number and grid reference of the item.

To display the selected page connector, either:

- select the item in the list and click the Go To button, or
- double-click on the item in the list.

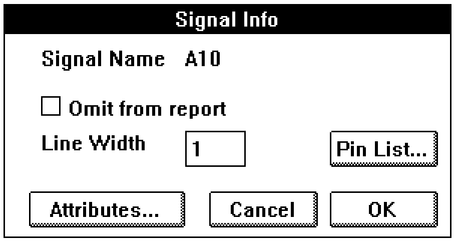
Pseudo-Device Info Box

If a pseudo-device other than a Page Connector is selected in the schematic, the signal Pin List box is displayed for the attached signal. Double-clicking on any item in this list will display the selected pin.

See more information under the Pin List command in the signal pop-up menu, below.

Signal Info Box

Selecting the Get Info command with a signal selected causes the following box to be displayed:



Signal Info

Signal Name A10

☐ Omit from report

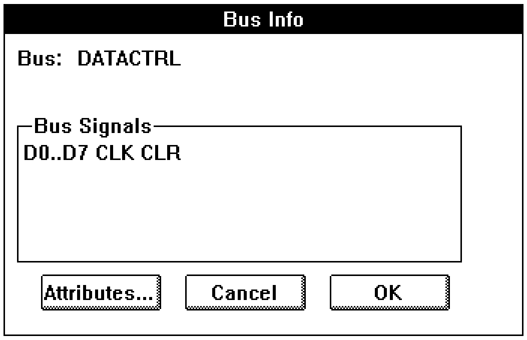
Line Width 1

The following table describes the information and options presented in this box:

Omit From Report	This switch controls whether the selected signal is included in any netlist output.
Line Width	The number typed into this box determines the displayed width of the signal. Any number > 1 displays the signal that number of times wider than the normal value. The maximum value is 255.
Pin List	This button displays a list of the real device and pseudo-device pins attached to this signal. Double-clicking on any item in this list will display the selected pin. More information on this box is given under the Pin List command in the signal pop-up menu, below.
Attributes	This button displays the general attribute data entry box for the selected signal. More information on the functions available in this box are given in Chapter VI - Attributes.

Bus Info Box

If a bus line is selected in the schematic, the Get Info command displays the following box:



Bus: DATACTRL

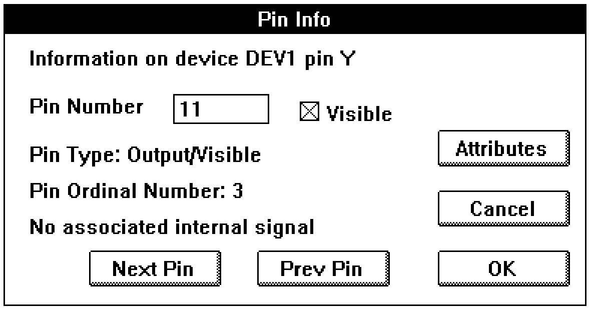
Bus Signals
D0..D7 CLK CLR

The following table describes the information and options presented in this box:

Bus Signals	This is a list of the signals contained in the bus. This list is determined by the breakouts and bus pins attached to the bus. You cannot directly change this list. See Chapter V - Advanced Schematic Editing for more information.
Attributes	This button displays the general attribute data entry box for the selected bus. NOTE: Except for the Name field, bus attributes are generally not included in any netlist output. More information on the functions available in this box are given in Chapter VI - Attributes.

General Pin Info Box

If the item selected is a device pin (non-bus and non-pseudo-device), then the following box is displayed:



Information on device DEV1 pin Y

Pin Number 11 ☒ Visible

Pin Type: Output/Visible

Pin Ordinal Number: 3

No associated internal signal

The following information and options are available:

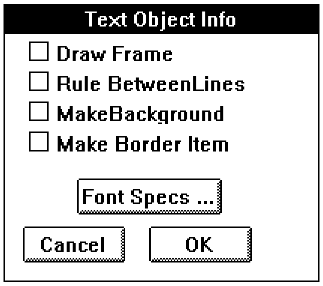
Pin Number	This is the physical pin number corresponding to this device pin. This can be empty if desired.
Visible	This check box determines whether the pin number is displayed on the schematic. For some devices, such as discrete components, it may be desirable to have a pin number associated with the pin for netlist purposes without displaying it on the diagram.
Pin Type	This information item gives the function and visible status of the pin.
Pin Ordinal Number	This number is the pins ordinal position in the device’s pin list, i.e. as viewed in the DevEditor. This can be important in some netlist formats where pin order is critical.
Associated Internal Signal	For sub-circuit devices, this item shows the name of the signal attached to the associated port connector in the internal circuit. For other devices, this will be "None".
Attributes	This button displays the general attribute data entry box for the selected pin.
Prev Pin/ Next Pin	These buttons allow you to move to the next or previous pin (by ordinal number) on the same device, without having to return to the schematic and select the pin.

Pseudo-Device Pin Info Box

If a pseudo-device pin is selected, the Get Info command displays the signal info box for the attached signal.

Text Info Box

If a text object is selected, the Get Info command displays the following box:



Text Object Info

□ Draw Frame
□ Rule BetweenLines
□ MakeBackground
□ Make Border Item

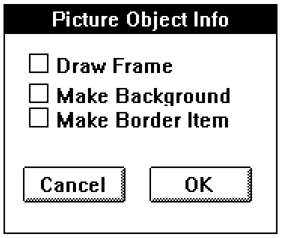
The following table summarizes the options available in this box.

See more information on text objects in Chapter IV - Schematic Editing and Chapter V - Advanced Schematic Editing.

Draw Frame	Turning this switch on causes a frame to be drawn around the text item on the schematic.
Rule Between Lines	Turning this switch on causes a line to be drawn after each row of characters.
Make Background	Turning this switch on makes the selected text block a background item, i.e. it cannot be affected by normal editing operations. In order to select it for editing, hold the Command and Option keys pressed while clicking on it.
Make Border	Turning this switch on marks this item as a border item. This means that it will be treated as part of the sheet border. I.e. it will be updated by the Import Sheet Info command, copied to new pages as they are created, and unaffected by the Center in Page command.
Font Specs	Clicking this button displays the standard text style dialog. Any changes made in text style affect only the selected item, but they also become the default for future text blocks.

Picture Info Box

If a picture object is selected, the Get Info command displays the following box:



□ Draw Frame
□ Make Background
□ Make Border Item

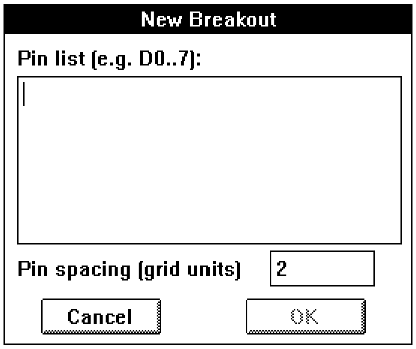
The following table summarizes the options available in this box.

See more information on picture objects for borders and title blocks in Chapter V - Advanced Schematic Editing.

Draw Frame	Turning this switch on causes a frame to be drawn around the picture item on the schematic.
Make Background	Turning this switch on makes the selected picture a background item, i.e. it cannot be affected by normal editing operations. In order to select it for editing, hold the Command and Option keys pressed while clicking on it.
Make Border	Turning this switch on marks this item as a border item.. This means that it will be treated as part of the sheet border. I.e. it will be updated by the Import Sheet Info command, copied to new pages as they are created, and unaffected by the Center in Page command.

New Breakout...

The New Breakout command is used to generate a standard bus breakout symbol for a group of signals. When this command is selected, the following box will be displayed:



New Breakout

Pin list (e.g. D0..7):

Pin spacing (grid units) 2

The breakout is created by entering a list of signals and the desired pin spacing and clicking the OK button. A flickering image of the breakout will now follow your mouse movements and can be placed and connected just like any other type of device.

> *See more information on busses and breakouts in Chapter V - Advanced Schematic Editing.*

Pin List

Type the list of desired breakout pins into this box. Rules for creating signal lists are as follows:

- blanks or commas can be used to separate individual names in this list, therefore bussed signals cannot have names containing a blank or comma.
- a range of numbered signals can be specified using the following formats:

  D0..7 or D0..D7

  is equivalent to

  D0 D1 D2 D3 D4 D5 D6 D7

  D15..0

  is equivalent to

  D15 D14 D13 D12 D11 D10 D9 D8 _ D0

  D15..D00

  is equivalent to

  D15 D14 D13 D12 D11 D10 D09 D08 D07 _ D00

  Note that the ".." format implies that bussed signal names cannot contain periods.
- the signals specified will always appear in the order given in this list from top to bottom in standard orientation. We recommend always specifying numbered signals from lowest numbered to highest, as in the first example above, since this matches the standard library symbols.
- there is no fixed limit on the number of signals in a bus, but we recommend dividing busses up by function (i.e. address, data, control, etc.) for ease of editing.
- the same signal name can appear multiple times in the list, if desired. In this case, these pins will be connected together through the bus.
- any combination of randomly-named signals can be included in the list, as in the following examples:

  D0..15 AS* UDS* LDS*

  CLK FC0..3 MEMOP BRQ0..2

Pin Spacing

The number in the Pin Spacing box will be the spacing between signal pins on the breakout symbol, in grid units. The default value is 4 to match the standard DesignWorks libraries, but any number from 1 to 100 can be entered.

Push Into...

Notes:

1) The Schematic tool uses this menu item only for sub-circuit operations. However, other program tools can attach a significance to this menu command for other types of devices having internal data.

2) If you have used the same device type in multiple places in the design, the Push Into command creates a temporary type which is distinct from all other usages. When the sub-circuit is closed you will be asked if you wish to update the other devices of the same type.

See "Internal Circuits - Editing in Chapter VII - Hierarchical Design" for more information.

This command opens the internal circuit of the given device in a separate window. This menu item will be disabled (gray) under any of the following conditions:

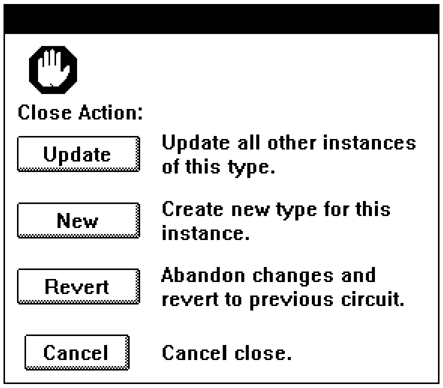
- the device is not a SUBCCT (sub-circuit) primitive type
- the device has its "restrict open" switch set in the Device Info box

If the selected device has no internal circuit, you will be asked whether you wish to create one. Clicking OK will create a circuit containing only the default port connectors matching the parent device.

Simply double-clicking on a device is a short-cut for the Push Into command.

Pop Up

This command closes the current sub-circuit and displays the circuit page containing the parent device. If the device being edited was the only one of its type, then the sub-circuit will be closed without comment. If any changes have been made to the internal circuit that would affect other devices of the same type, then the following box will be displayed:



Close Action:

Update all other instances of this type.

Create new type for this instance.

Abandon changes and revert to previous circuit.

Cancel close.

The following close actions are available:

Update	Clicking this button will cause the changes made to be copied to all other devices of the same type. This is the default action and can also be activated by typing the Enter key.
New Type	Clicking this button will make a new type definition for the device that was being edited. It will now be independent of any other devices that used to be the same type.
Revert	All changes made to the internal circuit will be discarded and it will revert to its old structure.
Cancel	Leaves the internal circuit open.

> *See more information on internal circuits and type definitions in Chapter VII - Hierarchical Design.*

Sub-circuit

This item displays a sub-menu of commands relating to sub-circuits. See the description of these command below.

Part Type

This item displays a sub-menu of commands relating to device types. See the description of these command below.

Packaging

This item displays a sub-menu of commands relating to device packaging. See the description of these command below.

Set Design Attributes

This command displays the standard attribute data entry box allowing you to enter attribute data for the design itself. Design attributes can be used to create variables in text blocks on the schematic and to insert global data in netlist output.

See Chapter V - Advanced Schematic Editing for more information on text variables.

See Chapter VI - Attributes for more information on attribute data entry.

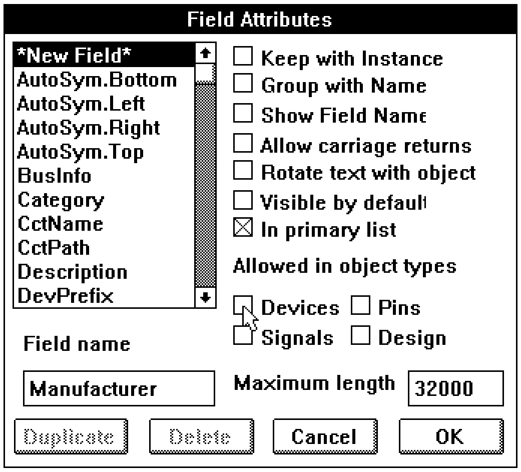
See Chapter XI - Report Generation for more information on design attributes in netlists.

Define Attribute Fields...

This command allows you to add or modify attribute field definitions for the current design. A short summary is given here.

See Chapter VI - Attributes for more information.

When this command is selected, the following box is displayed:



Field Attributes

☐ Keep with Instance
☐ Group with Name
☐ Show Field Name
☐ Allow carriage returns
☐ Rotate text with object
☐ Visible by default
☒ In primary list

Allowed in object types

☐ Devices ☐ Pins
☐ Signals ☐ Design

Field name: Manufacturer

Maximum length: 32000

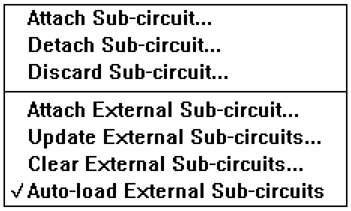
Note the following characteristics of these settings:

- • Changing the values of any of the option switches for an existing field do not affect any values already in the design. Only new data will be affected. See Chapter VI - Attributes for tips on using the Duplicate function to update the way existing data is displayed.
- • The name and option settings for most predefined fields are fixed and will be disabled in this box.
- • Regardless of changes made in this box, no design data is updated until you click the OK button.
- • At least one of the "Allowed in Object Types:" switches must be selected.

Option	**Meaning**
Keep with Instance	This option applies only in Physical Hierarchy mode. When this switch is checked, the selected attribute field may take on a different value for each instance of a device, pin or signal.
Group with Name	When this switch is checked, any value entered for this field will be displayed beneath the Name field by default. When the Name is moved, this field will also be moved.
Show Field Name	When this switch is checked, the field name will be shown with the value on the schematic.

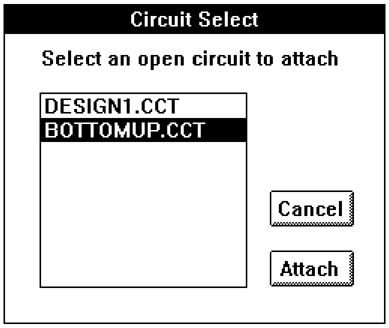
Allow Carriage Returns	If this switch is checked, then the Return key may be used to enter Carriage Return characters into this field.
Rotate Text With Object	If this box is checked, displayed values will be rotated if the associated object is rotated on the schematic. Otherwise, when the object is rotated, the text will move to an appropriate new position but be drawn in the normal orientation.
Visible by Default	If this box is checked, whenever a value is entered for this field it will be displayed on the schematic in a default position. If the field is found in the default attributes for a device, the field will be displayed automatically on the diagram when the device is placed.
In Primary List	If this box is checked, the field will be considered a "Primary" field and will be displayed in all attribute entry boxes when "Primary Only" is selected.
Allowed in Object Types	This group of check boxes allows you to select which types of schematic objects may have this field associated with them. You can check any one or more boxes.
Maximum Length	You can specify a maximum length (in characters) for the field data.
Field Name	This box allows you to enter the name of a new field or rename an existing one. If you rename a field so that it has the same name as another existing field, this becomes a Merge operation. You will be prompted to confirm this operation before proceeding.
Duplicate	This button allows you to create a new field with the same settings and (optionally) data contents as an existing one. You will be prompted to confirm the data duplication option before proceeding.
Delete	This button allows you to permanently remove a field definition and all associated data from the design. THIS CANNOT BE UNDONE!!!

Sub-circuit Sub-Menu


Attach Sub-circuit...
Detach Sub-circuit...
Discard Sub-circuit...
Attach External Sub-circuit...
Update External Sub-circuits...
Clear External Sub-circuits...
√ Auto-load External Sub-circuits

Attach Sub-circuit...

This command allows you to select an open design to attach as a sub-circuit to the selected device. When this command is selected, the following box will appear:



WARNING: This operation cannot be Undone!!!

NOTE: For best results the sub-circuit and the target circuit should have the same hierarchy mode.

Clicking Attach on the Attach Internal box will cause the following actions to be taken:

1) If the current design (i.e. the one containing the parent device) contains other devices of the same type as the selected device, then a separate, temporary type will be created for the selected device, as is done with the Push Into command, above.

2) The attribute definition table in the selected internal circuit is compared with the table in the master design. If the Attach operation would result in new fields being defined in the master design’s table, then you will be prompted for

confirmation. If you click OK, then new fields are merged into the master design's table.

3) Attribute values associated with the internal circuit are merged into the master design. If a given field already has a value in the master design, the value in the internal circuit is discarded.

4) The logical linkage between the selected device and the new internal circuit is completed. If any mismatch is detected between the port connectors defined in the internal circuit and the pins on the parent device, you will be warned.

5) The title of the internal circuit is updated to reflect its hierarchical position in the master design.

6) Auto-packaging, if enabled, is disabled since this operation potentially invalidates the design's package table.

7) The newly-attached internal circuit's window is brought to the front. It is now considered to be an internal circuit that has been opened for editing and modified. When you close the internal circuit, you will be asked if you wish to update other devices of the same type.

For more detailed information on the Attach Internals operation, see Chapter VII - Hierarchical Design.

Detach Sub-circuit...

This command makes the currently displayed sub-circuit into a separate design and redefines the parent device as having no internal circuit.

WARNINGS:

1) This operation permanently removes the sub-circuit from the selected device and all other devices of the same type in the selected design. If you do not wish to update other devices in the design, use the Make Unique command to isolate the selected device first.

2) The Detach operation cannot be Undone!!!

In particular, Detach Internal performs the following operations on the sub-circuit displayed in the front most window:

1) The circuit is unlinked from its parent device, making it into a separate design.

2) The title of the sub-circuit is set to a default "Designxxx" name.

3) The internal circuits of all other devices of the same type in the design are removed.

4) Auto-packaging, if enabled, is disabled since this operation potentially invalidates the design's package table.

> *For more detailed information on the Detach Internal operation, see Chapter VII - Hierarchical Design.*

Discard Sub-circuit...

This command is the same as Detach Sub-circuit except that it deletes the sub-circuit after it has been disassociated from the parents type. The parent device is redefined as having no internal circuit.

WARNINGS:

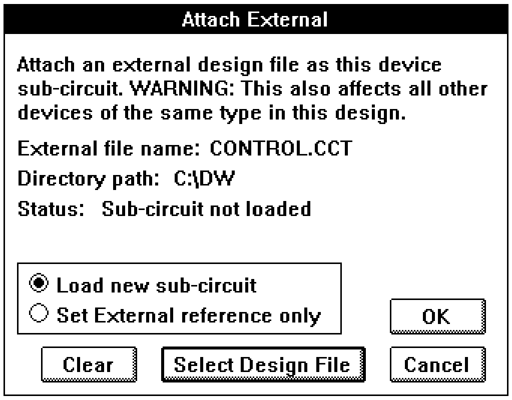
1) This operation permanently deletes the sub-circuit from the selected device and all other devices of the same type in the selected design. If you do not wish to update other devices in the design, use the Make Unique command to isolate the selected device first.

2) The Discard operation cannot be Undone!!!

> *For more detailed information on the Discard Internal operation, see Chapter VII - Hierarchical Design.*

Attach External Sub-circuit...

This command allows an external circuit file reference to be associated with the type of the selected part or for the external reference in a part to be cleared.



Attach External

Attach an external design file as this device sub-circuit. WARNING: This also affects all other devices of the same type in this design.

External file name: CONTROL.CCT

Directory path: C:\DW

Status: Sub-circuit not loaded

● Load new sub-circuit
○ Set External reference only

At the top of the dialog is displayed the current external design file associated with this part and its status. To change the associated external design file click on the "Select Design File" button. This will display a file selection box. The new file and directory selected will then be used to update the contents of the dialog.

Contained within the group box, below the status line, are two options. These control what happens when the OK button is pressed. The "Set External references only" button indicates that the external references values displayed at the top of the dialog should be saved as attributes of the selected part. This allows a part to reference an external design file with out having to load it now. The other option "Load new sub-circuit", also updates the parts attributes, but in addition also reads the design file and attaches it as the part’s sub-circuit.

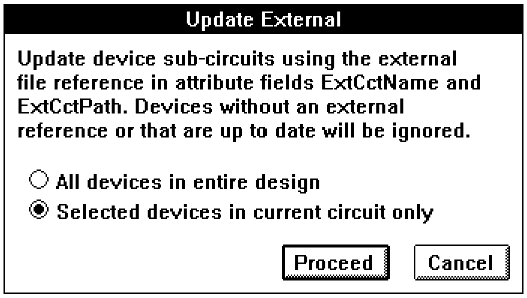
WARNINGS:

"Load new sub-circuit" attaches a sub-circuit to the selected device and all other devices of the same type in the selected design. If you do not wish to update other devices in the design, use the Make Unique command to isolate the selected device first.

The "Clear" button only removes the external reference attributes from the part. The operation doesn’t effect the part’s sub-circuit, independent of the sub-circuits internal or external origins.

Update External Sub-circuit...

This command allows the user to update part sub-circuits for parts which have external file references. The scope of the update can be controlled by the radio buttons. The first option indicates that all the devices in the entire design will be examined for external references. While the second option limits the scope of the update to only selected devices in the current circuit.



Update External

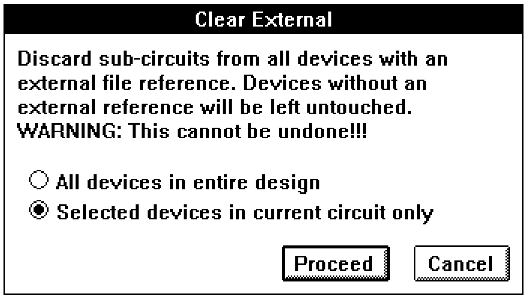
Update device sub-circuits using the external file reference in attribute fields ExtCctName and ExtCctPath. Devices without an external reference or that are up to date will be ignored.

○ All devices in entire design
● Selected devices in current circuit only

WARNINGS: This command attaches a sub-circuit to the selected device and all other devices of the same type in the selected design. If you do not wish to update other devices of the same type in the design, use the Make Unique command to isolate the selected device first.

Clear External Sub-circuit...

This command allows the user to discard part sub-circuits for parts which have external file references. The scope of the update can be controlled by the radio buttons. The first option indicates that all the devices in the entire design will be examined for external references. While the second option limits the scope of the discard to only selected devices in the current circuit.



Clear External

Discard sub-circuits from all devices with an external file reference. Devices without an external reference will be left untouched. WARNING: This cannot be undone!!!

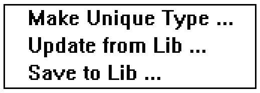
○ All devices in entire design
● Selected devices in current circuit only

WARNINGS: This command discards the sub-circuit of the selected device and all other devices of the same type in the selected design. If you do not wish to update other devices of the same type in the design, use the Make Unique command to isolate the selected device first.

Auto-load External Sub-circuit...

This menu item acts like a switch. When checked it indicates that parts that have external circuit references should have their sub-circuits loaded at the time the part type becomes first known to the design. Generally, this is the first time a part is placed. If the menu item is not checked then the parts external circuit references will not be resolved and no sub-circuit will be loaded. The sub-circuit may be loaded later by using the Update External Sub-circuit command documented above.

Part Type Sub-Menu


Make Unique Type ...
Update from Lib ...
Save to Lib ...

WARNING: The commands in the Part Type sub-menu can have major hidden effects on the structure of a hierarchical design. Do not use them without a clear understanding of what they do!

Make Unique Type...

This command makes the selected device into a unique type. In other words, even if other devices in the same design were originally drawn from the same library, this one will now be considered to be unique. This is primarily intended for hierarchical designs. This allows the internal circuit for the selected device to be modified or deleted without affecting other similar devices.

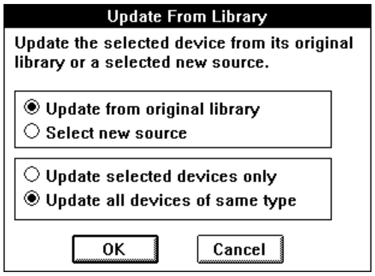
As an example, you could create a standard template for a Programmable Logic Device in a library, complete with internal simulation circuit. Several of these devices could be placed in the design, even though they will eventually have different internal programming. The Make Unique Type command would then be used to isolate them so that their internal circuits could be separately updated.

Update from Lib...

This command updates the definition of the selected device from its original source library.

WARNING: Depending upon the options selected, this command may replace the entire internal circuit of the selected device. THIS CANNOT BE UNDONE!

Selecting Update from Lib causes the following box to be displayed:


Update From Library

Update the selected device from its original library or a selected new source.

● Update from original library
○ Select new source

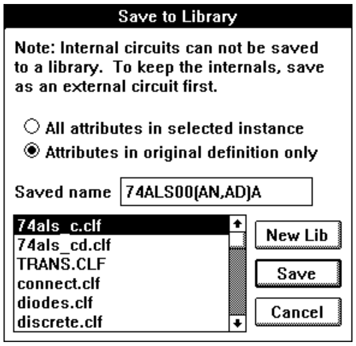
○ Update selected devices only
● Update all devices of same type

The following table summarizes the options available.

Update from Original Library	The original library and part name will be used to locate a new part definition. The LibName, LibPath, LibDate and LibType attribute fields are used.
Select New Source	A box is displayed allowing you to select any open library and part as the source for the update operation.
Update Selected Device Only	Only the selected device will be updated. NOTE: This in effect redefines the device to be different from others that were originally derived from the same library part.
Update All Devices of the Same Type	All devices of the same type will be updated.

Save to Lib...

This command saves a type definition for the selected device to a library. The following box will be displayed:



Save to Library

Note: Internal circuits can not be saved to a library. To keep the internals, save as an external circuit first.

○ All attributes in selected instance
● Attributes in original definition only

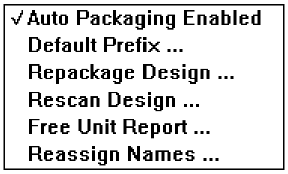
Saved name 74ALS00(AN,AD)A

74als_c.clf
74als_cd.clf
TRANS.CLF
connect.clf
diodes.clf
discrete.clf

The following table summarizes the options available.

All attributes in selected instance	If this option is selected, all the attribute values associated with the selected device will be made part of the saved library part.
Attributes in original definition only	If this option is selected, only attributes that were originally defined for the library part will be saved.
Saved Name	The part name under which the new library entry will be saved.
New Lib	This button will display a standard file save box allowing you to create a new, empty library.

Packaging Sub-Menu



The menu items in the Packaging sub-menu are used to control the automatic gate package assignment feature. The Packager can make name and unit assignments either interactively (i.e. as schematic changes are made) or in batch mode (i.e. only on command).

Following is a short description of the packaging commands.

See Chapter V - Advanced Schematic Editing for more information.

Auto Packaging Enabled

This command turns automatic device packaging on and off. When auto-packaging is enabled (i.e. the menu item is checked), the Packager automatically assigns a Name or InstName and unit number to each device symbol that is placed in the design. When disabled, the device attribute fields associated with packaging are left at their default values for new devices. Packaging can then be assigned manually by the user, or as a batch using the Repackage Design command.

NOTE: Auto-packaging cannot be enabled if the design contains packaging errors such as duplicate gate assignments or invalid packaging attribute data. If the package table is out of date when you attempt to enable packaging, the Packager will first prompt for permission to rescan the design.

Default Packaging Prefix...

This command sets the default package name prefix. This prefix is used by the Packager to generate a device name if the Name.Prefix field is not specified in the device. This value is stored in the design's PkgPrefix field.

Repackage Design...

This command forces the entire design to be repackaged, i.e. new name and unit assignments are made for all devices in the design not marked as "Locked" or "Ignore". This should be done if auto-packaging has been off while manual editing changes were made, or if parameter changes have been made that would affect packaging.

The Repackage operation performs the following steps:

- Clear the design's package table.
- Scan all devices in the design, placing all devices marked as "Lock and Check" or "Lock and Don't Check" in the package table. This ensures that these names will not be used for automatic unit assignment. Devices marked "Ignore" are not placed in the table, meaning that their names could be used in automatic unit assignment. Any errors detected in the package assignment of devices marked "Lock and Check" are reported.
- Sort all devices in the design by hierarchy level, then by page, then by horizontal grid position, then by vertical grid position.
- In the sorted order defined above, assign a new package name and unit to all devices not marked "Lock" or "Ignore".

Rescan Design...

This command forces the Packager to rebuild the design's package table from the current state of all device attribute fields. This can be used in the following circumstances:

- To check for errors or obtain a Free Unit Report after manual packaging changes have been made.
- After back annotation or any other external process that may have changed package assignments.
- After changing any of the attribute fields that may affect packaging.

If any packaging errors are encountered (e.g. duplicate gate assignment or invalid attribute fields), they will be reported at the end of this process and the design's package table will be considered invalid. No device attribute fields are affected by this command.

Free Unit Report...

This command generates a text report file listing free device units available in the current design.

Reassign Device Names...

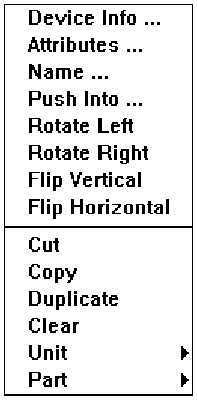
This command is a more restricted version of the Repackage Design command and can be used in any hierarchy mode. It assigns a new name to all devices in the current circuit that are either unnamed or have a default name. It is intended as a quick means of tidying up automatically-assigned names when a circuit is created or edited.

NOTE: If the design is in "Flat" hierarchy mode (the default) then this command will override the gate package assignments made by the Packager. In other hierarchy modes, the Packager stores the package assignments in the "InstName" field, which is unaffected by this command.

Note the following rules:

- Only the current circuit is affected. Other circuit levels higher or lower in the hierarchy are unaffected.
- Gate packaging information is ignored, i.e. every device symbol is assigned a unique name regardless of package availability or "locked" status.
- The "Unit" attribute field is unaffected.
- Only the "Name" field is changed, regardless of hierarchy mode.
- Names are assigned in order sorted first down the columns of the references grid, then across the page, then through subsequent pages.
- Names are assigned using the device's prefix field (i.e. the one named in the design's "PrefixField" field) or the design's "DevPrefix" field.
- Names that have been edited by the user are no longer considered "default" names and will not be changed.

Device Pop-Up Menu



The device pop-up menu is displayed by pressing the right mouse button on any device in the current circuit.

Device Info...

This command displays the general device information box. This is equivalent to selecting the device and using the Get Info command. See more information under the Get Info command above.

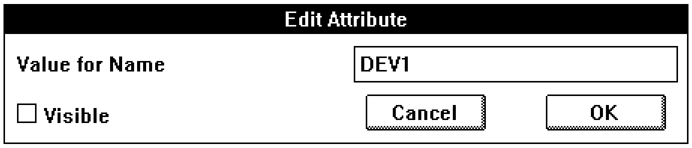
Attributes...

This displays the standard Attribute Data box allowing you to enter or edit attribute data for the selected device.

See Chapter VI - Attributes for more information.

Name...

This command displays a simple edit box allowing you to enter or edit the device name. This provides a simpler method of editing only the name, as an alternative to using the Attributes command above.



Edit Attribute

Value for Name DEV1

□ Visible

Push Into...

This command is equivalent to the Push Into command in the Options menu. See that entry for more details.

Flip and Rotate Commands

These four commands are equivalent to deleting the selected device and replacing it in the selected new orientation. If the new rotation causes any device pins to touch adjacent signal lines, connections will be made unless the Option key is held.

The Clipboard is not affected by these commands.

See more information on device rotation in Chapter V - Advanced Schematic Editing.

Cut

This is equivalent to selecting the Cut command in the Edit menu while this device is selected. The device is copied to the Clipboard and removed from the circuit.

Copy

This is equivalent to selecting the Copy command in the Edit menu while this device is selected. The device is copied to the Clipboard.

Duplicate

This is equivalent to selecting the Duplicate command in the Edit menu while this device is selected. The given device is duplicated and the program enters Paste mode immediately.

The Clipboard is not affected.

Clear

This is equivalent to selecting the Clear command in the Edit menu. The device is deleted from the circuit.

The Clipboard is not affected.

Attribute Value List Sub-menus

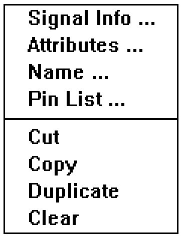
The device pop-up menu may contain one or more sub-menus as its last items. These sub-menus allow you to select from a list of possible values for a device attribute field. Selecting an item in one of these lists changes the value in the associated attribute field.

Most of the standard libraries provided with DesignWorks contain fields defining the part code, the package type and the gate unit (e.g. a, b, c, etc.). When one of these fields has more than one possible value, the list is displayed in one of these menus. For example, if the sub-menu "Unit" appears, this means there are multiple gate units available in this device type. Selecting one of the units in the Unit sub-menu causes the Unit attribute field to be changed for that device and the pin numbers to updated accordingly.

Users can add their own value list sub-menus for any attribute field where it is convenient to select one of a number of possible values. Any field defined for the design, called for example "Value", can have another field associated with ".List" appended to the name, so in this case "Value.List". The "Value.List" field should contain text items separated by commas. Each one of these items will appear on a separate line in the "Value" sub-menu in the device pop-up.

See Chapter VI - Attributes for more information.

Signal Pop-Up Menu



The signal pop-up menu is displayed by pressing the right mouse button over any signal line.

Signal Info...

This command displays the general signal information box. This is equivalent to selecting the signal and selecting the Get Info command in the Options menu. See more information on this command above.

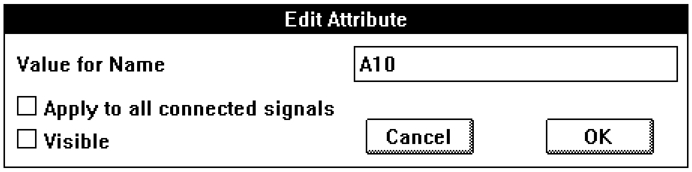
Attributes...

This command displays the general attribute data entry box for the selected signal.

NOTE: For name changes, it is best to use the Name command described below since it provides more options for applying the name change.

Name...

This command displays a name edit box for the selected signal. The following box is displayed offering you several options on how the name changed is to be applied:


Edit Attribute

Value for Name A10

□ Apply to all connected signals

□ Visible

Apply to All Connected Signals

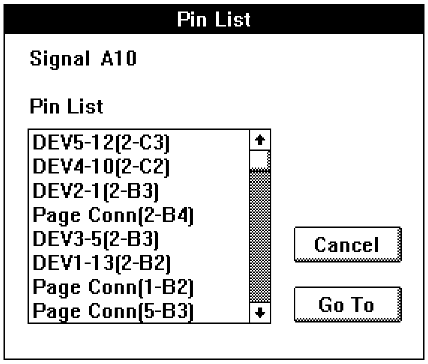
This option allows you to choose whether the name change applies only to the selected signal segment (i.e. thereby breaking its connection with other like-named signals), or to all interconnected signal segments.

Visible

This option allows you to choose whether the name entered should be displayed on the schematic. If the name was already visible and you uncheck this box, it will be removed from the schematic. In this case, the name will still be associated with the signal as an invisible attribute. If the name was not previously visible and you check this box, it will be displayed somewhere adjacent to one of the signal line segments.

Pin List...

This command displays a list of all device pin connections comprising the selected signal. Only pins in the current circuit are listed; pin connections in other hierarchy levels are not shown.



Pin List

Signal A10

Pin List

DEV5-12(2-C3)
DEV4-10(2-C2)
DEV2-1(2-B3)
Page Conn(2-B4)
DEV3-5(2-B3)
DEV1-13(2-B2)
Page Conn(1-B2)
Page Conn(5-B3)

Selecting any item in the pin list and clicking on the Go To button will cause the appropriate page to be opened (if necessary) and that pin to be displayed in the center of the circuit window. If the selected pin is invisible (e.g. a bus internal pin) the window will be centered on the parent bus pin.

For convenience in navigating a schematic, connections to pseudo-devices such as bus breakouts and ground symbols are also shown in this list. In this sense, the list is not a logical "netlist", but rather a method for checking connectivity and navigating the diagram.

Each item in the pin list is formatted as follows:

For normal devices: *device-pin* (*page ref*)

For pseudo-devices: *device type (page ref)*

The elements of this format are as follows:

device	If the device is named, this will be its name, otherwise it will be a "#" mark plus the type name.
pin	If the pin has a pin number, this will be the pin number, otherwise a "#" and the pin name.
page ref	This will be a reference to the page number and grid position, formatted as specified in the Design Preferences box.

Cut

This is equivalent to selecting the Cut command in the Edit menu while this signal is selected. The signal is copied to the Clipboard and removed from the circuit.

Copy

This is equivalent to selecting the Copy command in the Edit menu while this signal is selected. The signal is copied to the Clipboard.

Duplicate

This is equivalent to selecting the Duplicate command in the Edit menu while this signal is selected. The given signal is duplicated and the program enters Paste mode immediately.

The Clipboard is not affected.

Clear

This is equivalent to selecting the Clear command in the Edit menu. The signal is deleted from the circuit.

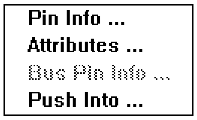
Attribute Value List Sub-menus

The signal pop-up menu may contain one or more sub-menus as its last items. These sub-menus allow you to select from a list of possible values for a signal attribute field. Selecting an item in one of these lists changes the value in the associated attribute field.

None of the predefined fields for signals have value lists associated with them. Therefore, a value list sub-menu will only appear here if user-defined value list fields have been created.

See Chapter VI - Attributes for more information.

Pin Pop-up Menu

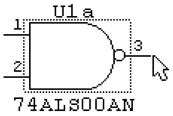


The pin pop-up menu is displayed by clicking the right mouse button on any device pin.

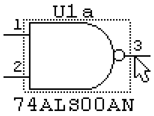
NOTE: Since an device pin is both a pin and a signal you determine whether you get the pin or signal pop-up menu as follows:

- clicking on the pin in the last 1/4 of the pin length away from the device will display the signal menu.
- clicking on the pin close to the device symbol will display the pin menu.

Selecting the Signal



Selecting the Pin



Pin Info...

This command displays a standard Pin Info box showing the name, function and internal circuit association of the pin, as well as allowing you to edit the pin's attributes.

The Next and Previous buttons on this box can be used to view and edit other pins on the same device without having to return to the schematic and select them individually.

This command is equivalent to selecting the pin and choosing the Get Info command in the Options menu.

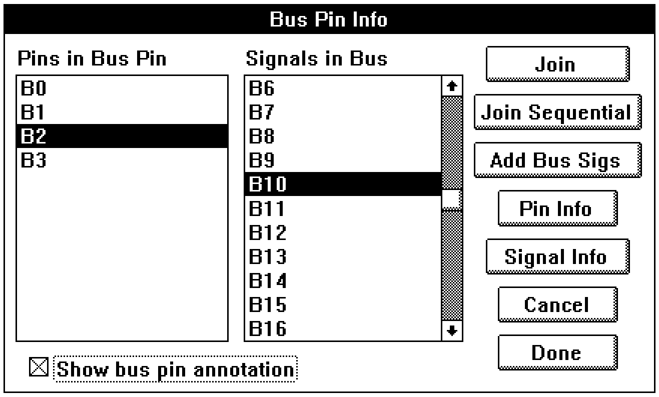
Attributes...

This command displays the standard attribute edit box for fields associated with the selected pin.

The Next and Previous buttons on this box can be used to view and edit other pins on the same device without having to return to the schematic and select them individually.

Bus Pin Info...

This menu item will be enabled only when a bus pin on a device is selected. It allows the association between the bus internal pins on the device and the signals in the bus to be changed. The following box will be displayed:



Pins in Bus Pin

B0
B1
B2
B3

Signals in Bus

B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16

☒ Show bus pin annotation

The left-hand list shows the names of the pins contained in the selected bus pin. The right-hand list shows all the signals in the attached bus. For each pin in the pin list, the signal in the same row in the signal list is the one attached to it. Signals in the signal list beyond the end of the pin list are not connected in this bus pin.

Changing Signal Connections

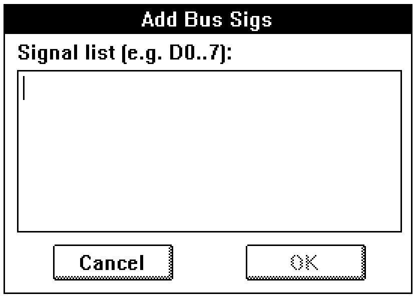
Two buttons are provided to change the association between pins and signals. The Join button causes the selected pin in the pin list to be joined to the selected signal in the signal list. If the selected signal is already attached to another pin in the list, then the signals will be swapped (i.e. a signal can only connect to one pin and vice versa). The signal list will be updated to show the new relationship.

The Join Sequential button provides a quick method of joining multiple numbered pins and signals. The selected pin is joined to the selected signal, as with Join, above. If the signal and pin names both have a numeric part, both numbers are incremented and the corresponding signal and pin are joined. This process is repeated until either the signal or pin name is not found in the list.

For example, given the lists appearing in the above picture, if pin B0 and signal B4 are selected, then Join Selected will join B0-B4, B1-B5, B2-B6 and B3-B7. Since there are no more numbered pins, the process would stop. Note that although the signal and pin names are the same in this example, this is not a requirement.

Add Bus Sigs

The Add Bus Sigs button allows you to add signals to the signal list so that they can then be joined to device pins. Clicking this button displays the following box:



A list of signals can be typed into this box using the same format as the New Breakout command. Following are examples of allowable formats:

D0..7

D0..15 AS* UDS* LDS*

CLK FC0..3 MEMOP BRQ0..2

The order of entry will affect the order that the signals appear in the list, but is otherwise not significant. For a complete description of the rules of this format, see the New Breakout command elsewhere in this chapter.

NOTE: These signals are only added temporarily. When you close the Bus Pin Info box, all signals that are not connected to any pin are removed from the bus.

Pin Info Button

The Pin Info button brings up the standard Pin Info box for the pin selected in the pin list. See the Get Info command for more information.

Signal Info Button

The Signal Info button brings up the standard signal info box for the signal selected in the signal list. See the Get Info command for more information.

Show Bus Pin Annotation

If this option is enabled, a list of the connections made in the bus pin will be displayed adjacent to the pin. This is done by creating a value in the BusInfo attribute field for the selected bus pin. This value can be edited manually, if desired, but will be updated automatically each time this box is displayed. The format of the signal list is the format used by the Add Bus Sigs option, above.

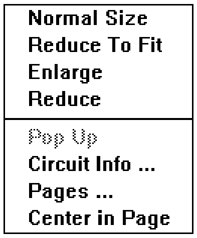
Attribute Value List Sub-menus

The pin pop-up menu may contain one or more sub-menus as its last items. These sub-menus allow you to select from a list of possible values for a pin attribute field. Selecting an item in one of these lists changes the value in the associated attribute field.

None of the predefined fields for pins have value lists associated with them. Therefore, a value list sub-menu will only appear here if user-defined value list fields have been created.

See Chapter VI - Attributes for more information.

Circuit Pop-up Menu



Normal Size/Reduce to Fit/Enlarge/Reduce

These menus items control the zoom scale factor and function exactly as the same-named items in the Drawing menu except that they attempt to zoom in around the area of the mouse click.

Pop Up

This command is equivalent to the same-named item in the Options menu. It closes the current circuit and displays the circuit containing the parent device. It will be disabled if the current circuit is the top level in the design.

Circuit Info

This command is equivalent to the Get Info command in the Options menu while no items are selected in the current circuit. It displays the circuit info box.

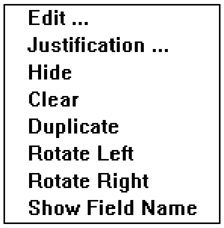
Pages...

This command is exactly the same as the Pages... command in the Drawing menu. It displays the Page box allowing you to create, delete, title and display circuit pages.

Center in Page

This menu item functions exactly as the same-named item in the Drawing menu.

Attribute Pop-up Menu



When any visible attribute data item on the schematic is clicked on with the right mouse button, the Attribute pop-up menu will appear.

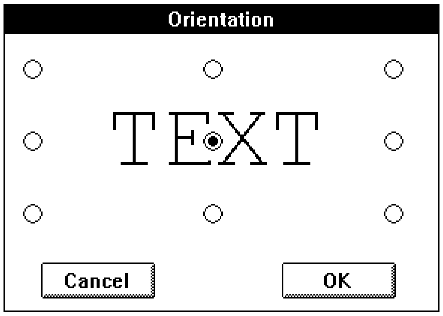
NOTE: Even though clicking on an attribute item selects and highlights the object associated with it, the commands in this menu affect only the attribute field that was clicked on.

Edit...

This command opens a text box allowing you to edit the contents of the selected field. All locations where this field is displayed on the schematic will be updated when the OK button is clicked.

Justification

This command allows you to change the vertical and horizontal justification used in the positioning the attribute text on the diagram. When this command is selected, the following box will be displayed:



The selected point on the text is considered to be the reference point for the given attribute block. This point will be kept fixed if any field value or text style changes cause the box to be resized.

For more information on attribute text justification, see Chapter VI - Attributes.

Hide

This command causes the visible attribute text that was clicked on to be removed from the schematic without removing the value from the field. I.e. if you select the associated object and open the attributes edit box, the same value will still be present. If the field was displayed in more than one place, only the one clicked on will be removed.

Clear

This command causes the value for this field to be set to null and all visible occurrences of it on the schematic to be removed.

Duplicate

This command creates another visible occurrence of the same attribute field. This text can then be dragged or rotated to any desired position on the schematic.

Rotate Left/Rotate Right

These two commands cause the single visible attribute text item that was clicked on to be rotated in the given direction.

Show Field Name

This command allows you to display the field name with the value on the schematic. When this item is checked, the display will be in the form "fieldName=value". Selecting this command again will cause the display to revert to the normal value display. This command applies only to the selected field on the selected object.

Chapter X - Device Symbol Editing

Introduction

The Device Editor tool is used to create device symbols, hierarchy blocks or general graphics (e.g. title blocks or mechanical drawings) for use on DesignWorks schematics. It provides a complete, object-oriented drawing environment with standard drawing tools, as well as specific functions tailored for symbol creation.

Installation

The DevEditor is a code module that works with DesignWorks using the MEDA interface. In order to be located when DesignWorks starts up, it must be in the "Tools" directory, or in the folder specified in the TOOLFOLDER statement in the DW.ini file.

Starting the DevEditor

<u>Creating a New Part</u>

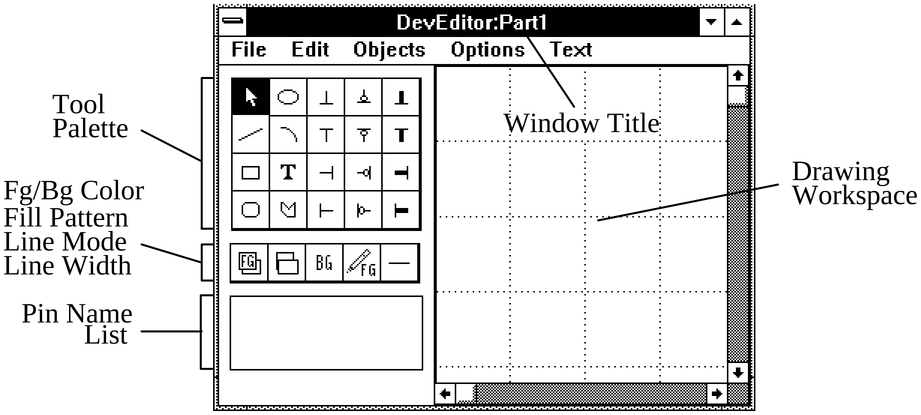
Select the name DevEditor in the Tools menu. This will open an empty graphics window, allowing you to create a new device type from scratch. If there is already an open DevEditor window this will bring it to the front.

<u>Editing Existing Part</u>

1) Select the part to be edited in the parts palette

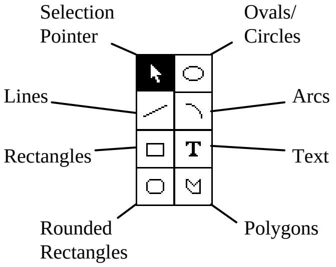
2) Choose the Edit Part command in the Parts palette pop-up menu.

Device Editor Window



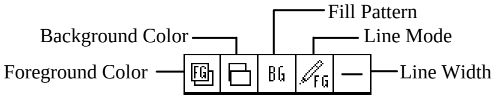
Window Title	"DevEditor:" followed by the name of the part being edited.
Tool Palette	Used to select a graphical editing or pin placement tool. The current tool is displayed selected.
Fg/Bg Color	Foreground and background pop-up palettes. The first pop-up is used to set the foreground color of a graphic object, the second sets the background color for an object.
Fill Pattern	Pop-up palette. Graphic objects are filled with this pattern.
Line Mode	Pop-up palette. Graphic objects have their edges drawn with this mode.
Line Width	Pop-up palette. Used to select the width of lines.
Pin Name List	Scrolling list of names for all of the pins which exists. A pin exists as long as it is in the pin name list. It need not have a graphic representation to still be valid.
Drawing Workspace	This area is used to construct the graphical appearance of the part.

Drawing Tools



The 8 items in the left half of the tool palette represent standard drawing tools.

Graphic Palettes

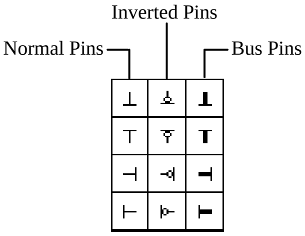


The foreground color, background color, fill pattern, line mode, and line width palettes change how the selected graphical objects are drawn. If there are selected objects when a palette is changed then the selected objects will be given the characteristic selected. If there are no graphical objects selected when a change is made to a palette, the default setting for the palette is changed. Graphical objects created in the future will use the new default setting.

When no objects are selected the palette shows the default settings. If a single object is selected the palette shows that object's settings. If multiple objects are selected then the default settings are displayed.

Fg & Bg Pop-up	The fg and bg color palettes determine the colors used for the "foreground" (i.e. "black" areas of lines, text and fill patterns) and "background" (i.e. "white" areas of text and fill patterns).
Fill Pattern Pop-up	The fill pattern palette determines the pattern used to fill the inside of ovals, rectangles and polygons..
Line Mode Pop-up	The line mode palette sets the pattern used for all outer framing lines on objects.
Line Width Pop-up	The line width palette determines the width of lines and object frames.

Part Pin Tools



The left row of pin tools are used to create pins on the part in any of the four orientations.

The second column of pins are provided as a convenience to create pins with an inversion bubble. A pin created with the inverted pin tools has no special characteristics, these are only provided as a graphical convenience.

The third column of pins are used to place bus pins on the part in any of the four orientations. Bus pins allow buses to be connected to the part. A bus pin's functionality is determined by the internal pin names it contains. See the Add Pins command below for information on creating bus internal pins.

When placing pins, a graphical pin is associated with a name in the pin list. The association is made by applying the following rules and using the first one that matches.

1) Associate a selected pin name in the pin list which is unplaced and is of the same type. ie.: normal and inverted pins can't be associated with bus pin names or internal bus pin names.

2) Associate an unplaced pin name of the correct type which follows the first selected pin name. Pin names are examined in a cyclic order so if the bottom of the pin name list is reached the search continues from the beginning.

3) If no association is made then a new pin name of the correct type is created and added to the bottom of the list.

Pin Name List

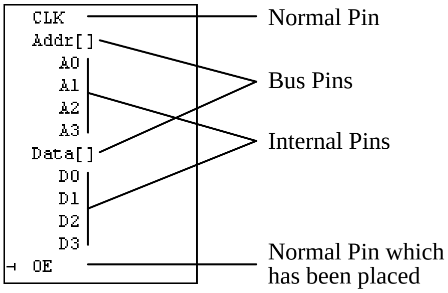
The pin name list box contains a scrollable list of the pin names associated with this device. This list is derived from the following sources:

- if the symbol was opened from an existing part, then the initial name list will be the pin names associated with the part.

- if a pin tool is clicked in the drawing workspace, a new name may be added with the form "PINxx" where xx is a sequential number. This is only done if no unplaced pin name could be used.
- if a saved DesignWorks circuit is selected as an external circuit, then the names of the port connectors in the circuit will be merged with the names in the pin name list.
- the Add Pins dialog can be used to create a list of pins to be merged with the pin names in the pin name list.
- the Auto Create Symbol command can be used to specify all the pin names and pin numbers for a device symbol.

The pin name that appears in the pin name list does not need to be the same as the graphical annotation which appears next to a graphical pin in the part's symbol. Some operations performed by the DevEditor - Auto Symbol Create - assume a correspondence but this is only done as a convenience. The result is that changing a pin name in the pin name list will never cause the annotation next to a graphical pin to automatically change.

Any pin name that has been placed (i.e. it has a corresponding graphical pin object which appears as part the symbol) will have a "⊣" mark beside it. Bus pins are marked in the list with "[ ]" following their name and the are displayed in bold. Internal pin names (i.e.: pin names which define the contents of a bus) are indented when displayed.



```
  CLK
  Addr[]
    A0
    A1
    A2
    A3
  Data[]
    D0
    D1
    D2
    D3
⊣ OE
```

If a placed pin name (i.e.: it has an associated graphical pin, and a "⊣" mark next to its name) is selected then both the pin name and the graphical name will be selected. The opposite is also true. If a graphical pin is selected so will its pin name. If a pin name is selected which has not been placed, then only the name is selected. Internal pin names never have graphical associations, therefore selecting an internal pin name never selects a graphical pin.

The relationship between pin names and graphical pins is not symmetrical. Every graphical pin must have a pin name, but every pin name may not have an associated graphical pin. This can lead to some surprises. For example,

- selecting all graphical pins in the drawing workspace may not select all pin names, unplaced and internal pin names will not be selected.
- deselecting all graphical pins in the drawing workspace may not deselect all pin names. Additional unplaced or internal names may have been initially selected.

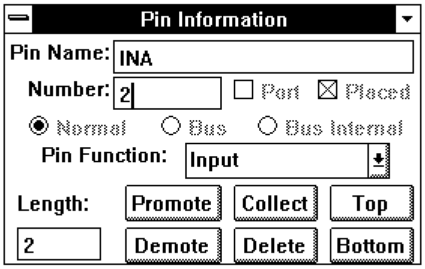
Shift Key Usage

The Shift key on the keyboard serves two purposes in the Device Editor:

- holding the Shift key pressed while clicking on an object adds to the current selection (i.e. selects the object without deselecting the other selected objects).
- holding the Shift key while using the graphical tools may augment the operation of the selected tool.

Pin Information Window

To make the floating Pin Name Info window appear double click on any pin name in the pin name list. The window will display the state of the selected pin names.



Pin Name: INA

Number: 2 ☐ Port ☒ Placed

● Normal ○ Bus ○ Bus Internal

Pin Function: Input

Length: 2

The floating palette is divided into two parts. The top is used to examine and modify the characteristics of a pin name. If more then one pin name is selected in the pin name list then the information in the window can not be changed.

The bottom of the pin name info window is used to modify the position and existence of pin names in the pin list. All of the buttons in the lower half of the window affect all of the selected pin names.

Pin Information Fields & Buttons

Pin Name

The name field has room for a 16 character identifier. The name must be unique within its context. The context for normal and bus pin names is all other normal and bus pin names. The context for a internal pin name is all other internal pin names in the same bus, except when the internal pin name is not yet in a bus. In this case its context is the same as normal and bus pin names. This means that regular pin names and/or bus pin names may not be the same, but that internal pin names only need to be unique within their own bus.

If only one pin name is selected in the pin name list then the name field can be changed. The change to the name will be applied when a new selection is made in the pin name list or when a carriage return is typed. If the name is not acceptable for some reason a beep will occur and the name will not be changed.

If more then one pin name is selected in the pin name list then the name field will show the first pin name selected, but the contents can not be changed.

Pin Number

A four character identifier is displayed in the pin number field. Even though this is referred to as a number it may contain any valid character. For example, "10", "14", "E", and "In" are all valid pin numbers. The value entered for the number is displayed on the stem of a pin.

If more then one pin name is selected in the pin name list then the number field will show the number of the first pin name selected, but the contents of the field can not be changed.

Normal, Bus, Bus Internal Buttons

The three radio buttons marked "Normal", "Bus", and "Bus Internal" show the current type of the pin name. If the pin name is not associated with a graphic pin then the radio buttons are enabled and may be changed. Changing the radio buttons causes the pin name to be displayed differently in the pin name list. If the pin name is associated with a graphic pin then the radio buttons will not be enabled but will display the current state of the pin name.

If more then one pin name is selected in the pin name list then the buttons will show the type of the first pin name selected, but the selected button can not be changed.

Pin Type List

The pin type pop-up list allows the type of the pin to be specified. The pop-up will only be displayed for pin names of type "Normal" or "Bus Internal". Pin names of type "Bus" do not have their own type.

Using the pop-up list the following functions may be assigned to the pin name: Input, Output, Tristate, Bidirectional, Open Collector, Open Emitter, Tied High, Tied Low, Latched Input, Latched Output, Clocked Input, Clocked Output, Clock Input, and No Connect.

The pin type is important to simulation - is this pin driving or driven, and additionally can be important in error checking and report and netlist generation.

See Appendix A - Device Pin Types for more information on pin types.

If more then one pin name is selected in the pin name list then the pin type pop-up list will not be displayed.

Port Checkbox

The port checkbox can not be changed by the user and is only a status indicator. When checked it indicates that the pin name matches a port name in an attached subcircuit. If the box is not checked and you believe it should be then one of the following is the cause.

> One, no attached external subcircuit.
>
> Two, no port with the same name in the subcircuit.
>
> Three, the pin name is an internal pin name and the pin name’s bus name is not the same as the bus port in the subcircuit.

Placed Checkbox

The placed check box can not be changed by the user and is only a status indicator. When checked it indicates that the pin name is associated with a placed graphical pin.

Pin Name List Buttons

Top & Bottom Buttons

The top and bottom buttons move the selected pin names in the pin name list to the top or bottom of the pin name list, respectfully.

Promote & Demote Buttons

The promote and demote buttons move the selected pin names in the pin name list up or down one pin name in the pin name list, respectfully. If a selected pin name reaches either the top of the list when promoting, or the bottom of the list when demoting, then it will stop and the remaining selected pin names will begin to collect at the top or bottom of the list, respectfully.

Collect Button

The collect button causes all of the selected pin names in the pin name list to be grouped following the first selected pin name.

Delete Button

The selected pin names in the pin name list and there associated graphic pins are deleted after the user is prompted to see if this is ok.

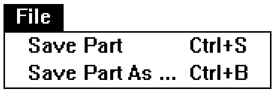
Changing the Selection

The pin name displayed can be changed by clicking in the pin name list or a keyboard short cut may be used if the window has the text focus.

Pressing the enter key moves to the next pin name, shift/enter moves backwards one pin name.

Chapter X

DevEditor Menu Commands



File Menu

Save Part

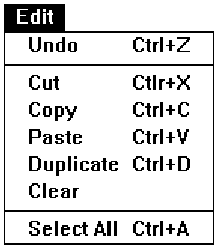
The Save Part command saves the contents of the current DevEditor window back to the library it was read from. If the open part was not read from a library (i.e. it was just created), this item will be disabled.

See How To Procedures / Editing an Existing Part / Saving the Part Again.

Save Part As

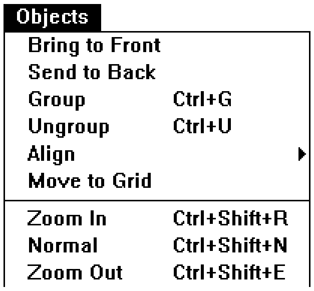
The Save Part As command brings up a dialog which allows you to select an open library and the name for the part.

See How To Procedures / Creating a New Part / Saving the Part.



Edit Menu

The standard Edit menu commands Cut, Copy and Paste can be used to move objects inside and between the DevEditor window and the DesignWorks circuit windows. Graphic objects created by other programs are not supported by the current version of DevEditor and will not appear if pasted into the DevEditor workspace area.



Object Menu

This menu contains all of the DevEditor graphic object specific operations. Each menu item will be discussed below.

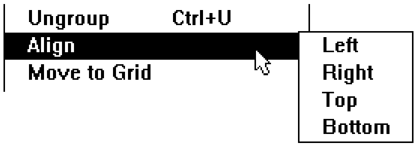
Bring To Front/Send To Back

These commands are used to set the front-to-back ordering of the selected objects relative to the other graphic objects.

Group / Ungroup

These commands allow you to make multiple graphic objects, except pins, be treated as a single object or visa versa.

Align



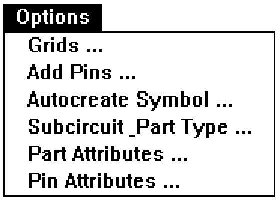
The Align sub-menu allows you to pick how the selected objects will be aligned. For example, Align Left causes all of the selected objects to be moved such that their left edges are aligned with the left most selected object’s left edge.

Move To Grid

This commands allow you to snap graphic objects to the grid.

Zoom In / Normal / Zoom Out

These commands allow the user to adjust how close-up an object is viewed.

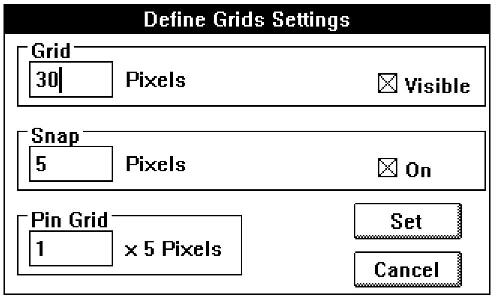


Options Menu

This menu contain options which may be used to create parts. Each menu item will be discussed below.

Grids

This command allows the user to specify the visible grid spacing and the snap-to grids for objects drawn using the drawing tools.



Grid
30 Pixels ☒ Visible

Snap
5 Pixels ☒ On

Pin Grid
1 x 5 Pixels

Grid Visible Checkbox	This check box determines whether visible grid lines are shown in the drawing workspace of the DevEditor window. The spacing between these grid lines is determined by the value in the "Grid Pixels" field.
Snap On Checkbox	This check box determines whether the corners of objects made with the drawing tools are moved to the nearest grid point.
Grid Pixels	This number determines the spacing between the visible grid lines. The units are in screen dots at the default zoom level.

Snap Pixels	This number determines the spacing between snap-to points for the drawing tools (not including pins). This does not affect objects that have already been placed. The units are in screen dots at the default zoom level.
Pin Grid	This number determines the snap-to grid for device pins. The value will be multiplied by 5 to meet the DesignWorks pin grid requirements. The units are in 5 screen dots at the default zoom level.

Add Pins

This command bring up the Add Pins floating palette.

See How To Procedures / Adding Pin Names.

Auto Create Symbol

This command brings up the automatic symbol creation dialog.

See How To Procedures / Automatically Creating Symbols.

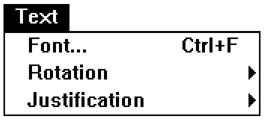
Subcircuit & Part Type

The dialog displayed when this command is selected allows you to specify the type of DesignWorks part being created. The DesignWorks types are: No Subcircuit, External Subcircuit, Symbol Only, Behavioral, and Primitive.

See How To Procedures / Selecting the Subcircuit / Assigning a Primitive Type.

Part and Pin Attributes

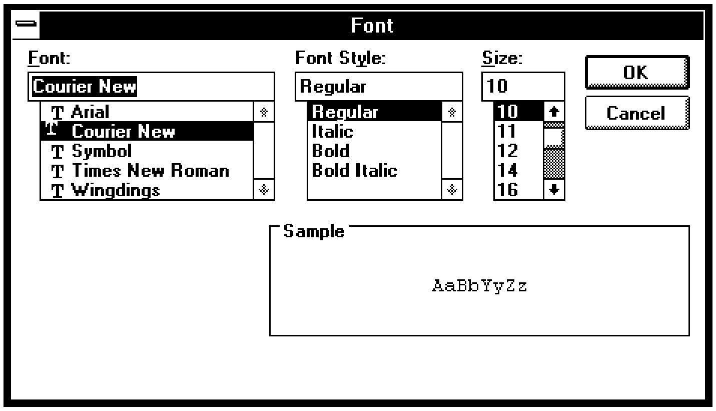
See How To Procedures / Assigning Attributes and Chapter VI - Attributes.



Text Menu

This menu contain items associated with the appearance of text.

Font



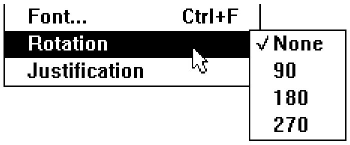
Font: Courier New

Font Style: Regular

Size: 10

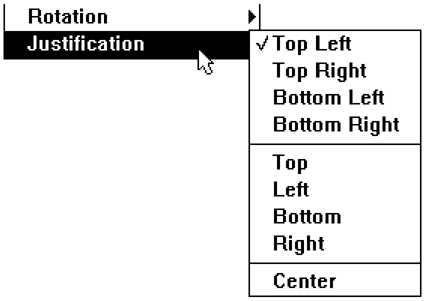
This menu item allows the font, style, and size for the selected text objects to be set. If no objects are selected then the selected text property becomes the new default.

Rotation



These menu items set the rotation characteristics for the selected text objects. If no objects are selected when an item from a menus is selected then the selected text property becomes the new default.

Justification



These menu items set the justification characteristics for the selected text objects. If no objects are selected when an item from a menus is selected then the selected text property becomes the new default.

DevEditor Procedures

Creating a New Part

NOTE: This procedure describes the creation of the simplest possible device symbol. If you are using the gate packaging or auto-naming features of DesignWorks, refer to the section Adding Packaging Information, below.

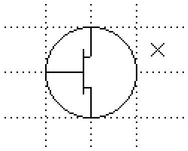
To create a new part, select the DevEditor item in the Tools menu. If there were no open DevEditor windows then this will open an empty drawing window with a default name and an empty pin list.

If there already existed a DevEditor window then selecting DevEditor in the Tools menu brought this window to the front. You can get a new DevEditor window by closing the open DevEditor windows and again selecting DevEditor from the Tools menu.

Drawing the Graphics

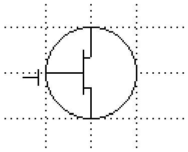
Draw the graphical shape which represents the part. Do this using the line, rectangle, rounded-rectangle, oval/circle, arc, and polygon tools.

NOTE: Device pins must be added using the pin tools. Do not draw any of the part’s pins with the graphic tools mentioned above.



Adding Pins

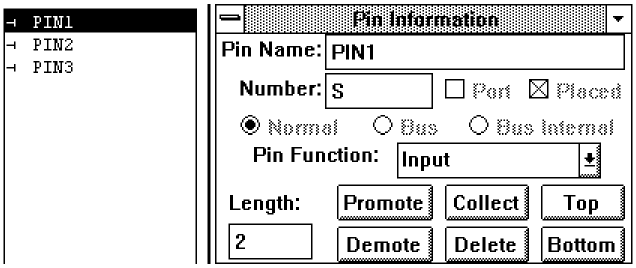
Place pins on the part use the 12 pin placement tools. The first column allows placement of normal pins. The second column provides pins with inversion bubbles. The last column of pin tools are used to place bus pins.



Note that each time you place a pin a new default pin name is added to the pin name list (See Part Pin Tools for more details).

The pin name and other pin information can be changed using the pin name info window (See Pin Name Info Window). For this example we want to name the pins "Source", "Gate", and "Drain". In addition we want to give them the following pin number "S", "G", and "D".

To change the information about each pin we first bring up the Pin Name Info Window by double clicking on a pin name in the pin name list. Double click on the first pin. When placing the pins we placed the top pin, then the bottom pin and finally the pin on the left. Therefore, the names in the pin name list are in the same order.



PIN1
PIN2
PIN3

Pin Name: PIN1
Number: S ☐ Port ☒ Placed
● Normal ○ Bus ○ Bus Internal
Pin Function: Input
Length: 2

Enter the name "Source" and the number "S" for the first pin we placed. You can use the tab key to move between the name and the number field. When you have entered both the name and number - Press the Enter key. Notice that the name was updated in the pin name list and that the pin number "S" has been displayed with the graphic pin.

Pressing the Enter Key copies the data to the pin and then advances to the next pin in the list. To change to a different pin you can; click on the graphic pin; click on the pin name in the pin name list; or press the Enter Key until the correct pin is displayed. Holding the Shift key when pressing the Enter key will cause pins to be selected in reverse order.

Notice that the pin whose information is displayed in the Pin Name Info window is also selected in the pin name list and if it has been placed then its graphic image is also selected. This allows easy verification that the correct pin is being renamed.

You should now see PIN2 selected in the pin name list, the bottom graphic pin selected, and the name PIN2 in the Pin Name Info window. Do as describe above for each of the remaining pins.

When you are finished with the Pin Name Info window it may be closed by selecting close from the system-menu in the windows top-left corner or since it is a floating window it can be just moved to the side.

Note: The actual names given to the pins in this simple part are not critical. It is a good idea though to use meaningful names for the following reasons:

- These names will be seen in the Schematic tool’s pin info dialog.
- They can be extracted in netlist output.
- They are used when binding pins to ports in subcircuits.
- In the case of bus internal pins, their names are used when connecting bus pins on the part to busses signals.

Creating a Part With Subcircuit

Port Interface

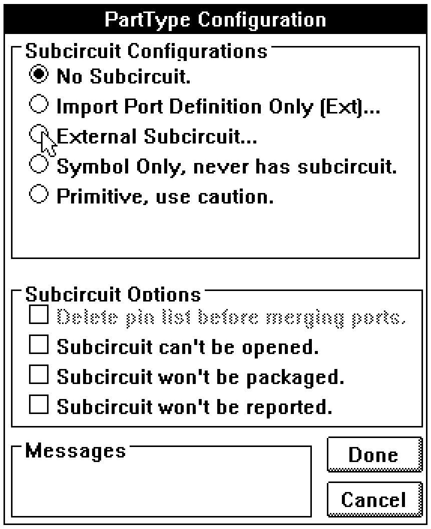
A part’s subcircuit is a schematic circuit which is associated with the part such that it is considered to be the part contents. A circuit which is to be used as a subcircuit must include parts called port connectors. Port connectors, which are named, allow signals in the circuit to be associated with pins on an enclosing part. Port connectors make this association by name, i.e.: a port connector named "A0" will only associate with a pin with the same name.

> *See Chapter VII - Hierarchical Design for more information on the port interface.*

Selecting the Subcircuit

The easiest method to create a part with an associated subcircuit is to begin by selecting the schematic circuit to be used as the subcircuit. This sounds like stating the obvious, but creating the part in any other order is more work.

To select the circuit to be used as the part’s subcircuit you must also define the type of part being made. Select the Subcircuit & Part Type command from the DevEditor menu.



PartType Configuration

Subcircuit Configurations

- ● No Subcircuit.
- ○ Import Port Definition Only (Ext)...
- ○ External Subcircuit...
- ○ Symbol Only, never has subcircuit.
- ○ Primitive, use caution.

Subcircuit Options

- □ Delete pin list before merging ports.
- □ Subcircuit can't be opened.
- □ Subcircuit won't be packaged.
- □ Subcircuit won't be reported.

Messages

This dialog allows you to select among several different options:

No Subcircuit	This is the default. It indicates that the part being made has no associated subcircuit, but it doesn't rule out that a subcircuit may be attached by the schematic capture section of DesignWorks.
Import Port Definition Only (Ext)	This is similar to the No Subcircuit option. No subcircuit becomes associated with the part, but a port interface is extracted from a external circuit (saved circuit) to define the port connections. The port names are read from the circuit without attaching the circuit to the part. Like the above option, this doesn't stop a subcircuit from being attached later. Since a subcircuit wasn't attached to the part, the port interface is only associated with the part for the duration of the editing session.
External Subcircuit	This option prompts for a saved circuit. The selected circuit becomes associated with the part being created as its subcircuit. A file reference to the circuit file will be saved with the part in the library. The circuit's port interface is imported so that the names of the ports are available in the DevEditor.

Symbol Only	This option is like the No Subcircuit option except its doesn't allow the schematic capture part of DesignWorks to associate a subcircuit with the part in the future.
Primitive	(See related section of this manual).

There are four subcircuit options in the lower part of the dialog:

Delete pin list before merging ports	This selection is only enabled when the selection above will result in a port interface being read. If this box is checked when the "Done" button is pressed then all of the old pin names in the pin name list will be deleted. This allows a new port interface to be brought in without any conflicts with existing pin names. If this option wasn't checked then port names would be merged with the names in the list. Duplicate pin names and their related properties remain unchanged, except they are now associated with the new port. Unmatched pin names in the pin name list remain exactly the same.
Subcircuit can't be opened Subcircuit won't be packaged Subcircuit won't be reported	These three switches define whether or not the parts subcircuit can be opened by double clicking on it; if the packager will descend into the subcircuit when repackaging a design; if the report generator will report the contents of the subcircuit in reports.

Drawing the Graphics

The graphic image of the part may be drawn in the same way as was described in the previous section on creating part's without subcircuits.

> *You can also generate a symbol automatically using the Auto Create Symbol command. See the section Auto Creating a Symbol.*

Associating Graphic Pins with Pin Names

Pin names that may be placed as graphical pins on the symbol are those that are not indented in the list and that don't have a symbolic pin symbol next to their names. Names that have a symbolic pin next to them already have a graphical association. Names which are indented are internal bus pin names and don't have a graphical representation.

Assignment of graphical positions to pin names may be done explicitly by using the pin placement tools or automatically using the Auto Create Symbol command. See Auto Creating a Symbol, below.

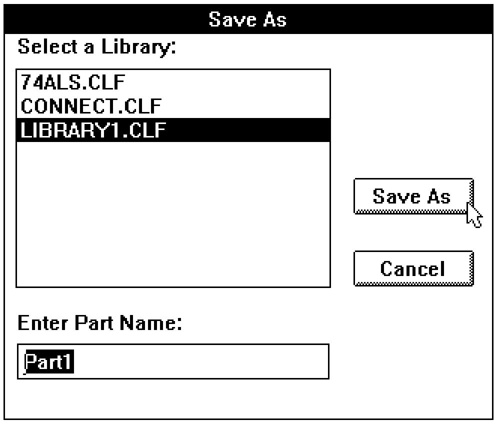
To explicitly assign a graphical pin position to a pin name requires that the pin name be selected in the list and that a graphical pin be placed using one of the pin tools.

A graphical pin assignment will only be made with an unplaced pins of the correct type. If a placed pin name is selected in the pin name list, and an attempt is made to associate a graphical pin with it, then the association will not be made with the pin name because it was already placed. Instead the new graphical pin will be associated with the next free pin name of the correct type or a new pin name will be created if there are no unplaced pin names of the correct type.

Because of the way the pin association mechanism works, i.e.: searching for the next placeable pin name, it is possible to place all pin names of the same type without having to make a new selection. Select the first unplaced pin name and place it. Now since the current selection is placed, the next graphic pin placed will be matched with the next pin name of the correct type. Place another pin and note how the selection moves forward to the next previously unplaced pin name of the same type.

Saving the Part

The newly-created part is written out to a library using the Save Part As command in the File menu.



Select a Library:

74ALS.CLF
CONNECT.CLF
LIBRARY1.CLF

Enter Part Name:

Part1

When the menu item is selected the "Save Part As" dialog will appear. It requires that a library name is selected from the list and that a name for the part is entered in the lower box. In this example the connectors library has been selected and the part’s name has been left to default to "Part1". A name entered will become the name of the part in the library and the title of the DevEditor window will be updated to correspond to the new part name. Note: only open libraries appear in the list

See Library Handling in the schematic capture manual for additional information.

Editing an Existing Part

Getting the Part from a Library

A part selected in the part palette may be edited by pressing the right mouse button down and then selecting "Edit" from the pop-up menu. The mouse must be over the parts list for the pop-up menu to appear

The part once opened is a copy of the part in the library. The part in the DevEditor window knows what library it can from and the name of the part, but it is not the same part as in the library. If a part is selected from the library while the same part is being edited the original part will be the one selected. In addition, editing a part and returning to the library does not change all the copies of that part in all the circuits that use it

See the schematic manual on methods on updating from a library.

Warnings About Editing

Changing Graphics

Editing changes can be freely made to all of the part’s graphical components other then pins. It is important to remember that graphic pins are associated with pin names and that even though graphic pins look the same they may not be interchangeable because of their associated name.

Consider the following example. A flip flop has pins named "D", "CLK", and "Q". The pins "D" and "CLK" are both left facing graphic pins and "Q" faces to the right. Simulation and netlisting problems will occur if the graphic pins "D" and "CLK" were to have their graphical locations swapped.

Swapping the pins’ positions would confuse the user because the "D" label would be located next to the graphical pin associated with the pin name "CLK".. The same would be true for the "CLK" pin.

To ensure that pins are placed correctly it is important to select the pin in the pin name list and verify that the expected graphic pin is selected.

Changing Pin Information

If the part has a subcircuit then it is important to remember that the pin types are derived from the ports in the subcircuit. Trying to change a pin’s type without also changing the underlying subcircuit is not allowed.

For parts based on a primitive types or behavioral models the functions of the pins must agree with the selected model. There is currently no check to ensure that the pin are set correctly for the model selected.

With other types of parts, Symbol Only and No Subcircuit, the types of the pins can be changed freely.

Changing Part Type

Changing a part’s type can have dramatic effects. It is recommended that a part who’s type has been changed is never saved into the same library with its original name. Doing this ensures that mistakes are easier to recover from. One thing to watch for is:

- Changing to an external subcircuit. When a circuit is associated with the part its ports are read and used to create pins with the same name. These new pins carry the type of the port and will replace like named pins that are already defined. This will result in a pin losing the type it was initially defined with.

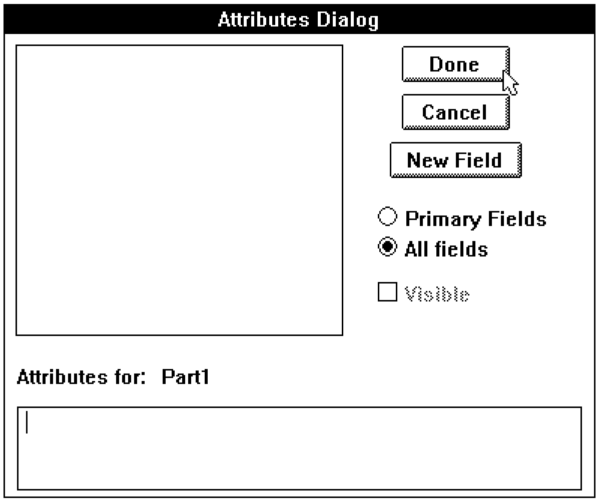
Saving the Part Again

A part opened from a library knows what library it came from. The Save Part command in the "File" menu replaces the original part in the library with the one being edited. This operation can not be undone so it is not wise to replace the original part unless you are sure it is correct. If you are not sure the part is correct, then it is better to save the part to a different library or with a different name to the same library. This allows you to test the part completely without loosing the original.

Assigning Attributes

Editing Part & Pin Attributes

To edit the attributes associated with a part select the Part Attributes command in the DevEditor Options menu. To edit the attributes associated with a part's pins select a pin in the pin name list and then select the Pin Attributes command from the Options menu. The only visible difference between the part attribute dialog, shown below, and the pin attribute dialog is the addition of "Next" and "Previous" buttons at the bottom of the window.



○ Primary Fields
● All fields
□ Visible

Attributes for: Part1

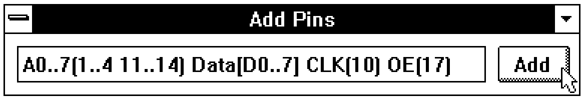
Attributes defined while in the DevEditor will be associated with the part in the library. These will become the default attributes for the part when it is placed in a circuit.

NOTE: There are numerous attributes which have special meanings and conventions for usage. Refer to Chapter VI - Attributes for a complete discussion of the setting, modification, and interaction of attributes.

Adding Pins

The Add Pins command in the DevEditor menu causes a floating palette to be displayed. Pin names the user wishes added to the pin name list, and optionally their pin number, may be entered into the palette. The pin names are created and merged with the contents of the pin name list when a carriage return or enter key is pressed or when the Add button is clicked. The Add Pins palette does not create graphic pins only pin names for the pin name list.

The created pin names are merged with the names in the pin name list. If a like named pin already exists in the list then it may be reordered to appear in the same order as in the Add Pins palette. If a pin being added has a pin number defined then this pin number will replace the pin number in the like named pin.



A0..7(1..4 11..14) Data[D0..7] CLK(10) OE(17)

Syntax for Pin Names

Pin names may be up to 16 characters long. There are three types of pin names which may be described. They are:

- **Normal Pin Names** may be specified as individual names, e.g.: A B C D, or as sequences, e.g.: A0..3. After each pin name or pin name sequence an optional pin number specification is allowed.

  A pin number specification defines the pin number(s) associated with the proceeding pin name or pin name sequence. The specification starts with an **(** and ends with an **)**. For a single pin name the pin number specification should contain a single number. For a pin name sequence multiple numbers and sequences may appear between the brackets.

  In the picture above CLK and OE both define normal pin names which have pin numbers 10 and 17, respectfully. The sequence A0..7 defines the pin names A0 through A7. These pin names were also given pin numbers. The pin numbers 1 2 3 4 11 12 13 and 14 were assigned.

  It is also possible to skip a pin name when assigning pin numbers to a pin name sequence. Consider the previous example, if we didn’t want to assign the number 11 but wanted all other pin names to get the same number, we would do the following: A0..7(1..4,,12..14) instead of A0..7(1..4 11..14). Two commas in a row causes a pin to be skipped when assigning the pin numbers. Three commas cause two pins to be skipped. Four commas, etc.

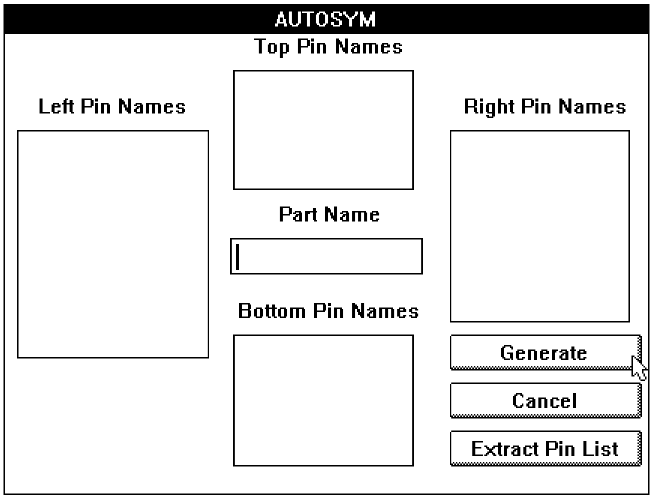
- **Bus Pin Names** are denoted by a name followed by **[...]**. Bus pins do not need to have internal pin names defined and may not have pin numbers.

- **Internal Pin Names** must appear between a **[** and a **]**. They have the same format as normal pin names and may have pin numbers. Internal pin names may be added without adding a bus name pin by not placing a bus name in front of the **[]**. For example, [A B C] adds the internal pin names "A", "B", and "C" to the pin name list.

Automatically Creating Symbols

The Auto Create Symbol command invokes the Automatic Symbol Generator which will create standard rectangular symbols given a list of the desired input and output pin names. For maximum flexibility, the symbol generated consists of separate graphic objects and is completely editable after it is generated.

The current settings for line width, fill patterns, color, text font, size and style, etc. are used in generating the symbol. The only exception to this is the type name text placed in the center of the symbol, which is written 3 points larger than the current setting and in bold.



Selecting the Auto Create Symbol command displays the above dialog box. The pin name boxes will contain the information entered the last time the DevEditor was invoked for this part. These can be modified as desired. The new settings will be merged with pin name list when the Generate button is pressed.

Entering Pin Names

The four pin name boxes allow you to specify the names of pins to appear on the left, right, top and bottom of the device symbol. The syntax described in the previous section, Adding Pin Names, is also used to define the pin names and numbers with the following extensions:

- inverted pins can be specified using the "~" character in front of the name. The inverted pin will be displayed with a bubble. The "~" will not appear in the symbol.

- items in a list can be separated by blanks or commas. Placing an extra commas between two items adds extra space between the pins on the symbol. Additional space can be added with more commas.

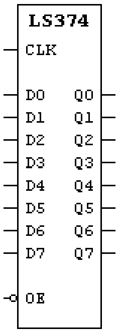
Entering the Symbol Name

The symbol name text box allows you to specify the text that will appear centered at the top of the symbol. This also becomes the new name for the part.

Generate Button

The Generate button causes the current contents of the active drawing window to be erased and replaced by the generated symbol. The pin name list will be merged with the new pins described in the dialog. This symbol consists of standard graphic objects so it can be edited using any of the drawing tools provided.

An example of a device produced by the Auto Create Symbol:



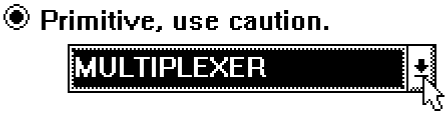
Extract Pin List Button

The Extract Pin List button updates the pin name boxes with the names extracted from the pin name list. Pins which are inputs and busses are placed on the left, outputs are placed on the right.

Assigning a Primitive Type

NOTE: The Primitive type settings should only be used with a clear understanding of their functions. Primitive types are intended primarily for use with the DesignWorks Digital Simulation option. See Appendix B - Primitive Device Types.

The primitive type of a part determines which of 38 different internal simulation model will be used to simulate the part. To describe parts which are larger or more complex a subcircuit should be used (See How To Procedures / Creating a New Part / With A Subcircuit).



● Primitive, use caution.

MULTIPLEXER

To select the primitive part type choose the "Subcircuit & Part Type" item in the DevEditor menu. Then select the "Primitive" radio button. This will cause a pop-up list to be displayed below it. The pop-up list will show the primitive type currently selected. Clicking on the pop-up list will display the other options available.

Creating a Gate

NOTES:

1) This section applies only when creating a gate for use with the DesignWorks Digital Simulator option. For general schematic use, use the default primitive type setting.

2) See information on entering gate packaging information later in this chapter.

To make an inverter (NOT gate) place 2N pins. The order must be input A, output A, input B, output B, ... Then set the primitive type to NOT. This will create a part with N inverters.

To make an AND, OR, XOR, or XNOR gate place N+1 pins. The first N pins must be inputs and the last pin an output. Then set the primitive type to the correct logical gate. This creates a single logical gate performing the logical function selected.

Creating a Breakout

A breakout is a special device that allows signals to be associated with a bus. It consists of 1 Bus pin, no internal pins, and N normal pins. Because the device type is breakout the normal pins will be connected to like named signals in the bus.

NOTE: Another way to split signals out of a bus is to use a part with a subcircuit as a splitter, which explicitly routes signals between pins. This has advantage of flexibility since signals do not have to be explicitly broken out but may instead be split into busses or any combination of busses and signals. The disadvantage is that since the splitter is a device it will be listed in hierarchical netlists. In flattened netlists it will not appear if marked as a non-protected device. In addition, there is a significant memory and file size penalty.

Creating a Power (Signal) Connector

A power or signal connector is a special type of device which is generally used to represent a power or ground source, e.g.: +5V, +15V, -15V, -5V, Ground, Vss, Vdd.

These devices have a special properties in the schematic. When a pin on one of these devices is connected to a signal it attempts to assign its pin name to the signal. If the signal doesn't have a name then it gets the name of the pin. If the signal is named, and the name is different from the pin's name, then the user will be prompted to select between the signal name and the pin name.

An additional property of signal connectors is that any signal they are connected to is exported across all pages of the schematic.

The type of the pin must be set correctly if simulations are to make sense. For signal sources like +5V, +15V, and Vss, a normal simulation would expect a logical value of 1 (True). For signal sources like Ground and Vdd, a logical value of 0 (False) is expected. This means the pin type should be "Tied High" or "Tied Low" (See Pin Name Information Window / Pin Information Fields & Buttons / Pin Function Pop-up).

If you don't want the signal connector to supply a signal value, but only its name and the fact that it makes the signal global, then the pin type should be set to "Input".

Creating a Port Connector

Port connectors have the property of associating a signal or bus with a pin on an enclosing part. The association is made by name. The port's Name is compared to the pin names on the parent part. Internal pins in busses are matched by pin name.

See Chapter VII - Hierarchical Design for more information on port connectors.

For a Signal

A port connector for a signal must have a pin which is of the correct type to interface the signal to the parent part's pin. For example, an input pin on the port connector would be correct to connect to a output pin on the parent part. The name of the port connector pin is not important. Only the name assigned to the port connector once it is placed is important; it must be the same as the parents pin.

For a Bus

A port connector for a bus must have a bus pin which contains pins of the correct type. For example, a bus pin with three internal pins A (input), B (input), and C (output) would be correct to connect to a parent part's bus pin which contained pins A (output), B (output), C (input). The name of the port connector bus pin is not important, but the internal pins must have the same name. Once the port connector is placed in a circuit its reference name is important, it must be the same as the parent's pin name.

Creating a Page Connector

Page connectors have the property of exporting the signal their connected to across all pages.

For a Signal

A page connector for a signal contains only input pins. The name of the pins is not important.

For a Bus

A page connector for a bus contains only bus pins with no internal pins. The name of the bus pin is not important.

Adding Packaging Information

A number of DesignWorks features rely on certain attribute fields being present in a part definition. If you create a part symbol with no default attributes it will have the following characteristics:

- The Packager will assume that it has only one unit per package, i.e. a unique package name will be assigned to each device and the Unit field will be left empty.

- If you have not assigned default pin numbers to the pins, they will have to be entered manually for each device on the diagram.
- Part number, component value and package attributes, if needed (e.g. for PCB netlists), will have to be entered manually for each device on the schematic.
- No power and ground connections will be made automatically.
- The default prefix stored in field DevPrefix will be used to auto-generate a device Name when the part is placed.
- The default prefix stored in field PkgPrefix will be used to auto-generate a device InstName when the part is placed.

The following sections describe the attribute entries that should be made in each part created in order to take advantage of these DesignWorks features.

Gate Packaging

For any part that can have multiple units assigned to a single package, the following fields should be set up:

Field Name	**Set In**	**Description**
Unit	Device	This field gives the default Unit setting when a device is placed and can be omitted, if desired. In any case, if Auto Packaging is enabled, it will be overridden by the Unit setting assigned by the Packager.
Unit.List	Device	This field contains a list of the units that can be represented by this symbol. The units are normally specified by a single lower-case letter, although any names up to 16 characters can be used. Items should be separated by commas, e.g. "a,b,c,d".
Unit.All	Device	This field contains a list of all units that can be present in this part type, i.e. including those represented by different symbols. This can be omitted if it is the same as "Unit.List". It will only be different for parts that have multiple functions that can be in the same package.
Part	Device	This field contains the device name which the packager uses to determine if two symbols share the same package. E.g. "74LS00".

VisPin.List	Each pin	This field contains a list of the pin numbers that can apply to this pin. There must be the same number of items in this list and in the same order as the Unit.List field in the device. E.g. "1,4,10,13".

See a more detailed description of the packaging mechanism and fields in Chapter V - Advanced Schematic Editing.

Auto-Naming

Both the Packager and the Auto-Naming features make use of attribute fields in the generation and positioning of names for devices.

NOTE: DesignWorks allows you to select any device field to be used as the prefix for default name generation. The prefix fields described here are the standard ones included in the DesignWorks libraries, but you can also define your own if you have special naming requirements. See the description of the PrefixField design attribute elsewhere in this manual.

Field Name	**Set In**	**Description**
Name.Prefix	Device	This field gives the prefix to be used in generating a device name or package name. In the DesignWorks standard libraries, this field is empty for most integrated circuit parts. They therefore take the default prefix specified with the design. For discretes, the common prefix character is specified.
Name.Spice	Device	This field is specified only for symbols to be used with a SPICE-based analog simulator. It is an optional name prefix matching the standard required for SPICE models.
Name.Pt	Device	This field contains an X,Y coordinate for the default position of the name when the device is placed. This is specified in 1/1000 inch and is relative to the top left corner of the symbol. E.g. "1000, -200". See Chapter V - Advanced Schematic Editing for more options in this field.

Part and Package Fields

The following fields are the standard ones used to specify the part number and package type for netlisting purposes.

NOTE: The standard part and package fields are listed here. You are free to add new fields for special purposes at any time and modify the field names used in the netlist formats.

Field Name	**Set In**	**Description**
Part	Device	This field gives the part name that will be used for netlisting purposes. This may be the same as the name as the part is saved under in the library, or it may be in a format required by some external package.
Part.List	Device	This field contains an optional list of part numbers that can be represented by this symbol. If specified, this list will appear in a value list menu in the device pop-up menu on the schematic.
Part.Pt	Device	This field contains an X,Y coordinate for the default position of the Part field when the device is placed. This is specified in 1/1000 inch and is relative to the top left corner of the symbol. E.g. "1000, -200". See Chapter V - Advanced Schematic Editing for more options in this field.
Package	Device	This field specifies the default package type for use by an external PCB layout package.
Package.List	Device	This field contains an optional list of package codes for the device. IMPORTANT: This field is unusual in that it is linked in DesignWorks to the Part.List field. If an item in Part.List is selected by the user, the corresponding item in Package.List is automatically placed in the Package field.

Other Fields

The following fields are included in the standard DesignWorks libraries although they are not connected with any specific program features.

Field Name	**Set In**	**Description**
Power	Device	This field contains a list of pins to be connected to the power net in netlist reports.
Ground	Device	This field contains a list of pins to be connected to the ground net in netlist reports.
Value	Device	This field contains the default component value for the part. It normally only applies to discrete components.
UnusedPins	Device	This field contains a list of the unused pins in this part type. This field is not included in any of the standard DesignWorks report formats, but is included for general usage.

Chapter XI - Report Generation

Introduction to Report

The DesignWorks Report tool works with the DesignWorks schematic entry tool to provide flexible text report generation capabilities.

Report Features

- Operates from within DesignWorks on any currently open design file.
- Produces the following types of reports, with numerous sorting and format options:
  - Netlists by signal or device
  - Bills of materials
  - Signal and device lists with graphical data
- Produces text files that can be read by any text editor, word processor, or spreadsheet or transferred to external design packages on the Macintosh or other machines.
- The detailed format of the output file is determined by a "form" file, allowing a wide range of output formats to be implemented by the user.
- Works with the DesignWorks "attributes" feature to allow arbitrary text data to be stored with any object and placed in a specified position in an output listing. This can be used for data such as special part numbers, manufacturing data, simulation parameters, etc.
- All types of listings can be sorted and provide counts based on name, attribute field value or other object parameters.
- Includes a number of standard form files for commonly-used formats. These can be used to interface with other design packages, or as samples for developing your own form files.

General Information on Report

The Report tool produces circuit information text files in a wide variety of formats. Essentially any information about a circuit can be placed in an output file.

Output File Formats

Report creates text files which are intended to be incorporated into other documentation or transmitted to other systems.

If it is desired to print any of these files, you will generally get the best results by converting the file to a font such as Courier, which has fixed character spacing. No font information is stored with the files.

Output File Name

The default output file name is the design file name with an ".Rpt" extension added. However, the form file can set any desired file extension and the user can enter any name before the file is saved.

IMPORTANT NOTE: Throughout this chapter we refer to names of design software systems produced by third parties. Many of these names are trademarks and are the property of their respective holders. We can take no responsibility for the accuracy of information in this manual relating to third party systems, as their specifications are subject to change without notice.

Using Standard Report Formats

A number of standard report forms are included with the package for use with popular PCB layout, simulation and other design packages. To use one of these formats, click on the Select Form File button after selecting the Report command in the Tools menu. You will then be prompted for the name of a report form file.If you use the same form file consistently, then it may be specified as the default report form. In the [Report] section of the Ini file add the following line

```
DefaultFormFile=<filename or path\filename>
```

IMPORTANT: Many of the standard formats provided with DesignWorks rely on specific attribute data being entered in the design. Use the Format Notes button described below to get specific details on each form file.

Viewing Format Notes

Any of the standard form files supplied will contain a number of comments explaining the format commands used. Some form files may rely on certain attribute fields being present in devices in the circuit. These notes can be viewed using the Format Notes button in the Report box:

Format Notes

A technical description of some of the standard formats is provided in the Examples section later in this chapter.

Form File Naming Convention

All form file names should have the extension ".RFM" so that they will appear in the Select Form File list. In addition, the following name extensions are used to indicate the hierarchical format:

Name Keyword	**Example**	**Description**
_f	orcad_f.rfm "orcad flat report form"	**Flat** - For use with Flat mode designs or to list the current circuit only of a hierarchical design.
_n	dougl_n.rfm "douglas flattening report form"	**Flatten** - For use only with Physical mode designs. Produces a flattened netlist.
_h	pspc_h.rfm "PSpice hierarchical report form"	**Hierarchy** - For use with Physical or Pure mode designs. Produces a pure hierarchy netlist, i.e. one with separate definitions for nested sub-circuits.

Common Changes to Standard Report Forms

It may be desirable to make minor modifications to one of the standard report forms in order to tailor it to your design needs. This section covers some common changes that can be made without an in-depth understanding of the form file format. Each of these features is covered in more depth in the following sections of this chapter.

Form files can be viewed and edited using a simple utility such as Edit or any standard word processor. If using a word processor, be sure to save the file using a "Text Only" option so that no extra formatting information is included.

Power and Ground Attribute Connections

The Report tool allows you to create common connections in a circuit using attribute entries in devices. This can be used to create power and ground entries in the netlist without having to show all these connections explicitly on the diagram. The standard form files normally contain entries specifying the Power and Ground attribute fields as sources for power and ground pin lists.

Any device attribute field may be specified as a signal source by the form file Command:

```
$SIGSOURCE(signalName) &fieldName
```

- "signalName" is the name of the signal to attach pins to in the netlist.
- "fieldName" is the name of the attribute field to search for in each device to look for pin numbers to attach to the named signal. If &fieldName is omitted, the signal name is taken to be the name of the attribute field to search.

The named device attribute field is assumed to contain a list of pin numbers to attach to the signal. This can consist of a single pin entry, such as "7", or a list, such as "5,6,9,14". For example, most standard report form files contain the following entries:

```
$SIGSOURCE(+5V) &Power
$SIGSOURCE(Ground)
```

There is no fixed limit to the number of SIGSOURCE entries that can be created.

Default File Name

The default output file name is determined by the $CREATEREPORT command in a form file. Following is a typical example:

```
$CREATEREPORT($DESIGNNAME.NET) $PROMPT
```

The contents of the parentheses determines the default name of the output file. The keyword $DESIGNNAME substitutes the name of the current design file. All other characters inside the parentheses are used verbatim. The $PROMPT keyword indicates that a standard file box should be displayed to confirm the file name. If this keyword is not present, the file is saved in the same folder as the design without any prompt.

Attribute Field Usage

Attribute field data is heavily used in generating netlist output files. Most of the standard form files are set up to use the pre-defined attribute fields. These are easily modified to use customized fields which you have added to your design.

Attribute fields are referenced in a form file using the form:

```
&fieldName
```

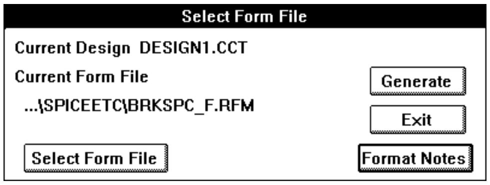
Wherever this format is used, any other field name can be substituted. For example, a typical device listing will appear as follows in a form file:

```
$DEVICES$DEVNAME &Part
```

The "&Part" item will be substituted with the contents of the Part attribute field. Any other field name can be referenced as desired, or additional ones can be added to the line.

Report Operation

The report generator is invoked by selecting the Report item in the Tools menu. The following box will appear:



Select Form File

Current Design DESIGN1.CCT

Current Form File

...\SPICEETC\BRKSPC_F.RFM

If the report generator cannot locate the file "DesignWorks Report Form" then there will be no default form file selected and the "Generate" button will be disabled.

Selecting a Form File

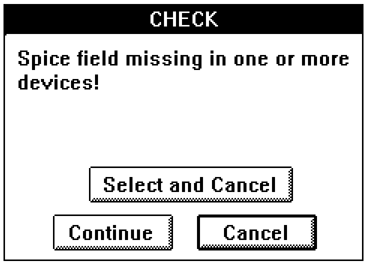
The Select Form File button allows you to choose the format of the report. Clicking this button will bring up a standard file box displaying all text files with names ending in "Form". Select the desired form file by double-clicking on it.

Generating a Report

Once the form file is selected, the "Generate" button will be enabled and you can proceed with report creation by clicking this button. The procedure after that is determined by the form file. Most form files will put up a standard file creation box to allow you to specify the name of the output file.

Circuit Checking

The report form file can contain optional commands to check for errors or missing data in the circuit before proceeding with generating the report. If such a check is encountered and a circuit error is detected, the following type of box will appear, although the text of the error message depends entirely on the form file:



Spice field missing in one or more devices!

This box gives you the option of continuing despite the error, canceling the report, or selecting the offending devices or signals on the circuit diagram, then canceling. Note that the "Select and Cancel" option will only select objects on the current (topmost) circuit page, if there are multiple pages in the circuit.

Device Reporting Options

The listing of devices and device internal circuits can be controlled using the Report Options button in the Get Info command for a device. The following options are available:

Omit from report	Nothing about this device or hierarchical block will appear in the netlist.
Report this device	This device or block will appear in the netlist as if it has no internal circuit.
Report sub-circuit	The internal circuit of this device will appear in the netlist. I.e. in a Pure Hierarchical Netlist, the internal circuit will be defined, in a flattened netlist, the internal circuit will be substituted and this device will not appear at all.

See more information on hierarchical netlists later in this chapter.

Report Errors

Two general types of errors can occur in generating a report:

- form file syntax errors
- circuit data errors

In general, no error messages are issued for incorrect syntax in a report form. Any items in the form file that are not recognized are assumed to be literal text and are transmitted to the output file unmodified.

A number of circuit data errors can occur during report generation, including unnumbered pins or excessive string length on data items such as pin numbers. If any of these errors occurs, a warning box will appear with an explanation of the error.

Chapter XI

Report Form Examples

This chapter provides some detailed examples of report forms. To illustrate the use of various options, most of the following examples will refer to the following sample circuit:

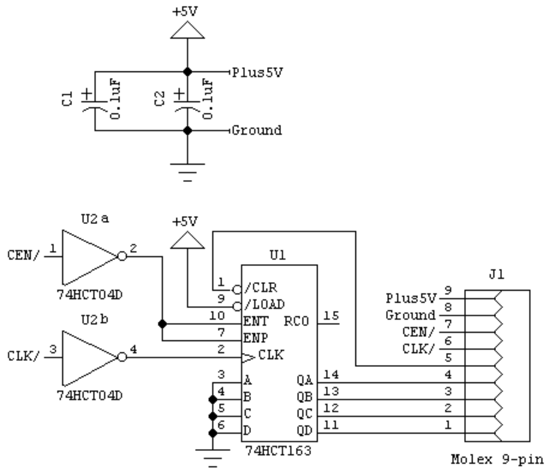


Fig. 1 - Sample Circuit for Report Generation

Note the following characteristics of this circuit:

- several devices have the same name and are intended to be considered as a single device in the reports.
- the connectors labeled "J1" each have a Part attribute entry "Molex 9-pin".
- the inverters labeled "U2" each have a Part attribute entry "74HCT04".
- the capacitors C1 and C2 each have Part attribute entries "Tantalum" and Value attributes "0.1µF".
- the devices U1 and U2 have Power and Ground attributes specifying the standard power and ground pins.

Basic Netlist Formats

A netlist (i.e. a list of the signal connections in a circuit) can take one of two general forms:

- a listing of the signals (or nets) making up the circuit with each signal name followed by a list of the device pins connected to the signal.

- a listing of the devices making up the circuit, with each device followed by a list of the signals attached to each pin. This type is discussed in a later section entitled "SPICE-type netlists".

The first form (which we will refer to as a signal list) is generated by the $SIGNALS listing command, the latter using the $DEVICES command. Thus, just about the simplest form file that will produce a netlist consists of just one line, as follows:

```
$SIGNALS$SIGNAME $PINS
```

The keyword $SIGNALS specifies that a listing of the signals in the circuit is to be produced, but does not in itself specify the format of the list. The text on the balance of the line determines the format of each signal item. In this example, each line will consist of the signal name ($SIGNAME), followed by a blank, followed by a list of the pin connections associated with this signal. The format of each pin item will be in the default format "device-pin#". This can be overridden using the $SIGPINFORMAT definition statement.

The output file generated from this file and the circuit in Fig. 1 will be as follows:

```
unnamed U1-7 U1-10 U2-2
unnamed U1-2 U2-4
D0 J1-1 U1-14
D1 J1-2 U1-13
D2 J1-3 U1-12
D3 J1-4 U1-11
CLR/ J1-5 U1-1
CLK/ J1-6 U2-3
CEN/ J1-7 U2-1
Ground C1-? C2-? J1-8 U1-3 U1-4 U1-5 U1-6
+5V C1-? C2-? J1-9 U1-9
```

Note the following problems in this netlist:

- several unnamed signals are shown with the same label "unnamed".
- discrete component pins (e.g. C1, C2) are listed with a "?" for the pin number, since no pin number was given on the diagram.
- power and ground pins are not appearing.

The following form file addresses these problems and implements some additional features:

```
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$AUTONUMBER(3)
$ASSIGNNAMES $SIGNALS
$SORT $SIGNALS $SIGNAME
$SIGNALS$SIGNAME$COL(15)$PINS
```

This produces the following output file:

```
+5V       C1-2 C2-2 J1-9 U1-9 U1-16 U2-14
CEN/      J1-7 U2-1
CLK/      J1-6 U2-3
CLR/      J1-5 U1-1
D0        J1-1 U1-14
D1        J1-2 U1-13
D2        J1-3 U1-12
D3        J1-4 U1-11
Ground    C1-1 C2-1 J1-8 U1-3 U1-4 U1-5 U1-6 U1-8 U2-7
S00001    U1-7 U1-10 U2-2
S00002    U1-2 U2-4
```

Note the following features of this report:

- the $SORT line causes the nets to be sorted by signal name.
- the $ASSIGNNAMES line causes default names of the form S00001 to be assigned to any signals that are unnamed.
- the $AUTONUMBER format command assigns pin numbers to any device with up to 3 pins.
- the $SIGSOURCE lines indicate that power and ground connections should be extracted from attribute fields of the given names in each device. Connections for the Ground net are extracted from the attribute field with the same name. Connections for the +5V net are extracted from the Power attribute field.
- the $COL keyword is used in the $SIGNALS line to set column alignment and improve readability. Note that this will only appear correctly on the screen if displayed with a fixed-space font such as Courier or Monaco. Another alternative is to put a Tab character in the file at that point, allowing tab settings to be used in a word processor to format the output.

Basic Bills of Materials

The simplest list of the devices making up a circuit is generated by the following single-line form file:

```
$DEVICES$DEVNAME
```

Given the sample circuit in Fig. 1, this will generate the following list:

```
U1
U2
U2
J1
C1
C2
J1
J1
```

Note that there is no particular ordering to this list and like-named devices are not combined. The first problem is fixed by using the $SORT command, as follows:

```
$SORT $DEVICES $DEVNAME
$DEVICES$DEVNAME
```

This generates:

```
C1
C2
J1
J1
J1
U1
U2
U2
```

Like-named devices can be combined into a single report line by inserting the command $COMBDEVSON. The following form also illustrates how attribute parameters can be displayed on each line in fixed columns:

```
$COMBDEVSON
$SORT $DEVICES $DEVNAME
$DEVICES$DEVNAME$COL(20)&Part$COL(40)&Value
```

This produces the following output file:

```
C1        Tantalum     0.1µF
C2        Tantalum     0.1µF
J1        Molex 9-pin
U1        74HCT163
U2        74HCT04
```

IMPORTANT: Note that the COMBDEVSON flag does not actually combine devices with the same name, but rather devices with the same sort position. Since we sorted by device name, this has the desired effect. This can be used to combine lines based on the contents of an attribute field such as the "Part" entry. This is demonstrated in the following form file, which also adds column titles and a use count for each type.

```
$COMBDEVSON
$SORT $DEVICES &Part
Device Type$COL(20)Num. Used$COL(36)Value$COL(50)References
$DEVICES&Part$COL(25)$COUNT$COL(36)&Value$COL(50)$DEVNAME
```

Note that we are now sorting by type, so COMBDEVSON means, in effect, "combine devices with the same ’Part’ field into one line". The $DEVNAME entry here may be ambiguous, since we may be combining devices with different device names onto a single line. If this happens, the report generator will list all values of DEVNAME associated with the "Part" value on that line, as follows:

```
Device Type   Num. Used     Value  References

74HCT04          1                 U2
74HCT163         1                 U1
Molex 9-pin      1                 J1
Tantalum         2          0.1µF  C1 C2
```

IMPORTANT: Since we are sorting by Part value, and not by name, it is important that like-named devices have the same attribute values, otherwise they will appear on different lines.

Spreadsheets and Word Processors

DesignWorks reports can readily be exported to spreadsheet programs such as Microsoft Excel or word processors such as Microsoft Word to take advantage of those programs’ formatting and data analysis capabilities. Typical applications include:

- creating formatted copy for presentations or permanent documentation.
- using spreadsheets to compute pricing, board area, power usage, or other parameters of a design.

Export Format

In general, the important element in exporting data to either type of package is *tabs*. Spreadsheets typically use tab characters to separate fields when importing text data, and word processors can make use of tabs to provide adjustable column formatting. Use of blanks and the $COL operator in a form file should be avoided, since blanks do not have a fixed width in most fonts and will result in inconsistent results.

Inserting Tabs

A tab character can be inserted anywhere in the report form file and will be passed directly to the output file. Consider the following sample component list generator (based on the last example in the previous section):

```
$COMBDEVSON
$SORT $DEVICES &Part
Device TypetabNum. UsedtabValuetabReferences

$DEVICES&Parttab$COUNTtab&Valuetab$DEVNAME
```

Note: The notation tab indicates a tab character entered using the "tab" key on the keyboard.

This format allows us to make use of the tab options in the word processor to improve the look of the report, as follows:

Device Type	**Num. Used**	**Value**	**References**
74HCT04	1		U2
74HCT163	1		U1
Molex 9-pin	1		J1
Tantalum	2	0.1uF	C1 C2

In cases where a single report form command causes multiple items to be placed in a report, such as $PINS, the $ITEMSEPARATOR specification must be used to insert tabs between items in the list. Thus exporting a netlist with tab separators can be done with the following report form (adapted from the preceding section on netlists):

```
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$ITEMSEPARATOR(tab)
$AUTONUMBER(3)
$ASSIGNNAMES $SIGNALS
$SORT $SIGNALS $SIGNAME
$SIGNALS$SIGNAMEtab$PINS
```

Page Separators

When exporting data to a spreadsheet, you will normally not want any page breaks to be inserted in the data since they would confuse the importing program. Page breaks can be disabled by setting the $LINESUSED and $PAGELENGTH format options to large values, as follows:

```
$LINESUSED(32000)
$PAGELENGTH(32000)
```

When exporting to a word processor, you can chose to use either of two methods:

- use the above settings and generate a continuous list, allowing the word processor to determine the page breaks.
- set $PAGELENGTH(0) and $LINESUSED(50) (or whatever number suits your application) to force the report generator to insert a form feed character (page break) and a page header on each page.

Other Report Generation Examples

This section illustrates how the Report Generator’s sorting and formatting features can be used to extract a variety of useful information from a circuit.

Error Checking Reports

The Report generator can be used to check for various kinds of circuit errors such as missing or overlapping names, missing attribute fields, unconnected signals, etc. Making use of the CHECK command, it is possible to create a form file that checks for these errors without even generating an output file.

Missing Attribute Fields

The following report form will check a circuit for any devices missing the field "SUPPLIER" and write out a list of the offending items:

```
$FIND $DEVICES $NOT &SUPPLIER
$SORT $DEVICES $DEVNAME
$DEVICES$DEVNAME
```

By replacing the $DEVICES command with $CHECK you can have it report the errors immediately instead of writing them to a file:

```
$FIND $DEVICES $NOT &SUPPLIER
$SORT $DEVICES $DEVNAME
$CHECK(Devices missing field SUPPLIER!) $DEVICES
```

Unconnected Signals

The following form file will create a listing of all signals with zero or one pin connections on them:

```
$FIND $SIGNALS $NOT $NUMPINS(2)
$SORT $SIGNALS $SIGNAME
$SIGNALS$SIGNAME
```

Using the Report Generator to Assign Names

A report form file can be used to assign names to groups of devices in a circuit without actually generating a report. For example, the following form file assigns a name of the form "Cn" to all capacitors in the circuit:

```
$FIND $DEVICES $TYPENAME(Cap*) $NOT $DEVNAME
$SORT $DEVICES $PAGE $POSY $POSX
$ASSIGNNAMES $DEVICES $FORMAT(C1)
```

We first find all devices with a library type name beginning with the letters "Cap" and not already named. We then sort the devices by page number and position on the drawing. Then we assign the default names.

Giving Priority to Nets

When a netlist is to be used to drive a PCB layout system, it is frequently desirable to bring certain nets to the front of the list to give them higher routing priority. This can be done by inserting an attribute field into the highest priority nets and sorting first on this field. Nets will appear in ascending order of the value in this field. Nets without any field specified will appear last.

In the following example, the field

```
Field     PRIORITY
Value     0
```

has been inserted in the attributes for the net Ground in the sample circuit used in the first sections of this chapter. A reference is made to this field in the $SORT command in the form file, as follows:

```
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$AUTONUMBER(3)
$ASSIGNNAMES $SIGNALS
$SORT $SIGNALS &PRIORITY $SIGNAME
$SIGNALS$SIGNAME$COL(15)$PINS
```

This gives the following results:

```
Ground         C1-1 C2-1 J1-8 U1-3 U1-4 U1-5 U1-6 U1-8
+5V            C1-2 C2-2 J1-9 U1-9 U1-16
CEN/           J1-7 U2-1
CLK/           J1-6 U2-3
CLR/           J1-5 U1-1
D0             J1-1 U1-14
D1             J1-2 U1-13
D2             J1-3 U1-12
D3             J1-4 U1-11
S00001         U1-7 U1-10 U2-2
S00002         U1-2 U2-4
```

Note that the Ground net is now listed before all others and the remaining items are still sorted by name.

Devices by page number

This example uses a sort by attribute fields to make an index to locate types used in a design. This is an extract from a multi-page circuit in which each device has a field called "type" specifying its vendor type number. In addition, discrete components have a field called "Value" with their component value. The following report form is used:

```
$COMBDEVSON
$SORT $DEVICES &Part &Value
TYPEtabVALUEtabPAGES

$DEVICES&Parttab&Valuetab$PAGE
```

This form file produces the following results:

```
TYPE      VALUE         PAGES

74HC04                  1 2 3
74HC193                 1 3
RES 1/4W  10K           1 2
RES 1/4W  20K           1
```

Industry Netlist Formats

In this section we will illustrate how the various Report options can be combined to produce standard netlist formats to interface with outside PCB layout, simulation and other design programs.

IMPORTANT: The following examples are provided as illustrations only and may not include all the format information needed to reliably interface with a given external system. We have no control over the specifications for third-party systems and these may change from time to time. The standard form files provided on disk with DesignWorks may differ from the examples given here. Use a text editor to view these files directly for more complete and up-to-date information.

SPICE-type Netlists

SPICE-based analog simulation programs read all their circuit structure and simulation information from a text file, which must include the following:

- a title line.
- setup commands which set parameters for the simulation, with one command per line. These usually begin with a ".", such as ".tran" or ".ac".
- one line of information per device in the circuit. Each line starts with the device name (the first letter of which determines its type), a list of the nodes attached to that device, then any parameters required for the device.
- an end marker line.

The order of these lines in the file is not significant, except that the first line is the title line, and the last line must be ".end".

A simplified SPICE report form file contains the following commands:

```
$CREATEREPORT($DESIGNNAME.IN)
$ASSIGNNAMES $SIGNALS $FORMAT(S00000)
$DESIGNNAME $DATE $TIME
&Spice
$DEVICES$DEVNAME $PINS &Spice
.end
```

These commands perform the following functions:

$CREATEREPORT($DESIGNNAME.IN) - causes the string ".IN" to be added to the name of the design file to create the default name for the output file.

$ASSIGNNAMES $SIGNALS assigns default names to any unnamed signal nets in the circuit.

$DESIGNNAME $DATE $TIME - create a title line in the output file consisting of the circuit file name and the date and time at which the report was generated.

&Spice - copies the contents of an attribute field called Spice from the design attributes (set up using Set Design Attributes command in the Options menu). This is used to insert SPICE parameter lines before the actual netlist.

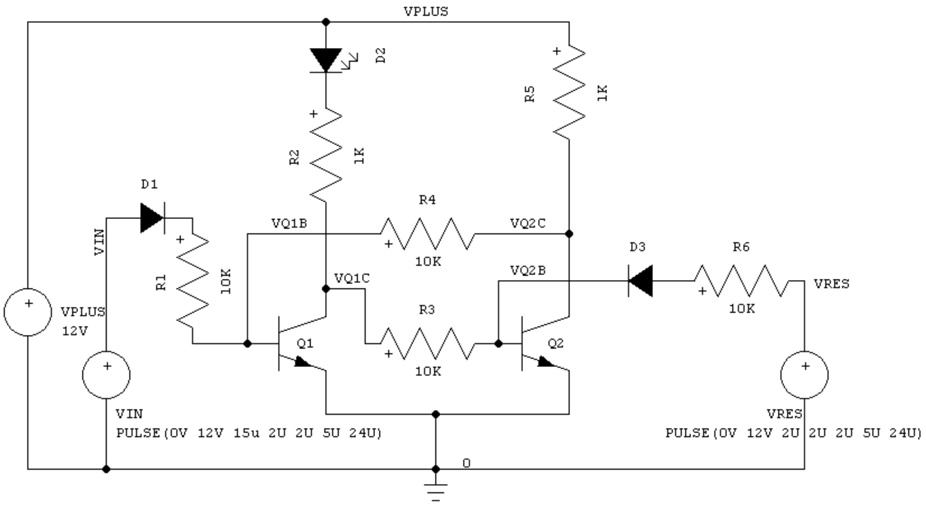
$DEVICES$DEVNAME $PINS &Spice - causes a list of devices to be written into the file. Each line will contain the device name, a list of the attached signals (nodes), followed by the contents of an attribute field called Spice. Note that SPICE uses the first letter of the device name to determine its type. It is up to the user to name each device appropriately.

See Chapter VIII - Before Starting a Major Design for information on automatic name generation for SPICE designs.

The $PINS item lists the pin signals in the order that the device pins were defined in the DevEditor. Therefore it is very important that the device libraries used to create the circuit have a pin order that matches SPICE’s requirements. The SPICE Devices library provided with DesignWorks meets this requirement.

.end - inserts the actual text ".end" in the output file, which is the SPICE termination marker.

Given the following circuit:



the above report form generates the following output file:

```
SPICE Demo  Monday, August 10, 1992  11:05 AM
.TRAN .5US 26US UIC
.model MOD1 npn(bf=80 rb=100 ccs=2pf tf=0.3ns tr=6ns
cje=3pf cjc=2pf
+   va=50)
.model DIODE1 D
Q1  VQ1C  VQ1B  0 MOD1
R1  S00001  VQ1B 10K
R2  VQ1C  VLED 1K
R5  VQ2C  VPLUS 1K
R3  VQ2B  VQ1C 10K
Q2  VQ2C  VQ2B  0 MOD1
D2  VLED  VPLUS DIODE1
D1  VIN  S00001 DIODE1
R4  VQ1B  VQ2C 10K
VPLUS  VPLUS  0 12V
VIN  VIN  0 PULSE(0v 12v 15u 2u 2u 5u 24u)
VRES  VRES  0 PULSE(0v 12v 2u 2u 2u 5u 24u)
R6  S00002  VRES 10K
D3  S00002  VQ2B DIODE1
.END
```

Say, for example, you wished to assign arbitrary names to the devices in your circuit without the restriction of using the first letter to determine the device type. This could be done by placing a field in the attributes for each device containing only the type letter, as in the following example:

```
Field     Value

SPTYPE    V
```

```
Spice     AC 1
```

The form file could then be modified as follows to insert this new parameter in the netlist:

```
$DEVICES&SPTYPE$DEVNAME $PINS &Spice
```

The SPICE name would then consist of the concatenation of the contents of the SPTYPE field and the device name. As an alternative to specifying SPTYPE in every device, this field could be entered into the device library entry for this type.

PCAD Format

A PCAD-format (PDIF) netlist for the sample circuit given at the beginning of the Examples section would look like the following:

```
% Project Report Gen Circuit Sunday, March 11, 1992
{COMPONENT Report Gen Circuit.PCB {ENVIRONMENT Report
Gen Circuit} {DETAIL {SUBCOMP
{I Tantalum C1
{CN 1 Ground 2 +5V
}
}
{I Tantalum C2
{CN 1 Ground 2 +5V
}
}
{I Molex 9-pin J1
{CN 1 D0 2 D1 3 D2 4 D3 5 CLR/ 6 CLK/ 7 CEN/ 8 Ground 9
+5V
}
}
{I 74HCT163 U1
{CN 1 CLR/ 2 SIG3 3 Ground 4 Ground 5 Ground 6 Ground 7
SIG1 8 Ground 9 +5V 10 SIG1 11 D3 12 D2 13 D1 14 D0 15
SIG2 16 +5V
}
}
{I 74HCT04 U2
{CN 1 CEN/ 2 SIG1 3 CLK/ 4 SIG3 7 Ground 14 +5V
}
}
}}}
```

Note that this is a listing by device, not by signal, similar to the SPICE form illustrated previously. Each new device is introduced by the "{I" then the type, name, and list of attached signals follows.

This was generated using the following form file:

```
$CREATEREPORT($DESIGNNAME.PCAD) $PROMPT
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$SIGSOURCE(+12V) &Plus12V
$SIGSOURCE(-12V) &Minus12V
$AUTONUMBER(3)
$COMBDEVSON
$ASSIGNNAMES $DEVICES $FORMAT(U1)
$ASSIGNNAMES $SIGNALS $FORMAT(SIG1)
$SORT $DEVICES $DEVNAME
$DEVPINFORMAT $PINNUM $SIGNAME
% Project $DESIGNNAME $DATE
\{COMPONENT $DESIGNNAME.PCB \{ENVIRONMENT $DESIGNNAME\} \{DETAIL
\{SUBCOMP
$DEVICES\{I &Part $DEVNAME$NEWLINE\{CN $PINS$NEWLINE\}$NEWLINE\}
\}\}\}
```

Note the following important features of this form file:

- the CREATEREPORT command causes the default report file name to be set to the name of the design with ".PCAD" appended. The user will be prompted to verify or change the name before proceeding.
- the SIGSOURCE commands provide the names of power and ground nets that can be specified in device attribute fields. Note that in the above sample netlist, several power and ground connections are shown that do not appear on the diagram. These are derived from device attributes. Every device in the circuit will be searched for an attribute field with any of the given names. If a field is found, the list of pin numbers it contains will be added to the net with the same name. Note that it is possible to use an attribute field with a different name than the net, if needed. See the SIGSOURCE command for details.
- the AUTONUMBER command specifies that devices with 3 or less pins will have pin numbers assigned automatically even if they do not appear on the diagram. This is intended to number discrete components which do not have numbers showing on the diagram.
- the COMBDEVSON command specifies that multiple devices with the same sort value (in this case, the same device name) should be combined into a single device entry.
- the ASSIGNNAMES commands cause default names to be created for any unnamed signals or devices. For devices, the names will be of the form "Un" where "n" is an integer. For signals, the names will be of the form "SIGn". The circuit is searched before assigning names to ensure that the new names do not match any existing ones.
- the SORT command sorts the devices by name, for improved readability.

- the DEVPINFORMAT command sets the format for each device pin in the following $DEVICES listing. In this case, each entry will show the pin number followed by a blank and the signal name.
- the "% Project" text is not a command, so it is transmitted directly to the output file. The $DESIGNNAME command is replaced by the name of the circuit being reported. The $DATE command is replaced by the date the report was created.
- in order to meet the PCAD format requirements, the "COMPONENT" line must start with an opening brace "{". Since this is the Report Generator's comment character, it must be preceded by an escape character "\". The backslash character does not appear in the output. The rest of this line is straight text, except for the $DESIGNNAME commands that are substituted as noted above.
- the DEVICES line is the heart of the report. Everything following the "$DEVICES" keyword on the line will be repeated once for each device listed. In this case, each new device will be introduced by the text "{I " (note the escape character required for each brace), followed by the type name (extracted from an attribute field), the device name, then a carriage return (NEWLINE). The second line of the entry will contain the text "{CN " followed by a list of the pins attached to the device. Each device entry will be terminated by two closing braces, matching the opening ones at the beginning.
- the whole file is terminated by three closing braces.

Draftsman EE Format

The format used by the Draftsman EE printed circuit board layout package is a listing by signal, as follows:

```
NET 1 +5V 1             C1.2
          2             C2.2
          3             J1.9
          4             U1.9
          5             U1.16
          6             U2.14
NET 2 CEN/              1             J1.7
          2             U2.1
NET 3 CLK/              1             J1.6
          2             U2.3
NET 4 CLR/              1             J1.5
          2             U1.1
NET 5 D0  1             J1.1
          2             U1.14
NET 6 D1  1             J1.2
          2             U1.13
NET 7 D2  1             J1.3
          2             U1.12
```

```
NET 8 D3  1            J1.4
          2            U1.11
NET 9 Ground           1            C1.1
          2            C2.1
          3            J1.8
          4            U1.3
          5            U1.4
          6            U1.5
          7            U1.6
          8            U1.8
          9            U2.7
NET 10 SIG1            1            U1.7
          2            U1.10
          3            U2.2
NET 11 SIG2            1            U1.2
          2            U2.4
```

This is generated using the following form file:

```
$CREATEREPORT($DESIGNNAME.DEE) $PROMPT
$ASSIGNNAMES $SIGNALS $FORMAT(SIG0)
$ASSIGNNAMES $DEVICES $FORMAT(DEV0)
$SORT $SIGNALS $SIGNAME
$SORT $DEVICES $DEVNAME
$CONTSTART(tab)
$MAXITEMSPERLINE(1)
$ITEMSEPARATOR(tab)
$SIGPINFORMAT $PINSEQ(1)tab$DEVNAME.$PINNUM
$SIGNALSNET $SIGSEQ(1) $BUSNAME(.)$SIGNAMEtab$PINS
```

Note the following unique features of this form file:

- the ASSIGNNAMES commands will generate names with the given default format.
- only one pin entry is placed on each line and each line starts with a tab character.
- all nets and pin entries must be numbered sequentially. This is done using the $PINSEQ and $SIGSEQ sub-commands with a given number starting point.
- the BUSNAME sub-command is used to prefix signal names with the name of the bus containing them, if any.

Cadnetix Format

The following form file generates a netlist appropriate for a Cadnetix PCB layout system:

```
$CREATEREPORT($DESIGNNAME.Cadnetix) $PROMPT
$ROWS(60)
$COLUMNS(80)
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$AUTONUMBER(3)
$BLANKREPLACE(_)
$COMBDEVSON
$PINFORMAT $DEVNAME $PINNUM
$CONTEND( $)
$ASSIGNNAMES $SIGNALS
$ASSIGNNAMES $DEVICES
$SORT $SIGNALS $SIGNAME
COPYRIGHT  1993 &Company
DESIGNERS: &Designers
DATE:      $DATE
FILE:      $DESIGNNAME.Cadnetix
PARTS LIST
ID &ProjectID
$SORT $DEVICES &Part
$DEVICES&Part &Package $DEVNAME
EOS
$MAXITEMSPERLINE(8)
$SORT $DEVICES $DEVNAME
NET LIST
$SIGNALSNN $SIGNAME $PINS
EOS
```

Note the following unique features of this format:

- the BLANKREPLACE command is used to ensure that blank characters do not appear in any names in the file.
- the CONTEND command sets the string " $" to be placed at the end of any data line that is being continued to the next line because the LINEWIDTH or MAXITEMSPERLINE settings were exceeded.
- the ASSIGNNAMES commands assign default names to any objects that were unnamed on the schematic. The default format of "D0001" for devices and "S0001" for signals will be used since no format is specified.
- a number of design attribute fields are referred to, including &Company, &Designers, and &ProjectID. These must be specified in the Set Design Attributes command or they will not appear in the report file.
- the MAXITEMSPERLINE command is used to specify how many pin entries can appear on a single line before a continuation line is created.

- the Report Generator stops scanning for a keyword once it recognizes something. For example the line "$SIGNALSNN" is interpreted as the keyword "$SIGNALS" followed by the literal text "NN". This does not apply to attribute field names, since these are not predefined.   They must be terminated by a blank or special character.

Report Form Language Reference

This section of the manual provides detailed information on the creation of report form files. It is not necessary to read this section in order to use the standard form files provided. In order to get a quick overview of the capabilities available, it is suggested that you review the Examples section first.

Overview

The form file required to create a report consists of a series of commands, variables and text, referred to as the Form Language. A form file can be created using any text editor or word processor. If you use a word processor, be sure to select a "Text Only" option when saving the file.

A number of standard form files are provided with DesignWorks and many of these are explained in the Examples section following. These can be used as guides in creating your own form files.

DesignWorks Object Types

For report generation purposes, a DesignWorks circuit consists of two types of objects: devices and signals. The Report tool has two general sets of options allowing information on each of these types of objects to be written to a file in a variety of forms.

Devices

The device listing features are used to create component lists, bills of materials and types of netlists that are listed by device, such as a SPICE or PCAD format. For each device in a list, any of the following data can be included:

- Device Name - this is the name applied to the device instance, e.g. "U1", "R23", "J12", etc.
- Device Attributes - the contents of any device attribute fields can be included in a listing.
- Device Token - the device token is an integer that is assigned automatically by DesignWorks when the device is first placed on the diagram and not changed as long as that device exists in that circuit. This token number can be used by an external program to determine which devices are new and which are old in a circuit, despite name or parameter changes.

- Device Sequence Number - this is a temporary integer that is assigned by the Report tool as part of its sorting process and may change during a report if multiple sorts are done. This can be used in cases where external programs require objects to be sequentially numbered.
- Device Type information - this refers to information retrieved from the library when the device was created, such as the type name (e.g. "74LS123"), number and types of pins, etc.
- Device Graphical Information - information about the device’s page number, position and orientation on the schematic are available to allow transfer of schematic data to other systems, or to create reference listings.
- Attached signals - a device listing can also contain a list of the signals attached to each pin of the device. Each signal entry can reference any of the signal data described below. See the $DEVPINFORMAT command.

Device listings are created using the $DEVICES command, described later in this chapter.

Signals

Signal listings are used primarily for creating netlists for transfer to external simulators or PCB systems. The following information is available about each signal in the circuit:

- Signal Name - this is the name of the signal as it appears on the schematic, or is assigned automatically by the report generator.
- Signal Attributes - the contents of any signal attribute field can be included in a signal listing.
- Signal Token - this is an integer that is assigned automatically by DesignWorks when the signal is first created and not changed as long as that signal exists in that circuit. Note however, that many common editing operations result in signals being merged or broken apart, which will often result in the signal token changing even though some of the connections in the old signal remain. This token number can be used by an external program to determine which signals are new and which are old in a circuit, despite name or parameter changes.
- Signal Sequence Number - this is a temporary integer that is assigned by the Report tool as part of its sorting process and may change during a report if multiple sorts are done. This can be used in cases where external programs require objects to be sequentially numbered.
- Signal Graphical Data - information about page numbers and coordinates of attached lines on the schematic can be included in a signals listing.

- Attached device pins - any signal list can contain a list of the device pins interconnected by this signal. Pin format options allow almost any of the device parameters mentioned above to be included in each pin entry. See the $SIGPINFORMAT command.

Signals listings are created using the $SIGNALS command described later in this chapter.

Form File Structure

When the Report command is selected by the user, the form file is opened and read sequentially. Any text that is not recognized as a valid command, variable or comment is written directly to the output file unchanged. When a command or variable is found, text generated by the system is substituted for the command or variable text. In some cases, such as when a signal or device list is called for, the line containing the command will be repeated once for each item in the list.

Any number of listing commands can be given in one file. Thus, a single output file can contain a number of netlists and component lists in a variety of formats, if needed.

The form file contains four types of data:

1) **Commands**: commands start with the "$" symbol and are in capital letters. Commands are used to control the format of the report. Some of the commands need parameters to set certain values (e.g. the number of characters on a line). Any parameters are enclosed in parentheses following the command word. If the desired output file format requires that a word starting with a "$" be placed in the form file, it is best to precede the "$" with the escape character "\" (backslash) to ensure that it will not be accidentally interpreted as a command.

Commands fall into the following general sub-groups:

- definition commands are those which set parameters for how data is generated but do not themselves generate any text in the output file. An example is the $SIGPINFORMAT command, which specifies how pins will be displayed in a signal list.
- listing commands are those which cause lists of items to be directed to the output file, for example the $SIGNALS command, which generates a list of the signals in the circuit. All the commands which follow a listing command on a line are item-specific, i.e. their exact meaning depends upon the type of item being listed.
- item-specific commands are those which can appear after a listing command and whose meaning depends upon the type of item being listed. For example, the $EVENTS command can appear in a $SIGNALS listing

only and causes a list of timing data for that signal to be produced. The $EVENTS keyword has no meaning in any other part of the file.

• system variables are commands that cause a specific internal value to be directed to the output file, e.g. the circuit file name or the system date.

2) **Attribute variables**: Attribute variables start with the "&" symbol and are used for handling the user-defined attribute fields in devices, signals, or the circuit itself. The variable will be replaced with the actual value of that field (found in each item). For example &Pwr might appear as 140mW in the report for one of the devices in the circuit.

If the desired output file format requires that a "&" be placed in the form file, it is best to precede the "&" with the escape character "\" (backslash) to ensure that it will not be accidentally interpreted as an attribute variable.

3) **Comments**: comments are messages enclosed between the left and right braces "{" and "}". Comments are intended for internal documentation of the command file. No characters between or including the braces, or a carriage return following the closing brace, will be transmitted to the report. If you need to transmit the character "{" to the output file, you must precede it by the "escape" character "\" (backslash) so that it will not be recognized as a comment.

4) **Data:** All other text and special characters (e.g. tabs and carriage returns) found in the command file are considered data and are transmitted to the output file unmodified. Any item starting with a $ or & but not recognized as a command or variable word will be transmitted to the report file. No error messages are generated for incorrect keywords.

Definition Commands

The definition commands control the general format of the report, for example:

1) the number of rows per page.
2) the number of columns per row.
3) the specification of an attribute field as a signal source (e.g. for ground or power connections).
4) other report formatting options

Each of these parameters is set by using specific command words, as indicated below.

Page Formatting Commands

$LINEWIDTH(n)

The LINEWIDTH command specifies the maximum number of characters that will be placed on a line before an automatic carriage return is inserted. See the MAXITEMSPERLINE command below, which can be used to specify the length as an item count instead of a character count. The CONTEND and CONTSTART commands can be used to indicate how continuation lines are implemented.

The default value for LINEWIDTH is 255.

$PAGELENGTH(n)

The PAGELENGTH command is used to specify the action to be taken when a page is full (i.e. when the number of lines specified in LINESUSED is exceeded). If PAGELENGTH is set to 0, then a Form Feed character (ASCII code 12) is placed in the file. This character is accepted by many word processors as a page break. If PAGELENGTH is set to any other integer, then that number is taken to be the total number of lines from the top of one page to the top of the next. To disable all page breaks, set PAGELENGTH and LINESUSED to very large values (e.g. 32,767). The default value is 66.

$LINESUSED(n)

The LINESUSED command specifies number of lines to use on each report page before moving to the next page. The actual length of the page is specified by PAGELENGTH above. The default value is 32767, i.e. page breaks are disabled.

$MAXITEMSPERLINE(n)

The MAXITEMSPERLINE command is used to restrict the number of entries placed on a line in repeating structures such as a netlist. The default value is 32,767.

Common Signal Connections

The "signal source" facility allows you to name certain signals to be treated as power and ground nets. This has the following two effects:

- It allows you to create common connections in a circuit using attribute entries in devices. This can be used to create power and ground entries in the netlist without having to show all these connections explicitly on the diagram.
- It informs the Report generator that the given signals should be merged across hierarchy levels.

See more information on power and ground nets in hierarchy later in this chapter.

Any device attribute field may be specified as a signal source by the Command:

```
$SIGSOURCE(signalName) &fieldName
```

"signalName" is the name of the signal to attach pins to in the netlist. "fieldName" is the name of the attribute field to search for in each device to look for pin numbers to attach to the named signal. If &fieldName is omitted, the signal name is taken to be the name of the attribute field to search.

The named device attribute field is assumed to contain a list of pin numbers to attach to the signal. This can consist of a single pin entry, such as "7", or a list, such as "5,6,9,14".

For example "Ground" connections can be created as follows:

- Using the Get Info command on a selected device, create an entry such as "7" in the attributes for a device, in this case connecting pin 7 to the Ground net.
- Place the command $SIGSOURCE(Ground) in the form file. This causes Report to search all device attributes for fields named Ground and use the field value as a pin number.

There is no fixed limit to the number of SIGSOURCE entries that can be created.

NOTE: The predefined fields Ground and Power are normally used for standard power and ground connections. These pin connections are prespecified for all digital components in the standard DesignWorks libraries. Corresponding $SIGSOURCE statements are included, where appropriate, in all standard Report form files. You can create your own special-purpose power nets by using the Define Attribute Fields command to define a new field, then adding the appropriate SIGSOURCE statement to the form file.

Auto-numbering Discrete Components

Automatic numbering of device pins can be specified by the Command:

```
$AUTONUMBER(number of pins)
```

Any device with less than or equal to the number of pins specified will have pin numbers automatically assigned if none were present in the circuit file. This option is intended to provide pin numbers for discrete components in a circuit, since they do not normally have pin numbers on a diagram. Accordingly, a typical use of this command would be AUTONUMBER(3). The default number is zero.

Setting Pin Order in Device Lists

Some design programs, such as SPICE and SILOS, require netlist formats in which the device pin order is significant. Since there are no standards for pin order, different packages will likely require different ordering schemes. In order to allow a variety of pin orderings to be stored with each device type, the Report tool can be asked to search a device attribute field for pin order information. The name of this field is specified in the DEVPINSEQUENCE command, as follows:

```
$DEVPINSEQUENCE &attrField
```

Once this field has been specified, if a $PINS sub-command is encountered in a $DEVICES listing, each device is searched for an attribute field with the given name. If the field is found (in the device instance or the type attributes), then it is assumed to contain a list of names of the pins to write out to the file. The pin names must be listed exactly as they appear in the DevEditor’s pin list, or, in the case of SIGSOURCE pins, as the fields are named in the device attributes. If the specified field is not found in a given device, then the normal pin ordering is used.

For example, assume a 2-input gate has pins named A, B and OUT. A pin sequence attribute field could be specified in the device (or in the library type attributes) as follows:

```
Field     SPICEPINS
Value     OUT,A,B
```

The Command:

```
$DEVPINSEQUENCE &PinSequence
```

in the form file would then cause subsequent listings of that device to have pins appear in the specified order. The predefined field PinSequence is intended for this purpose.

Setting Report Hierarchy Mode

The command $HIERARCHY is used to enable hierarchical netlisting and set the type of processing that will be done in generating a file from a hierarchical design. If this command does not appear in a file, then the default mode will be CIRCUIT, i.e. the contents of the current circuit level only will be listed. Example:

```
$HIERARCHY $FLAT
```

The HIERARCHY command must appear before any ASSIGNNAMES, FIND, SORT, DEVICES or SIGNALS commands to ensure that the internal devices are included in these operations.

See more information in the section on Hierarchical Net Listing below and in the Report Examples section.

Format Options

$COMBPINSON/$COMBPINSOFF

When ON causes multiple pin connections on the same device to appear, for example, as:

```
IC1-2,5,6,12
```

instead of:

```
IC1-2  IC1-5  IC1-6  IC1-12
```

Default is OFF.

$ALIGNCOLSON/$ALIGNCOLSOFF

When ON causes extra blanks to be inserted between netlist or component list entries to force column alignment. This improves readability when used with a fixed-space font such as Courier. If ALIGNCOLS is OFF, then the item separator will be a tab if TABFIELDS is ON, or a single blank otherwise. Default is OFF. NOTE: You may have to insert a number of blanks at the front of continuation lines using CONTSTART to force correct alignment of the first item in each continuation line.

$SPACE(number)

The SPACE item sets the column spacing used when ALIGNCOLSON is set. Default is 16. E.g:

```
$SPACE(32)
```

sets the column spacing to 32.

$ITEMSEPARATOR(string)

The ITEMSEPARATOR command specifies a string that will be inserted between successive items in an item list. If ITEMSEPARATOR is not specified, then item spacing will be determined by TABFIELDS and ALIGNCOLS. If it is specified, it will override TABFIELDS and ALIGNCOLS. Example:

```
$ITEMSEPARATOR(,)
```

This example will insert a comma between items in a list.

$UNNAMEDDEVS(string)

The UNNAMEDDEVS command specifies a string that will be output whenever a device name is called for and the device in question is unnamed. For example:

```
$UNNAMEDDEVS(?????)
```

$UNNAMEDSIGS(string)

The UNNAMEDSIGS command specifies a string that will be output whenever a signal name is called for and the signal in question is unnamed. The default value is "unnamed". For example:

```
$UNNAMEDSIGS(?????)
```

$ONEPINSON/$ONEPINSOFF

When ON suppresses the pin number from being printed in netlist entries for devices with only one pin. This will cause entries such as test points to appear as TP1 instead of TP1-1. Default is OFF.

$CONTSTART(string)

The CONTSTART item specifies a string of characters that will be inserted at the beginning of each line that is a continuation from a previous line (i.e. whenever MAXITEMSPERLINE or LINEWIDTH is exceeded while writing a list). Examples:

```
$CONTSTART(            )
$CONTSTART($ )
```

The first example will insert a number of blanks in front of each continuation line to set column alignment. The second example inserts a $ sign and a blank to mark continuation lines.

$CONTEND(string)

The CONTEND item specifies a string of characters that will be inserted at the end of each line that is being continued onto the next line (i.e. whenever MAXITEMSPERLINE or LINEWIDTH is exceeded). Example:

```
$CONTEND( >)
```

This example will insert a blank and a > sign at the end of a line when an item list has overflowed the line.

$BLANKREPLACE(string)

The BLANKREPLACE item specifies a string of characters that will be substituted for a blank in any device or signal name. This is done to accommodate systems which cannot accept blanks in names. Examples:

```
$BLANKREPLACE()
$BLANKREPLACE(_)
```

The first example will remove any blanks from names. The second example will replace each blank in a name with an underscore character.

$UNCONNPINSON/$UNCONNPINSOFF

When ON, allows device pins that have no signal lines attached to them and are not connected to any other signals by name to appear in the netlist, otherwise they are suppressed. Default is ON.

$REPEATITEMON/$REPEATITEMOFF

This option, when ON, forces the last item in any list to be repeated at the start of any continuation line. This is used by some systems to recognize continuations.

$CONTREPEATON/$CONTREPEATOFF

This option, when ON, forces the first part (up to the start of the first pin item) of the first line in a signal list to be repeated at the start of any continuation line. This is used by some systems to recognize continuations by repeating the signal name or other parameters on each successive netlist line for the same signal.

$ATTRNUM(number) &attrField

The ATTRNUM command provides a way of placing sequential numbers for devices or signals in a report, while allowing the numbers for specific objects to be set by an attribute field. For example, some netlist formats require that power and ground nets be numbered with specific values, then all other nets sequentially numbered with higher values.

The ATTRNUM command specifies the name of an attribute field to search for in each object. If it is found, the field value is placed in the report, otherwise a sequential number is substituted. The attribute field must also appear somewhere in the relevant $DEVICES or $SIGNALS command line. The substituted number is generated in the same format as it is given in the ATTRNUM command, and starting with the value given. I.e. leading zeros and other text are retained.

For example:

```
$ATTRNUM(00100) &NETNUM
$SIGNALS&NETNUM $PINS
```

In this case, if a field called "NETNUM" exists in a signal, the contents of that field will be used. Otherwise, a number will be generated having 5 digits and starting at 100 for the first one generated.

File Control Commands

Output File Creation

The command $CREATEREPORT is used to set the report file name and prompt the user to change it if desired. The format of this command is as follows:

```
$CREATEREPORT(reportName) $PROMPT
```

"reportName" can consist of any mix of random text and the variable "$DESIGNNAME" (which inserts the name of the circuit being reported) or an attribute field name &field (which inserts the contents of the specified design attribute field.

The $PROMPT option specifies that the user should be prompted with a standard file creation box with the given name as a default. Otherwise the file is created immediately with no user interaction. If a file already exists by the given name, it is overwritten.

The following example creates a file with the name of the circuit plus the text " RPT" attached and prompts the user to verify or change the name.

```
$CREATEREPORT($DESIGNNAME RPT) $PROMPT
```

The following example creates a report named the same as the circuit plus ".IN" and does not prompt the user for confirmation.

```
$CREATEREPORT($DESIGNNAME.IN)
```

The following example creates a file called exactly "Output File" with no user confirmation.

```
$CREATEREPORT(Output File)
```

The following looks for a field in the design attributes called "SPICEFILE" and uses the contents of that as the report file name, with no user confirmation.

```
$CREATEREPORT(&SPICEFILE)
```

IMPORTANT: If the CREATEREPORT command appears in the form file after some command that generates output text, then the user will be prompted to create a file with the circuit name plus " RPT" and the $CREATEREPORT will be ignored. The best way to ensure correct operation is to put this command at the very beginning of the file. If it does not appear to be working, check for blank lines or spurious characters at the beginning of the file.

$CLOSEREPORT

The CLOSEREPORT command causes the current report output file to be closed. This can be followed by another CREATEREPORT command to create another output file, thus allowing a single form file to generate multiple report files. The output file is normally closed automatically at the end of the form file, so this command is only necessary to create multiple output files. If CLOSEREPORT is not followed by a CREATEREPORT command, and subsequent listing commands attempt to write output data, then the user will be prompted to provide a name for a new output file.

Pin Format Commands

The following format commands specify the way individual pins are presented in reports. Each format command must be followed on the line by a combination of format sub-commands (described below) and text which described the appearance of each item entry. All leading and trailing blanks in the format description are ignored. Any text not recognized as a sub-command is placed in the file as part of each item entry.

$PINTYPEFORMAT string1 string2 ...

This command sets the output string used for each pin type wherever $PINTYPE is specified in a device listing. The strings are given in the following order, separated by blanks:

Item order	**Function**	**Default string**
1	Input	INPUT
2	Output	OUTPUT
3	3-state	3STATE
4	Bidirectional	BIDIR
5	Not used	?????
6	Open Collector	OC
7	Undefined	?????
8	Bus	BUS
9	Perm. low	LOW
10	Perm. high	HIGH
11	Latched input	LTCHIN
12	Latched output	LTCHOUT
13	Clocked input	CLKIN
14	Clocked output	CLKOUT
15	Clock input	CLOCK
16	Open Emitter	OE
17	No connection	NC

Note: “Undefined” can only occur if a device was being created in the DevEditor and was closed before all its pins had been specified. If less than 17 strings are specified, the trailing items retain their default value.

For example, to set the pin type string for inputs and outputs (leaving all others at their default values):

```
$PINTYPEFORMAT Inp Out
```

$DEVPINFORMAT/$SIGPINFORMAT

This command sets the format for pin lists appearing in a $DEVICES listing or $SIGNALS listing, respectively. For $DEVICES, the default format is "$SIGNAME", meaning the signal name alone will appear for each device pin. For $SIGNALS, the default format is "$DEVNAME-$PINNUM", giving a pin item such as "U1-12".

The following format keywords are valid after $DEVPINFORMAT or $SIGPINFORMAT:

$CHILDSIGNAME	The name of the signal attached to this pin in the device's internal circuit, or null string if none. This is intended for creating sub-circuit headers for formats requiring the names of the internal signals attached.
$DEVHIERNAME	The hierarchical name (i.e. prefixed by names of parent devices) of the device.
$DEVINSTNAME	The contents of the InstName field on the device.
$DEVNAME	The name of the current device.
$DEVSEQ(n)	The sequence number of the current device, i.e. an integer assigned to the device, starting at "n" for the first device in the sorted sequence and incremented for each subsequent one. This number is not permanently associated with the device but it strictly its sequence number in the current sort. The "(n)" part is optional. If it is omitted the sequence starts at zero. If no sort has been done, all devices will have sequence number "n". NOTE: DEVSEQ will be zero if no SORT has been done.
$DEVTOKEN	The token number permanently assigned to the device.
$GRID	The page reference grid position (e.g. C3) of the current item.
$NEWLINE	Inserts a line break in the pin entry.

$PAGE	The page number of the current item.
$PINDIR	A two-character code representing the orientation of the current pin. **Output String** **Direction** E1 East N1 North W1 West S1 South
$PINNAME	The name of the current pin (i.e. as it appears in the DevEditor’s pin list window).
$PINNUM	The visible (user-applied) number of the current pin.
$PINSEQ(n)	The sequence number of the current pin, i.e. a sequence number starting at "n" for all pins listed. This can be followed by an optional number in parentheses representing the starting point for numbering. If it is omitted, zero is assumed.
$PINTYPE	A string representing the type of the current pin. The string used for each pin type can be set using the $PINTYPEFORMAT command, described above.
$POSX	The X coordinate of the current pin in 1/1000".
$POSY	The Y coordinate of the current pin in 1/1000".
$SIGHIERNAME	The hierarchical name (i.e. prefixed by the names of parent devices) of the current signal.
$SIGINSTNAME	The contents of the signal’s InstName field.
$SIGNAME	The name of the current signal.
$SIGSEQ	The sequence number of the current signal, i.e. an integer assigned to the signal, starting at "n" for the first signal in the sorted sequence and incremented for each subsequent one. This number is not permanently associated with the signal but it strictly its sequence number in the current sort. The "(n)" part is optional. If it is omitted the sequence starts at zero. If no sort has been done, all signals will have sequence number "n".
$SIGTOKEN	The token number assigned to the signal.

$SYSPIN	The system pin number of the current pin, i.e. its ordinal number as it appears in the pin list in the DevEditor, starting at 1.
&attrField	The contents of the given attribute field on the current pin.

Format Notes Section

The $NOTES keyword introduces a special comments area that can be viewed by the user by clicking on the Format Notes button in the Report dialog.

NOTE: We recommend including a $NOTES section in every form file you create indicating the purpose (i.e. output format) of the form, the expected hierarchy mode of the design, and all attribute fields used.

The notes section starts with the $NOTES keyword on a line and ends with a $END keyword on a line. All text on the intervening lines will be ignored when generating a report but will be displayed in the Format Notes box on the screen. Following is an example:

```
$NOTES
This format generates a "flattened" netlist file for
PSpice from a hierarchical or flat design.

Non-standard attribute fields required:

Field name             Where        Usage

Spice     Devices      SPICE parameters

$END
```

Page Header Format

The Header section controls what will appear at the top of each page of the report. This section starts with a $HEADER keyword and ends with $ENDHEADER. Note that the contents of this section is not passed to the output file immediately, but only when a page feed occurs. The Header section command words and their results are given below:

$PAGE - Puts out the current page number of the report being generated.

$DATE - Puts out the current day, month and year.

$TIME - Puts out the current system time.

$DESIGNNAME - Puts out the name of the circuit file being used as input for the report.

$ENDHEADER - Marks the end of the header section.

&attr - if an attribute variable appears anywhere in the header section, then the program attempts to match the variable name with a field in the design attributes. If a match is found, the value of the field is substituted. This is intended to accommodate global items such as titles, circuit parameters, routing instructions or other info that is not related to any one device or signal.

Example page header specification:

```
$HEADER
XYZ Industries of Smithville, Inc.
File: $DESIGNNAME      Date: $DATE
$ENDHEADER
```

Assigning Default Names

$ASSIGNNAMES type $FORMAT(pattern) &field $TYPENAME $REASSIGN
$ASSIGNINSTNAMES type $FORMAT(pattern) &field $TYPENAME $REASSIGN

The ASSIGNNAMES command is used to apply names to unnamed devices or signals in the circuit before a listing is generated. If this is not done, any object which has not been named (either manually or by the auto-naming or packaging features) will appear as "unnamed" in a net or component list.

However, in some cases, especially hierarchical designs, it is likely that duplicate names exists due to duplication of hierarchical blocks. For this reason, the $REASSIGN keyword may be added. When it appears anywhere after an ASSIGNNAMES or ASSIGNINSTNAMES, all names that were auto-assigned by either the Schematic or the Report tool and have not been edited by the user will be reassigned a unique name.

$ASSIGNINSTNAMES works exactly like $ASSIGNNAMES except that it sets the InstName attribute field instead of the Name field. This is used for creating netlists from designs created in Physical Hierarchy mode.

IMPORTANT NOTES:

1) The names assigned by ASSIGN(INST)NAMES will be invisible but permanently associated with the object. They can be made visible using the Attributes command associated with the object.

2) If the report hierarchy mode has been set to $FLAT or $FLATDOWN and the design is in Physical hierarchy mode, then the ASSIGNNAMES command sets the InstName field, not Name.

3) ASSIGN(INST)NAMES reassigns all temporary names that were assigned by any previous report.

Note that ASSIGNNAMES can be used in conjunction with the FIND command to assign names to selected subsets of devices, such as all resistors, etc. See the Examples chapter for more information.

The "type" specification (either "$DEVICES" or "$SIGNALS") is mandatory, all other items are optional.

The $TYPENAME option is valid only for devices and specifies that the device type name should be used as part of the name with a numeric part appended.

The $FORMAT item specifies a default format for the assigned names, which can be overridden by the &field item, as explained below. The pattern specified in parentheses after $FORMAT contains a non-numeric part followed by a numeric part (specified as zero digits). The non-numeric part is copied directly to each default name generated. The numeric part is converted to an integer, which is incremented for each name generated. Leading zeroes can be specified by including more than one zero in the numeric part. Example formats:

```
$FORMAT(IC0)
```

will give IC1, IC2, IC3, etc.

```
$FORMAT(U000)
```

will give U001, U002, U003, etc.

If $FORMAT is not specified, format D00000 will be assumed for devices, and S00000 for signals.

The &field item allows you to specify an attribute field to search for a string to use as a name prefix. All discrete component libraries supplied with DesignWorks contain a user field called Name.Prefix which specifies the commonly-used letter for the device type, such as "R" for resistors, "C" for capacitors, etc. Thus, if the item &Name.Prefix appears in the ASSIGNNAMES command, resistors will be numbered R1, R2, etc., and capacitors C1, C2, etc.

As with other user field searches, the Report Generator searches in a number of places to find a prefix. For devices the device instance is searched first, then the data in the library entry. This allows you to specify defaults for values which can be overridden for specific cases.

Regardless of how the format is specified, the circuit is searched before any name assignments are done and the non-numeric and numeric parts of each object name are extracted. The numeric parts of any assigned names are generated starting at the next integer higher than any in the circuit.

Examples:

```
$ASSIGNNAMES $DEVICES $FORMAT(U0)
```

This will assign names starting at U1 (or the first "Un" name that is unused). The names will be removed after the report has been generated.

```
$ASSIGNNAMES $SIGNALS
```

This will assign signal names to unnamed signals starting with S00001. The names will be removed from the circuit after the report has been generated.

```
$ASSIGNNAMES $DEVICES &Function
```

This will assign names using the string found in the field Function in each device, or names such as D00001 if such a field is not found or is empty. The assigned names will be kept permanently with the circuit.

Selection and Sorting Commands

The SORT Command

The SORT command provides the ability to sort device and signal listings on any data field. Items with the same value in any field can be optionally merged into a single line. This allows listings to be organized to suit various applications, for example:

- device or signal lists can be sorted by name to enhance readability.
- device lists can be sorted by type so that each line lists one device type with all information common to that type and a list of instances of the type.
- a device list can be sorted by page number to show the devices used on each page.
- signals can be sorted by an attribute field, for example to give priority to certain nets so they are listed first for autorouting purposes.
- devices can be sorted by any attribute field, for example by component value, stock number, cost, etc.

The SORT command has the following form:

```
$SORT objectType field1 field2 ...
```

"objectType" must be either "$DEVICES" or "$SIGNALS".

"field1", "field2", etc. are identifiers indicating which fields to sort on. The list is sorted first on the first field. If any items in the list have identical values in that field, then the groups of like-valued items are sorted on the next specified field, etc.

Once a sort has been done, it remains in effect for all subsequent listings on that object type until the next SORT command. Each SORT clears the previous SORT. Sorting can be in ascending or descending order, as described below.

The following field specifications are valid in all SORT commands:

$ASCENDING	This specifies that all following fields specified should be sorted in ascending order. This is the default direction, so ASCENDING will normally only be required after a DESCENDING to reset the sort direction for subsequent fields on the same line.
$DESCENDING	This specifies that all following fields will be used to sort in descending order.

There are two different methods which may be used when comparing elements in a sort. The default method, which matches all previous DesignWorks versions, is $NUMERIC. In this mode, any numeric portion of a text string is extracted and treated as an integer. For example, this means that U10 will appear after U9 in a component list. The disadvantage of this method is that U010 and U10 will be treated as the same name, or component values .001 and .01 will be combined (because "001" and "01" have the same integer value). The $LEXICAL option performs a purely lexical sort, i.e. by ASCII code. E.g. U1, U11, U100, U2, U22, U3, etc. This is intended when sorting by component values or other random data, but will not normally give the desired results for names.

The following example shows the usage of this keyword.

```
$SORT $DEVICES &Part $LEXICAL &Value
```

This example will sort devices by Part field using the default numerical method, then sort any devices with the same Part value by the string in the Value field using the lexical method.

$LEXICAL	Performs a purely lexical sort. E.g. U1, U11, U100, U2, U22, U3, etc. Intended when sorting by component values or other random data.
$NUMERIC	Default method. Numeric portion of a text string is extracted and treated as an integer. E.g. U1, U2, U3, U11, U22, U100.

If "objectType" is $DEVICES, the following field specifications can be used:

$DEVNAME	Sort on the device name. If no name exists in the circuit, and no name has been assigned using ASSIGNNAMES, then the item will be placed at the end of the list. If the device name contains a numeric portion (e.g. U12), then the numeric part of the name is sorted on its integer value, instead of its character value. E.g. the name U12 will appear before U110, instead of after, as it would in a purely lexical sort.
$DEVHIERNAME	Sort on the device hierarchical name, i.e. a name generated by prefixing the device name with the names of all parent devices in the hierarchy. This is normally only used in flattened netlists.
$DEVINSTNAME	Sort on the contents of the InstName field. This is normally only used in flattened netlists.
$POSX	Sort on the horizontal position of the top left corner of the device symbol on the diagram.
$POSY	Sort on the vertical position of the top left corner of the device symbol on the diagram.
$TYPENAME	Sort on the device's type name, i.e. as it appears in the Parts palette.
$ORIENT	Sort on the device orientation.
$PAGE	Sort on the page number that the device appears on.
$NUMPINS	Sort on the number of device pins.
$DEPTH	Sort on the contents of the Depth attribute field. NOTE: This field will only be valid if the $HIERARCHY $PURE mode is enabled.
&attrField	Sort on the contents of the attribute field named attrField in each device. If that field is empty or does not exist in a given device, then that device will be placed at the end of the list.

If "objectType" is $SIGNALS, the following field specifications can be used:

$SIGNAME	Sort on the signal name. If no name exists in the circuit, and no name has been assigned using ASSIGNNAMES, then the name "unnamed" will be assumed. See the note under "DEVNAME" above regarding sorting numeric portions of the name.
$SIGHIERNAME	Sort on the signal hierarchical name, i.e. the name of the signal prefixed by the names of all parent devices in the hierarchy. This is normally only used in flattened netlists.
$NUMPINS	Sort on the number of device pins attached to this signal.
$SIGSOURCE	Sort on the order of appearance of this signal in a SIGSOURCE command. If the signal does not appear in a SIGSOURCE command (as most won't), it will appear after the SIGSOURCE signals. This is used to place all power and ground nets at the front of a netlist.
&attrField	Sort on the contents of the attribute field named "attrField" in each signal. If that field is empty or does not exist in a given signal, then that signal will be placed at the end of the list.

The combination options COMBSIGS and COMBDEVS instruct the report generator to combine items with the same sort value into a single line in the listing. This can be used to produce a variety of useful listing formats, for example listings of all devices having a certain type name.

IMPORTANT: The "COMB" options will have no effect if no sort has been specified since they only operate on sorted values.

See the examples at the end of this manual for more information on how to use the sorting capability.

$COMBSIGSON/$COMBSIGSOFF

When ON causes signals with the same sort value to be combined into a single net in the netlist. Default is OFF.

NOTE: COMBSIGSON should not be used in most normal netlisting applications. Any like-named signals have already been logically connected by DesignWorks and will be listed in one entry even if COMBSIGS is OFF. COMBSIGSON will have the additional effect of merging like-named signals, even if they appeared on different pages of the schematic.

$COMBDEVSON/$COMBDEVSOFF

When ON causes devices with the same sort value to be merged into a single entry in the component list. Default is OFF.

The FIND Command

The FIND command provides the ability to select subsets of the circuit data for reporting purposes. This can be used to produce the following types of reports:

- listings containing only objects that are selected on the circuit diagram.
- netlist formats that require different formats for different types of devices or signals. For example, some formats require a list of the I/O ports of a circuit separately from other signals.
- error checking listings. For example, all devices missing a certain attribute field or all signals with less than two pin connections.

The FIND command has the following form:

```
$FIND objectType field1 field2 ...
```

"objectType" must be either "$DEVICES" or "$SIGNALS".

"field1", "field2", etc. are identifiers indicating which fields to find on. After the FIND command is complete, only objects matching all the field specifications will be left in the list. If no field items are specified, all object will be found. This can be used to clear a previous FIND.

Once a FIND has been done, it remains in effect for all subsequent listings on that object type, or until another FIND is done.

IMPORTANT NOTES:

1) If a FIND command is used, it should appear before any other command that refers to the signal or devices lists, such as $DEVICES, $SIGNALS, $SORT, $CHECK and $ASSIGNNAMES. If this is not done, then these other commands will have operated on objects that are not included in the final list, which may give confused results.

2) A FIND command always creates a new list of objects from the design, regardless of previous FIND commands.

The following field options are valid for devices or signals:

$NOT	This inverts the meaning of all field specifications following on this FIND command. I.e. find all objects that do not match the given specification.
$SELECTED	This selects only objects that are selected (i.e. highlighted) in the circuit diagram.
&attrField(pattern)	Select on the contents of the attribute field named attrField in each signal or device. If the pattern is not specified, the object will be found if the given field exists, regardless of contents (even if empty). If the pattern is specified, then the contents of the field must match the pattern exactly. See below for information on patterns.

If "objectType" is $DEVICES, the following field specifications can be used:

$DEVNAME(pattern)	Select on the device name. An optional pattern is specified in parentheses after the keyword (see below for more information on name patterns). If the pattern is not specified, then all named devices are selected and all unnamed devices are excluded. If the pattern is specified then the name must match the pattern exactly to be selected.
$TYPENAME(pattern)	Select on the name of the device’s type, i.e. the name that appears in the Parts palette. If the pattern is not specified, this item will be ignored (i.e. all devices will be selected).

If "objectType" is $SIGNALS, the following field specifications can be used:

$SIGNAME(pattern)	Select on the signal name. An optional pattern is specified in parentheses after the keyword (see below for more information on string patterns). If the pattern is not specified, then all named signals are selected and all unnamed signals are excluded. If the pattern is specified then the name must match the pattern exactly to be selected.
$NUMPINS(min)	Select signals only with greater than or equal to the number of pins specified.
$SIGSOURCE	Select only signals that appear in a SIGSOURCE command.

String Patterns

The SIGNAME, DEVNAME, TYPENAME and attribute field specifications allow an optional pattern to be given in a manner similar to the FIND command in DesignWorks. All objects with a field value matching the pattern will be selected. A number of "wildcard" characters can be used to specify ranges of field values, as follows:

- "*" matches any string of zero or more characters. For example "D*" will match "D", "D0", "D12", "DogGone", etc. "*" by itself matches anything, although the SIGNAME and DEVNAME specifications will not select objects that are unnamed.
- "#" matches any string of 1 or more digits. E.g. "XA#" will match "XA0", "XA123", etc.
- "?" matches any single character. For example, "CTRL?" will match "CTRLX", "CTRL/", etc.

The wildcard characters can be used in combination, as in "DATA#??" which will match "DATA123", "DATA1SQ", etc.

The CHECK Command

The CHECK command is used to detect missing or incorrect data in a circuit file before a report is generated. It has two distinct functions:

- It can be used in combination with the FIND command to locate any devices or signals meeting some error condition (e.g. missing an attribute field). It causes a message to be displayed on the screen if there were any objects of the given type selected by the last FIND command. Thus, errors are detected by creating a FIND command that selects only objects with the problem in question, then following this with a CHECK command to display the error message.

• It can be used to check that the hierarchy mode of the current design is appropriate for use with this form file. In this form, the $CHECK command is followed by one or more keywords indicating the modes that are incorrect. If the mode of the current design is any of these, the message box is displayed.

IMPORTANT: In a correct circuit, the FIND command used for checking purposes will result in no objects being selected for listing. Therefore you must follow the CHECK command with another FIND otherwise no report will be generated.

The format of the CHECK command is one of the following:

```
$CHECK(message) objectType
$CHECK(message) modes
```

• "objectType" is either "$DEVICES" or "$SIGNALS".

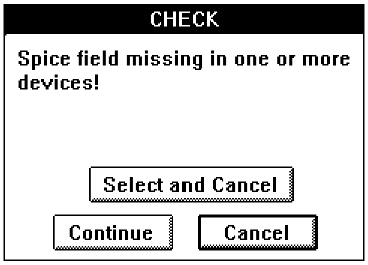
• "modes" is one or more of "$PURE", "$PHYSICAL", or "$FLAT".

• "message" is an optional text message to be displayed if an object of the given type is found on the list. If the message is not specified, a standard message will be displayed.

For example, the following sequence will display a message if the attribute field "SPICE" is missing from any devices:

```
$FIND $DEVICES $NOT &Spice
$CHECK(SPICE field missing in one or more devices!) $DEVICES
$FIND $DEVICES { Re-find all devices }
```

This will cause the following box to be displayed if the error condition is detected:



CHECK

Spice field missing in one or more devices!

The "Select and Cancel" option gives the user the opportunity to have the offending devices or signals highlighted on the circuit diagram.

Listing Commands

The $DEVICES and $SIGNALS commands are the heart of the form file in that they cause the actual listing to be generated. The format of the listing is determined by the format specification that follows the command on the line and by the various options that have been discussed above.

Note the following general rules regarding these commands:

- in a basic listing, one line is generated for each item in the list, but this will be affected by the following:
  - if the COMBDEVS or COMBSIGS flags are on, multiple items with the same sort value will appear on one line.
  - the $NEWLINE format command can force multiple lines to be generated for each item.
  - limits on line length imposed by the $LINEWIDTH and $MAXITEMSPERLINE commands may cause multiple lines to be generated.
- if the COMBDEVS or COMBSIGS flags are used to merge multiple items into one line, some of the format commands may be ambiguous. For example, if a device list is sorted by type name and COMBDEVS is ON, each line may refer to multiple devices. In this case, a format command such as $DEVNAME will output the names of all devices covered by each line. This can be used to create a bill of materials which lists each device type, followed by the names of all instances of that type.
- the format of the listing for each device or signal is completely determined by the text and commands which follow the $DEVICES or $SIGNALS keyword. There is no default format, so if no line format is specified, a sequence of blank lines will be written to the file.
- DesignWorks "Pseudo-devices" such as page connectors, signal connectors and bus breakouts are never included in any listing.
- any characters appearing after $DEVICES or $SIGNALS that is not part of a format sub-command will be placed verbatim in each line of the listing. For example, blanks, tabs or commas can be used to format each line.

Listing by Signal

$SIGNALS lineFormat

The SIGNALS command creates a listing of the signals in the circuit, along with any related data such as signal names, signal attribute data and the names of the devices and pins that a signal is attached to. The format of each line is completely specified by the data following the $SIGNALS keyword on the line.

Sub-Commands to be used after SIGNALS

$SIGNAME - writes out the name of the signal.

$SIGSEQ(n) - writes the sort sequence number of the signal. These numbers are assigned in order, starting with the number "n", when the signals are sorted. If "n" is not specified, then numbering will start with zero. NOTE: If no SORT has been done, all signals will have sequence number "n".

$SIGTOKEN - writes the signal token number that was assigned by DesignWorks when the signal was first created. This number will be unchanged for the life of the signal in this circuit.

$BUSNAME(*separator*) - if the signal being listed is contained in a bus, then the name of the bus is inserted, followed by the separator text. If the signal is not contained in a bus, then nothing is inserted.

$COUNT - outputs a count of the signals with the same sort value that have been combined on this line. If COMBSIGSOFF has been set, this will always be 1.

$COUNTINST - works exactly like $COUNT except that it counts the number of signals with different InstName values, instead of Name values.

$PINS$PINS - list all of the pins attached to the signal by their device names and pin numbers. The format of each pin item is determined by the $SIGPINFORMAT command.

$SINGLE - tells the report generator to write only a single value for all items following on the line, regardless of how many items were combined with the same sort value.

$PAGE - lists all circuit pages that this signal appears on.

$COL(N) - If the current position in a line is less than N spaces form the left hand end then blanks are inserted until the Nth column is reached. For instance $COL(20) will force any output to be indented to the 20th column of each line.

$NEWLINE - Writes a carriage return character into the file.

&attr - attributes in ReportsignalsWrites the contents of an attribute field of the given name. For example a field called WIDTH could be defined for signals using the Define Attribute Fields command. The Report Generator will extract information from these fields by using the command &WIDTH which could be used to print out the width of each trace to be passed to a PCB layout package.

Examples:

```
$SIGNALS$SIGNAME $PINS
```

This line will produce a basic netlist, with each line consisting of the signal name, a blank and a list of the device pins (device name and pin number) attached to this signal, as in:

```
CLOCK U1-12 U6-19 U7-12
```

The default pin format is shown above. This can be changed by the SIGPINFORMAT command, as follows:

```
$SIGPINFORMAT $PINNUM$DEVNAME
$SIGNALS#$SIGSEQ $PINS
```

will produce a netlist of the following form:

```
#00001 12U1 19U6 12U7
```

See the more detailed examples later in this manual.

Listing by Device

```
$DEVICES lineFormat
```

The $DEVICES command generates a listing of the devices and their related data, such as the device name, the device type and any device attributes. "lineFormat" consists of any combination of random text and the following sub-commands.

$DEVNAME - the name of the device.

$DEVSEQ(n) - the sequence number of the current device, i.e. an integer assigned to the device, starting at "n" for the first device in the sorted sequence and incremented for each subsequent one. This number is not permanently associated with the device but it strictly its sequence number in the current sort. The "(n)" part is optional. If it is omitted the sequence starts at zero. NOTE: If no sort has been done, all devices will have sequence number "n".

$DEVTOKEN - writes the device token number that was assigned by DesignWorks when the device was first created. This number will be unchanged for the life of the device in this circuit.

$TYPENAME - the device’s type name. I.E. the name used to refer to this type in the library.

$COUNT - the number of devices being merged to form this line, i.e. the number having the same sort value.

$COUNTINST - works exactly like $COUNT except that it counts the number of devices with different InstName values, instead of Name values.

$POSX - the X coordinate of the top left point on the device symbol in 1/1000 in.

$POSY - the Y coordinate of the top left point on the device symbol in 1/1000 in.

$SINGLE - tells the report generator to write only a single value for all items following on the line, regardless of how many items were combined with the same sort value.

$ORIENT - the orientation of the device symbol. One of the following direction codes will be output.

Output String	**Direction**
E1	East
E2	East flipped
N1	North
N2	North flipped
W1	West
W2	West flipped
S1	South
S2	South flipped

$NUMPINS - the number of pins on the device.

$NUMINPS - the number of inputs on the device. This number is valid only on primitive devices.

$NUMOUTS - the number of outputs (i.e. anything other than an input) on the device. This number is valid only on primitive devices.

$PAGE - the circuit page that this device appears on.

$GRID - the page reference grid position.

$PINS - a list of the device’s pins. The format of each item is determined by the $DEVPINFORMAT command. If a DEVPINFORMAT command is not specified, the default pin format shows only the name of the attached signal. The order in which the pins are written out is determined by the following rules (in descending order of priority):

- if a $DEVPINSEQUENCE command has appeared previously in the form file, and the specified attribute field exists in the device, then the pins are written in the specified order.

- if a $PINNUM appears in the DEVPINFORMAT, the pins are sorted in order of the this pin number.
- if neither of the above applies, then the pins are written in the order they were created, i.e. the order they appear in the DevEditor's pin list.

$COL(N) - If the current position in a line is less than N spaces form the left hand end then blanks are inserted until the Nth column is reached. For instance $COL(20) will force any output to be indented to the 20th column of each line.

$NEWLINE - Writes a new line character into the file. This allows a device entry to occupy several lines in the output report.

&attr - Inserts the value of the attribute field specified.

Report searches in a number of places to find device attribute fields, in the following order:

- the attributes for this device
- the attributes for all devices with the same sort value (if COMBDEVSON has been set)
- the attributes for the device type (i.e. as set in the library using the DevEditor)

One result of the above search order is that default attribute values for devices can be specified in the library using the DevEditor. These can then be overridden if necessary by specifying the same field in each instance.

Other Listing Commands

The following items can appear in the body of the form to provide counts of the objects listed in the report. They are only valid after a listing has been generated with the given object type since the count is made as the listing is generated.

$SIGCOUNT	The number of items in the last signals listing. If COMBSIGSON has been set, this will mean in effect the number of different sort values in the list.
$PINCOUNT	After a netlist, this will be a count of the total number of device pins listed.
$DEVCOUNT	The number of items in the last devices listing. If COMBDEVSON has been set, this will mean in effect the number of different sort values in the list.

$CHANGECOUNT	The number of changes (i.e. editing operations) performed on the circuit since it was created. This number can be used to check for matching versions of a circuit file.
$NEWPAGE	The NEWPAGE command causes a page feed to occur. The exact data written out by NEWPAGE depends upon the LINESUSED and PAGELENGTH settings.
$DATE	Writes out the current day, month and year.
$TIME	Writes out the current system time.
$DESIGNNAME	Writes out the name of the circuit file being used as input for the report.
&attr	If an attribute variable appears anywhere in the body of the form file (i.e. not after a $SIGNALS or $DEVICES keyword) then the program attempts to match the variable name with a design attribute field. If a match is found, the value of the field is substituted. This is intended to accommodate global items such as titles, circuit parameters, routing instructions or other information that is not related to any one device or signal.

Hierarchical Net Listing

The Report Generator has the ability to produce a hierarchical netlist of a circuit, i.e. a list including the internals of hierarchical blocks. The listing can proceed into nested blocks to any desired depth to provide the required detail. For the following discussions, this facility is referred to as “hierarchical netlisting”. Without hierarchical netlisting enabled, only objects in the current circuit level (i.e. that displayed in the topmost window) will appear in a netlist.

> *See the hierarchical netlist example in the Report Form Examples section of this chapter for more information.*

Types of Hierarchical Netlists

Two types of netlists can be produced from a hierarchical design:

- Flattened Netlist: A flattened netlist is generated by substituting each hierarchical block symbol with its internal circuit. All information about the hierarchical blocks themselves, nesting levels, etc. is lost. The resulting output appears to have been generated from a flat design. This format is most commonly used for printed circuit board netlists.

- Pure Hierarchical Netlist: A Pure Hierarchical Netlist consists of a netlist of the master circuit for the design, and then separate netlists of the internals of each type of hierarchical block used. Each internal circuit can be written into a separate file, or all netlists may be concatenated in a single file in either top-down or bottom-up order. This format is commonly used for simulators (e.g. SPICE) and FPGA tools.

IMPORTANT NOTE: Since a Pure Hierarchical Netlist contains only a single copy of the definition of each internal circuit, it cannot contain any instance data.

See Chapter VII - Hierarchical Design for more information on instance vs. definition data.

The $HIERARCHY Command

The $HIERARCHY command in a form file sets the type of netlist output to be generated.

NOTE: The hierarchy mode of the netlist does not have to match that of the design. For example, you can generate a PURE netlist from a PHYSICAL mode design. The following table indicates which design modes are appropriate for the various netlist types.

Report Mode	**Design Mode**	**Description**
$CIRCUIT	FLAT PHYSICAL PURE	Generate a normal netlist for the single current circuit only. This is the default mode.
$TOPCIRCUIT	FLAT PHYSICAL PURE	Generate a normal netlist for the master circuit of the current design only.
$FLAT	FLAT[1] PHYSICAL	Generate a flattened netlist for the entire design.
$FLATDOWN	FLAT[1] PHYSICAL	Generate a flattened netlist for the current circuit and all nested blocks.
$PURE	FLAT[2] PHYSICAL[3] PURE	Generate a pure netlist for the entire design. In particular, this causes a $DEVICES listing to list each unique type of device only once.
$PUREDOWN	FLAT[2] PHYSICAL[3] PURE	Generate a pure netlist for the current circuit and all nested blocks.

Notes:

1) A flattened netlist will normally make reference to instance attributes in the design. Instance attributes are not normally used in a Flat mode design, so this combination is not recommended.

2) PURE netlist mode can be used with a FLAT design but no internal circuits will be listed.

3) In PURE netlist mode, the internal circuit for each block type is listed only once, regardless of how many times it is used in the design. Therefore, instance data is not normally included in a PURE netlist.

Restricting Reporting of Internal Circuits

The listing of internal circuits of individual blocks can be controlled using the Report Options button in the Get Info command for a device. The option "Report this device" will cause this device or block to appear in the netlist as if it has no internal circuit. The option "Report sub-circuit" will cause the internal circuit of this device will appear in the netlist. I.e. in a PURE netlist, the internal circuit will be defined, in a FLATTENED netlist, the internal circuit will be substituted and this device will not appear at all.

Listing Format for Internal Circuits

For flattened netlists, the output format is usually the same as would be generated from a flat design, except that hierarchical or instance names (see next section) are sometimes substituted for definition names.

For Pure Hierarchical Netlists, formatting is more complex since the format of the master circuit is usually different from the internal circuit definitions. For this reason, the Report tool treats these two types of circuits as two separate format definitions.

The general outline of a Pure Hierarchical Netlist format is as follows:

```
{
  $PURE mode lists each device type once
  so the next $DEVICES listing will only
  list unique type definitions.
}
$HIERARCHY $PURE
{
  The DEFINECIRCUIT section gives the listing
  format for the internal circuits
}
$DEFINECIRCUIT
{ Listing commands for internal circuit }
$END
{
  Use FIND command to locate all devices that
  have a non-empty Depth attribute, i.e.
  they have an internal circuit.  The Depth
  attribute field contains an integer indicating
  the number of circuit levels below this device
}
$FIND $DEVICES &Depth
{
  Sort by the Depth field to list from the
  bottom up.  The $DESCENDING option could
  be used to sort from the top down.
}
$SORT $DEVICES &Depth
{
  The following $DEVICES listing will list
  each unique type of device only once because
  we are in $PURE mode.  $INTERNAL causes the
  internal circuit to be output for each device
  in the format defined in $DEFINECIRCUIT.
  Other text or commands could be included in
  the $DEVICES line to output a circuit block
  header or terminator.
}
$DEVICES$INTERNAL
{
  Now switch to TOPCIRCUIT mode and do a
  separate listing for the master circuit
}
$HIERARCHY $TOPCIRCUIT
{ Commands to list master circuit }
```

Depth Ordering in Pure Netlists

The Depth attribute field is used to select device types that have internal circuits and to sort by depth for top-down or bottom-up listings.

NOTE: The Depth field is not maintained by the Schematic module during normal editing operations. The Depth value is calculated by the Report tool only when a PURE netlist is generated.

The Depth field will be empty for devices with no internal circuit or those with the "Report This Device" option selected in the device Report Options. For devices with an accessible internal circuit, the Depth field will contain an integer indicating the number circuit levels below the device. E.g. if the device's internal circuit contains only bottom-level devices, the value will be 1.

The Depth field can therefore be used with the $FIND command to perform separate listing operations on devices with or without internal circuits. For example:

```
$FIND $DEVICES &Depth
```

will select all devices having an internal circuit, whereas:

```
$FIND $DEVICES $NOT &Depth
```

will select devices having no internal circuit. Similarly, the $SORT command can be used to determine whether internal circuits are defined from the lowest level up or vice-versa.

```
$SORT $DEVICES &Depth
```

will cause the lowest level devices to be listed first, i.e. bottom up. This is the preferred order for most formats because it ensures that each type of internal circuit is defined before it is referred to in any other circuit.

Instance vs. Definition vs. Hierarchical Names

Devices and signals have three different types of "names" that are significant in producing netlists from hierarchical designs. These are summarized in the following table.

Report Keyword	**Attribute Field**	**Description**
$DEVNAME $SIGNAME	Name	Normally used in Flat mode designs. Not guaranteed to be unique across a hierarchical design.
$DEVINSTNAME $SIGINSTNAME	InstName	Not normally used in Flat mode designs. InstName is used for package assignment in Physical mode designs. Should be unique across design.
$DEVHIERNAME $SIGHIERNAME	(temporary generated by Report)	A generated name consisting of the device or signal Name field prefixed by the names of all containing hierarchical blocks.

Note the following rules when selecting which type of name to use in a report format:

- $DEVNAME/$SIGNAME should not be used in a flattened netlist unless you intend to manually guarantee that it is unique across the entire design and each type of hierarchical block is only used once.
- In Physical mode designs, $DEVINSTNAME should be used as the device "name" in netlists intended for PCB-layout use, since it normally contains the package assignment.
- In flattened netlists, $SIGHIERNAME should be used for the signal "name" since it ensures a unique name that is easily located in the schematic. $SIGINSTNAME can be used but must be assigned manually to each signal.

Power and Ground Connections in Hierarchy

Power and ground symbols (i.e. Signal Connector devices) do not make an immediate logical connection across hierarchy levels. For this reason, power and ground connections in different circuit levels will be considered separate nets in a flattened netlist, unless you do one of the following:

- Specify the name of each power and ground net in a $SIGSOURCE statement in the form file. This instructs the Report tool to look for those named nets and integrate them into a single entity.
- Create individual ports for the power and ground connections on each hierarchical block symbol and treat them as regular nets.

Report Command Summary

This section lists the keywords available in the various parts of a form file.

Definition Commands

```
$ALIGNCOLSON/$ALIGNCOLSOFF
$ATTRNUM(number) &attrField
$AUTONUMBER(number of pins)
$BLANKREPLACE(string)
$CLOSEREPORT
$COMBPINSON/$COMBPINSOFF
$CONTEND(string)
$CONTREPEATON/$CONTREPEATOFF
$CONTSTART(string)
$CREATEREPORT(reportName) $PROMPT
$DEFINECIRCUIT/$END
$DEVPINSEQUENCE &attrField
$HIERARCHY
   $FLATDOWN $FLAT $CIRCUIT $TOPCIRCUIT $PUREDOWN $PURE
$HIERNAMESEPARATOR
$ITEMSEPARATOR(string)
$LINESUSED(n)
$LINEWIDTH(n)
$MAXITEMSPERLINE(n)
$NOTES/$END
$ONEPINSON/$ONEPINSOFF
$PAGELENGTH(n)
$REPEATITEMON/$REPEATITEMOFF
$SIGSOURCE(signalName) &fieldName
$SPACE(number)
$UNCONNPINSON/$UNCONNPINSOFF
$UNNAMEDDEVS(string)
$UNNAMEDSIGS(string)
```

Item Format Commands

```
$DEVPINFORMAT/$SIGPINFORMAT

   $CHILDSIGNAME
   $DEVHIERNAME
   $DEVINSTNAME
   $DEVNAME
   $DEVSEQ
   $DEVTOKEN
   $GRID
   $NEWLINE
   $PAGE
   $PINDIR
   $PINNAME
   $PINNUM
   $PINSEQ
   $PINTYPE
   $POSX
   $POSY
   $SIGHIERNAME
   $SIGINSTNAME
   $SIGNAME
   $SIGNUM
   $SIGSEQ
   $SIGTOKEN
   $SYSPIN

$PINTYPEFORMAT string1 string2 ... (17 items)
```

Page Header Format

```
$PAGE
$DATE
$TIME
$DESIGNNAME
$ENDHEADER
&attr
```

Selection and Sorting Commands

```
$ASSIGNNAMES type $FORMAT(pattern) &field
          $TYPENAME

$SORT objectType field1 field2 ...

$COMBDEVSON/$COMBDEVSOFF
$COMBSIGSON/$COMBSIGSOFF

$FIND objectType field1 field2 ...

$CHECK(message) objectType
```

Listing Commands

```
$SIGNALS format

   $BUSNAME(separator)
   $COL(N)
   $COUNT
   $NEWLINE
   $PAGE
   $PINCOUNT
   $PINS
   $SIGHIERNAME
   $SIGINSTNAME
   $SIGNAME
   $SIGSEQ(n)
   $SIGTOKEN
   $SINGLE
   &attr

$DEVICES format

   $COL(N)
   $COUNT
   $DEVHIERNAME
   $DEVINSTNAME
   $DEVNAME
   $DEVSEQ(n)
   $DEVTOKEN
   $GRID
   $INTERNAL
   $NEWLINE
   $NUMINPS
   $NUMOUTS
   $NUMPINS
   $ORIENT
   $PAGE
   $PINCOUNT(&seqField)
   $PINS
   $POSX
   $POSY
   $PRIMNAME
   $SINGLE
   $TYPENAME
   &attr

$CHANGECOUNT
$DATE
$DESIGNNAME
$DEVCOUNT
$NEWPAGE
$PINCOUNT
$SIGCOUNT
$TIME
&attr
```

Chapter XII - Find and ErrorFind Tools

Introduction

Find and ErrorFind are tools used to locate devices, signals or pins in a schematic. Once the objects are found they may be sequentially selected or selected as a group.

Find searches by name or attribute value whereas ErrorFind searches for possible error conditions. After errors are found they may be selected and marked as OK or corrected.

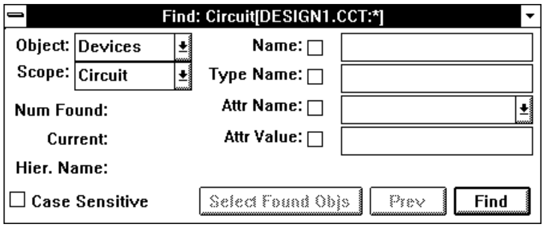
Installation

Find and ErrorFind are code modules that work with DesignWorks using the MEDA interface. To locate the modules when DesignWorks starts up, they must be in the same folder with the DesignWorks program itself, or in the folder specified in the TOOLFOLDER statement in the DesignWorks setup file, usually called Tools.

Starting Find or ErrorFind

To invoke Find or ErrorFind, select the name in the Tools menu. This opens a find palette window.

The Find Window



Find: Circuit[DESIGN1.CCT:*]

Object: Devices

Scope: Circuit

Search Scope

Search for Popup A popup list that displays the name of the type of object to be found. When clicked on, the popup list allows the user to select one of the following choices Devices, Signals, Pins, Selected Devices, Selected Signals, or Selected Pins.

Search Scope Popup A popup list that displays the name of the scope to search for objects. When clicked on, the popup list allows the user to select one of the following choices.

Design	All unlocked circuits in the design are searched.
Circuit	Only the current circuit is searched.
Here/Down	All unlocked circuits from the current circuit and down in the hierarchy are searched.
Page	Only the current page is searched.

Search Criteria

Criteria Switches

These checkboxes indicate if their corresponding criteria will be used when trying to find a match. The check box will be automatically selected if a value is entered in the corresponding field.

Match Criteria

Text entered into these fields is compared to the values in the objects examined to see if they match. The following "wildcard" characters may be used:

*	Any string of zero or more characters
?	Exactly one character
#	Any string of one or more decimal digits

Attr Field Popup

A popup list that displays the name of the attribute field to be matched. When clicked on, the popup menu allows the user to select one of the attribute fields valid for the type of object being searched for.

Information Fields

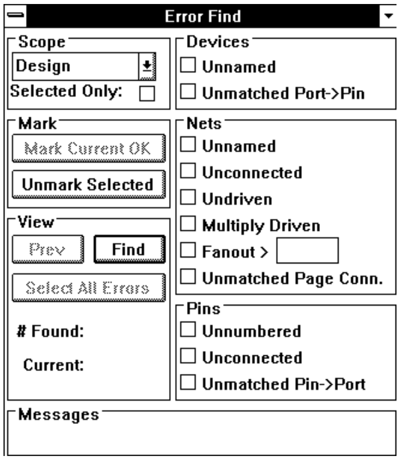
These fields may not be edited by the user but instead display results back to the user. The top field displays the total number of objects found that matched the criteria. The next field indicates which of the found objects is currently selected.

The bottom field displays the objects hierarchical name, i.e.. the object’s name prefixed by its position in the hierarchy.

Find Controls

There are three buttons. The first "Select Found Objs" selects all of the objects found. The second button "Prev" moves backwards in the list of found objects and selects the object before the current one. The last button "Find" / "Next" has two operations. Initially, when the button is labeled "Find", it is used to initiate the search. Once the objects have been found then its label changes to "Next" and when pressed it advances the selection to the next found object. You can return to the "Find" by changing any of the search criteria or by making a structural change to the circuit.

The ErrorFind Window



Search Scope

Search Scope Popup

A popup list that displays the scope of the search for errors. When clicked on, the popup list allows the user to select one of the following choices.

Design	All unlocked circuits in the design are searched.
Circuit	Only the current circuit is searched.
Here/Down	All unlocked circuits from the current circuit and down in the hierarchy are searched.
Page	Only the current page is searched.

Selected Only Option

If the Selected Only box is checked, only objects selected on the schematic will be checked for errors.

Device Errors

Two types of device errors may be searched for:

Unnamed	Any device with an empty Name field.
Unmatched Port to Pin	Port connectors which don't have a corresponding pin on the parent device.

Signal Errors

Six types of errors associated with signals may be located. They are:

Unnamed	The signal's Name field is empty.
Unconnected	The signal has one or zero pins attached to it.
Undriven	No output pins are attached to the signal.
Multiply Driven	Multiple output pins are attached to the signal.
Fanout > Limit	The number of input pins attached to the signal is greater than the specified amount.
Unmatched Page Connector	The signal has exactly one page connector attached to it.

Pin Errors

Three types of errors are associated with pins, they are

Unnumbered	The pin has no pin number.
Unconnected	The pin is unconnected, i.e. no signal line has been attached to it and it is unnamed.
Unmatched Pin to Port	The device has an internal circuit but this pin has not been matched with a port connector.

Ignoring Errors

ErrorFind allows you to indicate that certain errors should be ignored on specific objects. This facility is controlled by the following two buttons.

Mark Current OK	The particular error that caused the object to be found is disabled for the current object, i.e. the object whose name is in the status box.
Unmark Selected	All selected objects in the schematic are "unmarked", i.e. they will be checked for all errors in the future.

Information Fields

These fields may not be edited by the user but instead display results back to the user.

# Found	This field displays the total number of objects found that matched the criteria.
Current	This field indicates which of the found objects is currently selected.
status box	The status box at the bottom of the window displays the object’s hierarchical name, i.e.. the object’s name prefixed by its position in the hierarchy.

Other Controls

Select Found Objs	Selects all of the objects found.
Prev	Moves backwards in the list of found objects and selects the object before the current one.
Find/Next	This button has two operations. Initially, when the button is labeled "Find", it is used to initiate the search. Once the objects have been found then its label changes to "Next" and when pressed it advances the selection to the next found object. You can return to the "Find" by changing any of the search criteria or by making a structural change to the circuit.

Chapter XIII - The Browser Tool

Introduction

The Browser is a tool used to view, edit and navigate schematic data in spreadsheet-style format. Rows of the spreadsheet display objects in the schematic, either devices, signals, pins or nets. A cell of the spreadsheet displays an attribute value for the object in that row and the attribute field name in that column except for Net mode which displays one pin per column.

Installation

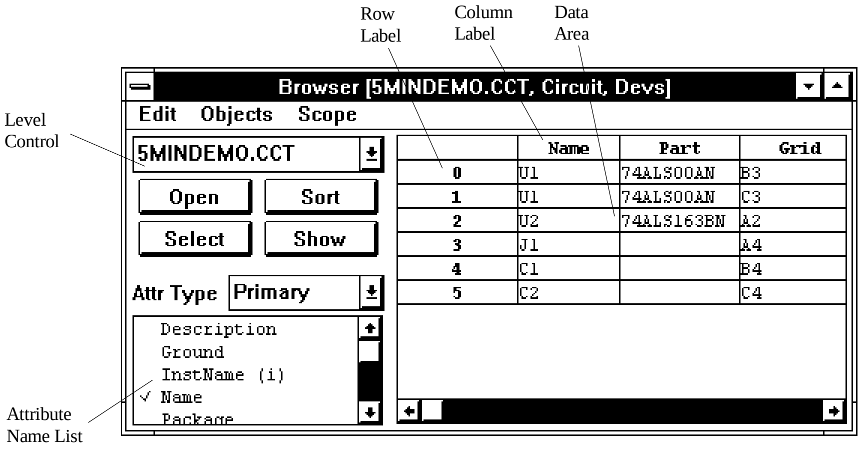
The Browser tool works with DesignWorks using the MEDA interface. In order to be located when DesignWorks starts up, it must be in the “Tools” directory, or in the directory specified in the TOOLSFOLDER statement in the DW.INI file.

Starting the Browser

To start the Browser tool, select it from the Tools menu in the DesignWorks window. Only a single Browser window may be active at one time across all designs currently open.

By default, the Browser tool will initially display all devices in the current circuit. If any objects are selected in the schematic, the Browser will display only selected devices or signals. Selected devices have priority over selected signals.

Browser Window



Browser [5MINDEMO.CCT, Circuit, Devs]

5MINDEMO.CCT

Attr Type Primary

	Name	Part	Grid
0	U1	74ALS00AN	B3
1	U1	74ALS00AN	C3
2	U2	74ALS163BN	A2
3	J1		A4
4	C1		B4
5	C2		C4

Item	Comment
Level Control	Displays the name of the current circuit level within the design being viewed. When selected, it displays the circuit hierarchy upwards from the current circuit. It can be used to navigate back to a particular level in the hierarchy.
Open Button	When a given device is selected in the data area, this control can be used to open the corresponding item in the schematic. See Browser Procedures / Button Usage (page 301) for more information.
Select Button	When devices, signals, pins or net connections are selected in the data area, this control can be used to select the corresponding item in the schematic. See Browser Procedures / Button Usage (page 301) for more information.
Sort Button	Sorts data in the data area. Columns to the left have priority over columns to the right. See Browser Procedures / Button Usage (page 301) for more information.
Show Button	When devices, signals, pins or net connections are selected in the data area, this control can be used to select and display the corresponding items in the schematic. See Browser Procedures / Button Usage (page 301) for more information.

Attribute Type	Shows and is used to select the type of attributes displayed in the Attribute Name List. See Browser Procedures / Displaying Attributes (page 300) for more information on displaying attributes.
Attribute Name List	Lists attributes of the type selected using the Attribute Type control. Attributes can be selected and deselected for display in the data area by double clicking on the attribute name. See Browser Procedures / Displaying Attributes (page 300) for more information.
Data Area	Displays attribute data associated with devices, signals, pins or net connections.

Browser Window Menu Commands

Edit Menu

Cut

Copies the currently selected data in the data area to the clipboard then clears the currently selected data in the data area.

Copy

Copies the currently selected data in the data area to the clipboard.

Paste

Pastes data from the clipboard into data cells starting at the top left cell selected. The number of cells filled depends on the format of the data on the clipboard. See the section “

Browser Procedures / Changing Attributes” (page 300) for limitations on changing attribute values.

Clear

Clears data in the currently selected cells.

Fill Down

Copies data in the cells at the top of the selection area to all subsequent selected cells below. See the section “

Browser Procedures / Changing Attributes” (page 300) for limitations on changing attribute values.

Fill Right

Copies data in the cells at the left of the selection area to all subsequent selected cells to the right

Make Visible

Makes object attributes included in the selection visible in the schematic. See the section “

Browser Procedures / Changing Attributes” (page 300) for limitations on changing attribute values.

Make Invisible

Makes object attributes included in the selection invisible in the schematic. See the section “

Browser Procedures / Changing Attributes” (page 300) for limitations on changing attribute values.

Objects Menu

Devices

The browser will display one line per device found in the current search scope.

Signals

The browser will display one line per signal net found in the current search scope.

Pins

The browser will display one line per pin found in the current search scope.

Nets

The browser will display one line per net with one cell used for each pin in the net.

Selected Devices

Same as devices but only currently selected devices will be displayed.

Selected Signals

Same as signals but only currently selected devices will be displayed.

Selected Pins

Same as pins but only currently selected devices will be displayed.

Selected Nets

Same as nets but only currently selected devices will be displayed.

Pseudo Devs

When checked, the browser will include pseudo devices (such as breakouts, page connectors, port connectors, etc.) in the browser data area when devices or selected devices are being displayed.

Parent Devs

When checked, the browser will display devices that contains subcircuits in the browser data area when devices or selected devices are being displayed. Normally, only the devices that appear in netlists are displayed.

This is most useful in Design scope where parent devices are not normally shown.

Scope Menu

Design

When checked, the entire design hierarchy is searched for objects of the specified type. Note that some of the objects attribute values displayed may not be editable because of definition/instance conflicts. Note that only the topmost level of a flat design will be scanned (unless overriden using the option outlined below).

Circuit

When checked, the search is limited to objects of the specified type that are contained in the circuit displayed in the last activated schematic window.

Here Down

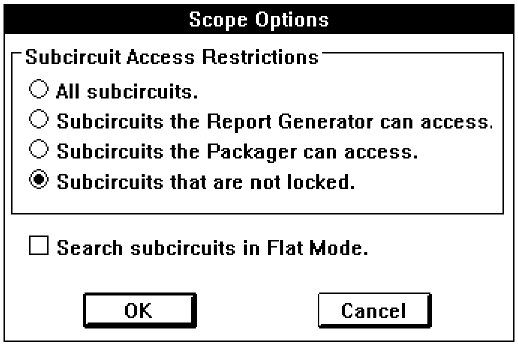
When checked, the search is limited to objects of the specified type that are contained in circuit displayed in the last activated schematic window (and all subcircuits in the hierarchy below this circuit). Note that only the topmost level of a flat design will be scanned (unless overriden using the option outlined below).

Page

When checked, the search is limited to objects of the specified type that are contained in circuit and in the page displayed in the last activated schematic window.

Options...

Objects included in the browser data area are set by the scope of the design selected (either design, circuit, here down or page). However, they can be further restricted using the following dialog:



Scope Options

Subcircuit Access Restrictions

○ All subcircuits.
○ Subcircuits the Report Generator can access.
○ Subcircuits the Packager can access.
● Subcircuits that are not locked.

☐ Search subcircuits in Flat Mode.

All subcircuits	All objects in all subcircuits are considered during the search
Report Generator	Only objects in subcircuits that are available to the report generator will be considered in the search.
Packager	Only objects in subcircuits that are available to the packager will be considered in the search
Not Locked	Only objects in subcircuits that are not locked are considered in the search
Search subcircuits in Flat Mode	This flag can be enabled to search all subcircuits in a flat mode design.

Browser Procedures

Displaying Attributes

Attributes to be displayed in the data area of the browser window are selected or deselected by double clicking on the attribute name in the attribute list. Attributes names have been grouped according to functionality and can be selected into the Attribute Name list using the Attribute Type control. The following groups have been provided:

Primary Attributes	These are attributes that have been selected in the Define Attribute Fields box in the Schematic Tool and marked as being primary. This is done to allow easier access to the more common attributes.
Secondary Attributes	These are all the attributes that are not marked as primary (see above).
Pseudo Attributes	These are characteristics about the object that the schematic knows but are not true attributes. These attributes may not be edited.
Definition Attributes	These are attributes that have only one value for all subcircuit instances
Instance Attributes	These are attributes that have a unique value for each subcircuit instance.
All Attributes	These are all attributes from the above groups.

Changing Attributes

Selecting & Editing Cells

A cell may be selected by clicking within its bounds. A range of cells may be selected by clicking in another cell with the shift key down. The selected cells will be inverted. Various menu commands can be used to affect selected cells.

A cell may be edited by double clicking in it (pseudo attributes cannot be modified). Modifying the value of a cell will cause the corresponding attribute value of the object in the schematic to change. The change happens **immediately** and there is **no undo**.

Keyboard Operations

The keyboard can be used to move within cells and between cells:

Key	Comment
Escape	Cancel changes made during this edit session (only valid during cell editing) and return to non-edit mode.
Enter (or Shift Enter)	Accept changes in the current cell (if editing), move to the next cell lower (higher) and begin editing this cell.
Tab (or Shift Tab)	Accept changes in the current cell (if editing), move to the next cell to the right (left) and begin editing this cell.
Arrow Keys	Move within a cell (if editing) or between cells.

Disallowed Changes

Any attribute value change that may cause a structural change to the connectivity of the schematic will be disallowed and aborted by the Browser. There are only a few special cases that will cause this to happen. For example:

- Renaming a visible signal to the same name as another visible signal on the same page would cause the two signals to become one signal.
- Making a signal visible which has the same name as an already visible signal with the same name would cause the two signals to become one signal.

Button Usage

Open Button

The Open button will be enabled if there is a single object selected in the spreadsheet that has either directly or indirectly a subcircuit. Objects that may be opened are Devices and Pins (both for Pin objects and Pin connections for Net objects). Pressing the button will cause the subcircuit to be opened.

It may not be possible to open a subcircuit. For example:

- The subcircuit is already opened for another instance (i.e. the subcircuit is already open for editing).
- The parent device of the subcircuit is not contained in an open window.

Select Button

The Select button will be enabled if there are objects selected in the spreadsheet. Pressing the button will cause the objects to be selected in the schematic but will not cause the schematic to be repositioned.

Note: If the object type being viewed is Selected Device, Selected Signal, Selected Pin, or Selected Net then this operation will cause the spreadsheet to update to display the new selected objects.

Sort Button

The Sort button causes the rows of the spreadsheet to be sorted into increasing order.

If the object type is Net or Selected Net, then the rows sorted by the name of the net that is in the row label area. For all other object types, the following comparison mechanism is used when sorting:

- The values of the first column are compared. If the values are not the same then this is sufficient to determine the sort order.
- If the values are the same then the next column is used to determine the sorting order. This is repeated as often as needed or until both rows are determined to have the same sort value.

Show Button

The Show button will only be enabled if there is a single object selected in the spreadsheet. Pressing the button will cause the schematic to be repositioned so that the object is visible. In addition, the object will be selected and all other objects in the same circuit will be deselected.

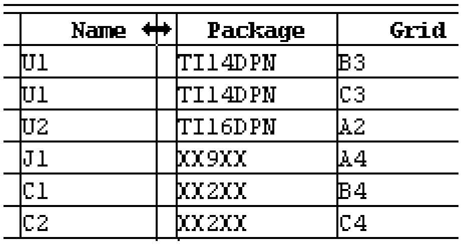
It may not be possible to show an object if the object is not already in an open window.

Note: If the object type being viewed is Selected Device, Selected Signal, Selected Pin, or Selected Net then this operation will cause the spreadsheet to update to display the new selected objects.

Adjusting the Spreadsheet

Sizing Rows & Columns

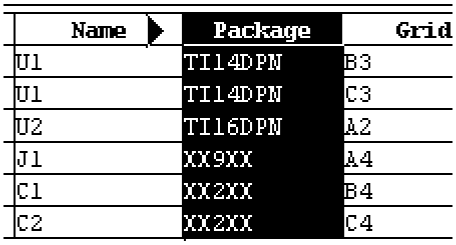
Rows and columns may be resized by moving the cursor into the row or column heading area and positioning it over the horizontal (row) or vertical (column) dividing line. When correctly positioned, the cursor will change shape. Press the mouse button and drag the line to size the row or column as desired.



Name	Package	Grid
U1	TI14DPN	B3
U1	TI14DPN	C3
U2	TI16DPN	A2
J1	XX9XX	A4
C1	XX2XX	B4
C2	XX2XX	C4

Moving Rows & Columns

Rows and columns are reordered by moving the cursor into the row or column heading area and positioning it over the row or column heading to move. Press the mouse button and drag the label to its new position using the arrow to determine where the final “drop” position will be.



Name	Package	Grid
U1	TI14DPN	B3
U1	TI14DPN	C3
U2	TI16DPN	A2
J1	XX9XX	A4
C1	XX2XX	B4
C2	XX2XX	C4

Note**:** It can be useful to move columns as the sorting mechanism is based on the column values.

Chapter XIV - Guide to Libraries

Introduction

The libraries provided with DesignWorks contain over 12,000 symbols including both ANSI and IEEE format symbols. The purpose of this manual is to describe the naming, graphical, attribute and organization conventions used in the libraries. This information will be of interest if you if you need to adhere to some particular drawing standard or if you plan to export data from DesignWorks to another system.

Installed and Optional Symbol Libraries

DesignWorks is shipped with compact and large libraries for ANSI symbols, IEEE symbols and large analog symbols:

- By default, the small (or compact) symbols are installed with the program. These are best suited for design on a small screen using a laser printer as an output device.

- If you require strict ANSI symbol sizes then you will need to install the optional ANSI symbol library containing the large symbols.

- If you require IEEE symbols then you will need to install the optional IEEE symbol library.

- If you require large size analog symbols then you will need to install the optional ANSI symbol library which includes the large ANALOG symbol library.

Symbol Format

The DesignWorks libraries include the ANSI (Large and Compact) and IEEE symbol equivalents for each part.

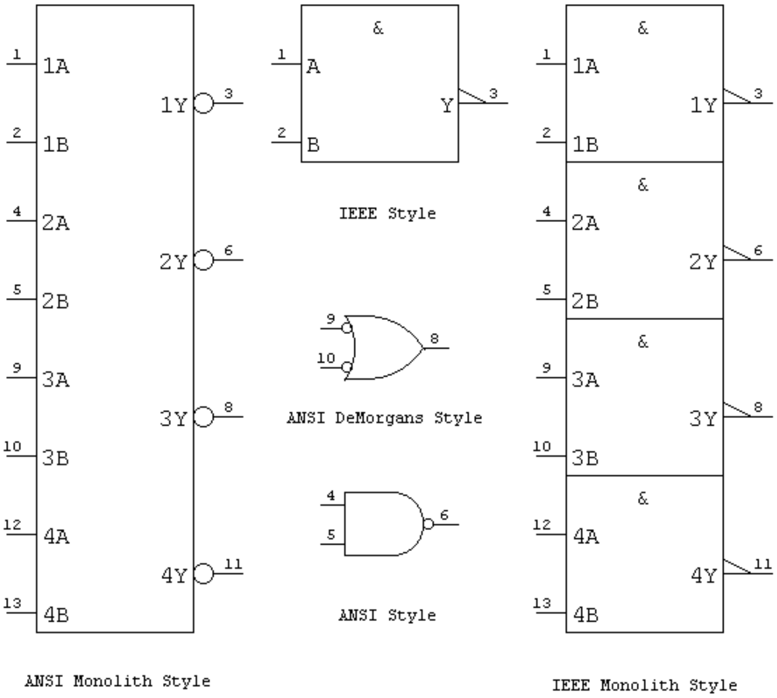


Figure 1 - The various symbol styles available

The basic installation of DesignWorks includes the Compact ANSI library symbols. These symbols are half the size of the standard ANSI symbols and are more suitable for most schematic design applications. The Compact ANSI symbol library has all the same parts as the ANSI symbol library.

Finding a Library

The BASICLIB directory contains commonly used libraries. By default these libraries are opened by the program at startup. These are:

74ALS_C.CLF	Compact ANSI 74ALS TTL symbols
74ALS_CD.CLF	Compact ANSI 74TTL Demorgan symbols
CONNECT.CLF	Connectors
DIODES.CLF	Diodes
DISCRETE.CLF	Discrete symbols
LINEAR.CLF	National Linear symbols
PRIMGATE.CLF	Primitive gates (for use with simulator)
PRIMIO.CLF	Primitive I/O (for use with simulator)
PRIMLOG.CLF	Primitive Logic symbols (for use with simulator)
PSEUDO.CLF	Pseudo Devices, various connectivity symbols.
SPICE.CLF	Spice symbols
TRANS.CLF	Transistors

The LIBS directory is used to store all other libraries and is itself divided into several directories. These are ANALOG, ANSICOMP, ANSI and IEEE.

Note: The default installation only installs the ANSICOMP set of symbols in the LIBS folder.

Within each of the ANSICOMP, ANSI and IEEE directories you will find:

54TTL	5400 TTL family symbols: 54, 54S, 54LS, 54AS, 54ALS, 54AC, 54ACT, 54F, 54HC, 54HCT, 54HCU.
74TTL	7400 TTL family symbols: 74, 74S, 74LS, 74AS, 74ALS, 74AC, 74ACT, 74F, 74HC, 74HCT, 74HCU.
75TTL	75ALS symbols.
CMOS	RCA CMOS library
ECL	ECL library symbols
MEMORIES	Motorola, Toshiba and Texas Instruments Memory symbols
MICPROCS	Motorola, Intel and Zilog microprocessor symbols
MISC	Some miscellaneous components
PLD	Programable Logic Device library symbols

Within the ANALOG (Large Analog symbols) directory you will find:

CONN_A_CLF	Connectors
DIODES_A_CLF	Diodes
DISC_A_CLF	Discrete symbols
MISC_A_CLF	Miscellaneous Spice symbols
NATLIN_A_CLF	National Linear symbols
SPICE_A_CLF	Spice symbols
TRANS_A_CLF	Transistors

NOTE: The large size analog libraries are installed along with the ANSI symbol library (Large ANSI parts).

The 54, 74 and 75 series components are divided into their families with each family having an expanded and monolith library. The ANSI and ANSICOMP symbols have additional libraries for DeMorgan equivalent gates.

The file names are designed to identify which style and symbol type they belong to. All libraries end with the extension ".clf":

_c	means compact ANSI style
_cd	means compact ANSI demorgan style
_cm	means compact ANSI monolith style
_i	means IEEE style
_im	means IEEE monolith style
_a	means ANSI style
_ad	means ANSI demorgan style
_am	means ANSI monolith style

For example:

- The 74ALS Compact ANSI library is called 74ALS_c.clf.
- The Compact ANSI 74ALS00 demorgan equivalent is found in the library 74ALS_cd.clf.

All the TTL components are derived from Texas Instruments data.

Interpreting Library Part Names

A component name is made of up to five parts.

- The Logic Family Name
- The Device Number
- Device Attributes
- Symbol Style
- DesignWorks unit for packaging

Most of these name parts are only used for the TTL family of parts. Other components may have just the manufacturer part name or a descriptive name as the library name, for instance "DIODE" or "2N222".

Some parts, like diodes, have over 100 different names in their attribute Part.List. In this case the library part name will be the first name that appears in the Part.List attribute (see below for a description of the Parts.List attribute).

Occasionally the part name is too long to display in the parts palette list, in this case the name will appear truncated.

The Logic Family Name

The logic family name is usually from one of the following:

- 74, 74S, 74LS, 74AS, 74ALS, 74AC, 74ACT, 74F, 74HC, 74HCT, 74HCU.
- 54, 54S, 54LS, 54AS, 54ALS, 54AC, 54ACT, 54F, 54HC, 54HCT, 54HCU.
- 75ALS.

The Device Number

The device number is the middle digits in the component name and corresponds to the symbol type. For example "123" in 74ALS123J.

The Device Attributes

The device attributes are the manufacturer attributes that apply to this symbol. The attributes are enclosed in parentheses "()" such as (W,J).

Symbol Style

The symbol style marker is one of the following:

A	This is an ANSI or Compact ANSI style symbol.
I	This is an IEEE style symbol.
D	This is a DeMorgan equivalent style symbol.
AM	This is an ANSI monolith or Compact ANSI monolith (non-expanded gates).
IM	This is an IEEE Monolith (non-expanded gates).

Unit for Packaging

The presence of a unit indicates that there are multiple symbols required to represent all the units in this package. The unit is appended to the name with a "." so ".a" would indicate that this component will default to unit "a".

If there are different symbols for some of the units within a package then the different units will each have their own entry in the library. The unit letter is appended to the end of the library name such as: "5450(J).a" and "5450(J).b". This component is a Dual 2-Wide 2-Input AND-OR-INVERT Gate (One Gate Expandable). In some cases the part that shows up as xxx.a will have a "c" and "d" units and the "b" unit will be a different entry because it does not share the same symbol.

Examples

Example 1:

A 54S00J may be listed in the library selection palette as "54S00(W,J)A"

"54S" is the TTL family name, this part of a library name is only used for the TTL components.

"00" is the component number

"(W,J)" indicates that there are two part suffixes for the 54S00 a "54S00J" and a "54S00W".

The "A" suffix indicates that this part is an ANSI style symbol.

Example 2:

A 5450J may be listed in the parts palette as "5450(W,J).aI".

"54" is the TTL family name, this part of a library name is only used for the TTL components.

"50" is the component number

"(W,J)" indicates that there are two part suffixes for the 5450 a "5450J" and a "5450W".

There may be a ".a" (or ".b", ".c" etc_) after the part suffix as in "5450(J).a". This is the unit and it indicates that there is more than one symbol for the gates in a package. Most of the TTL gates have multiple units in a package, the unit name only appears in the library if the graphic symbol for one or more of the units differs from the others.

The "I" suffix indicates that this part is an IEEE style symbol.

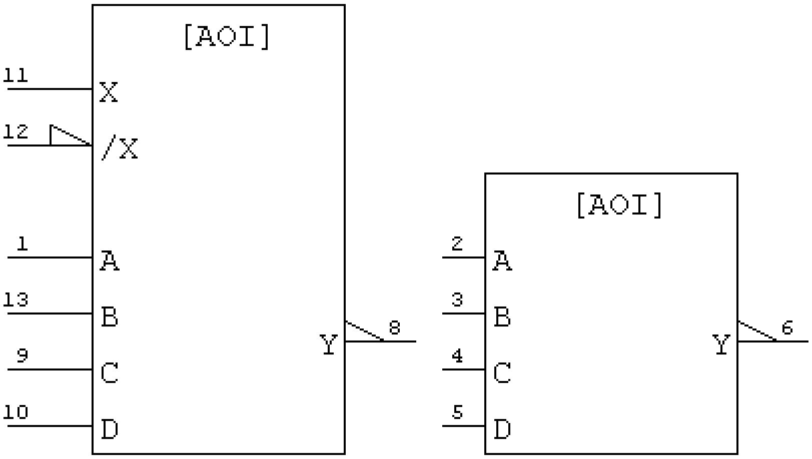


Figure 2 - Example Symbols

Selecting the Unit, Part and Package

A component may contain many elements, each element in a component is referred to as a *Unit*. For example, a 7404 hex inverter has six inverter units. In DesignWorks, units are identified by either a letter (a,b,c...) or a number (1,2,3...).

The *Part* is the manufacturers part number attached to a symbol.

The *Package* is a description of the footprint for a component, or which type of socket is required to hold a component of that type.

After a symbol has been placed on a design page it will have the default unit, part and package values. The defaults are always the first items in the Unit.List, Part.List and Package.List type attributes. To change the values for the Unit and Part attributes you must use the device Pop-Up menu. This is done by pressing the right mouse button with the mouse cursor over top of the device.

The *Unit* attribute is set by selecting a unit from the Unit menu item. It may be changed by the Packager when the part is placed. The *Part* attribute is set by selecting a part from the Part menu item. The *Package* attribute is set automatically when a new part from the Parts menu is selected.

If there is no Part.List or only one item in the list then the Parts menu item will not appear in the device Pop-Up menu. If there is no Unit.List or only one item in the list then the Units menu item will not appear in the device Pop-Up menu.

Example

> If you want a 54LS37W, Quad 2-input Positive Nand Buffer, you would open the library named 54LS and select the part "54LS37(W,J)" from the parts palette. To select between the 54LS37W and 54LS37J use the *device pop-up* menu by pressing the right mouse button with the cursor over the part. In this menu the *part* pull-right menu item is used to select which of the valid parts you wish to use.

Also in the device pop-up menu is a unit entry. This is used to select which of the units in a package you wish. For example in a quad nand gate there are 4 units (a,b,c,d). The default is unit a, selecting unit b causes the pin numbers on the symbol to change to the second nand gates pin numbers. The packager tool (described elsewhere) will do this automatically for you when you are placing parts.

Attributes in Library Symbols

Attributes on Types

Items noted with an asterix (*) must be properly defined in order for the packager to function correctly.

Name.Prefix

This is the string used as a prefix to auto name a device when it is placed.

Name.Spice

This is the string used as an optional prefix when creating designs for SPICE-based analog simulators.

Name.Pt

This type attribute contains a formatted string specifying the X and Y coordinates of the Name attribute that is generated when a symbol is placed. The format for the string is "(X-coord, Y-coord)". For example "(150, -75)". The values are in 1000ths of an inch and represent the point relative to the symbol. For more details on how to set this see *Default Position Fields* in the manual section on Attributes.

Part (*)

This type attribute contains the part name that this symbol represents. The value for this attribute is the manufacturer part name, e.g. "74ALS00AN".

Part.List

This type attribute contains a list of all the part names that this symbol may represent. The values for this attribute are the manufacturer part names, e.g. "74ALS00AN, 74ALS00AD". This list is linked to the Package.List, see Package.List below.

Part.Pt

This type attribute contains a formatted string specifying the X and Y coordinates of the Part attribute that is generated when a symbol is placed. The format for the string is "(X-coord, Y-coord)". For example "(150, -75)". The values are in 1000ths of an inch and represent the point relative to the symbol. For more details on how to set this see *Default Position Fields* in the manual section on Attributes.

Unit (*)

This is the unit that will be selected as the default unit when a device is placed. It is usually "a".

Unit.List (*)

This type attribute field contains a list of the units available for this symbol. The unit names are separated by commas, e.g. "a,b,c,d". In types having different symbols for the logical units in one package, this is the list of only those units which can be used with this symbol.

Unit.All (*)

This type attribute field contains a list of all the units available in this package, regardless of whether or not they apply to this symbol. Normally this will be the same as "Unit.List", but will be different for devices having more than one symbol for the logical units in a package, for instance an And-Or-Invert gate.

Power

This type attribute contains a list of the pins connected to power sources in this symbol. The list is separated by commas, e.g. "1, 20".

Ground

This type attribute contains a list of the pins connected to ground in this symbol. The list is separated by commas, e.g. "10, 39".

UnusedPins

This type attribute contains a list of the unused pins in this package. The list is separated by commas, e.g. "13,23".

Permutable

This type attribute contains a list of the swappable pins and gates in this package. It is a list of bracketed pin numbers. Pin numbers are separated by commas, Parentheses "()" enclose items that can be swapped while brackets "[]" enclose items that cannot be swapped. This attribute is intended for use by PCB packages. For more information on the format of the Permutable attribute see the description elsewhere in this manual .

Restrict

This type attribute is an integer value from 0 to 7 used to set access restrictions for the device's internal circuit. The value is the sum of the following:

1 = Don't report
2 = Don't package
4 = Don't push into

The normal setting for this attribute in library parts is 3 (don't report and don't package).

PkgLevel

This type attribute is an integer value from 0 to 3 used to control packaging on this part. Possible values are:

0 = Normal
1 = Lock and check
2 = Lock and don't check
3 = Ignore

The normal setting for this attribute in library parts is 0 (Normal).

Function

This type attribute contains a terse symbol function code. A full list of the currently used function codes and category codes is given below.

Category

This attribute contains a terse symbol category code. A full list of the currently used function codes and category codes is given in the table below. This list will of course be augmented as required for future libraries.

Category	**Functions**
ARITH	ACCUM, CARRY, ALU, MULT, ADD
BUFF	BUFFINV, DELAY, BUFF, INV
COMP	ADDCOMP, ERRDET, PARITY, COMP
CONN	CONNM, CONNF
CTR	PRESCALE, TIMER, OSC, FREQDIV, RATEMULT, CTR
DIODE	TUNNEL, LED, BRIDGE, ZENER, DIODE
ELMECH	RELAY, SWITCH
FF	DFF, JKFF

GATE	OAI, XNOR, GATE, XOR, AOI, OR, AND, NOR, NAND
LATCH	MULTIVIB, LATCH
LIN	ADCONV, VOLTREG, LIN, OPAMP
MEM	ROM, EPROM, EEPROM, PROM, DRAM, FIFO, SRAM
MISC	LVLCNV, PLL, RCVR, MISC, FILTER, DSWITCH, PLSSYNCH
MUX	AMUX, BITSHIFT, MUX, ENCODE, DECODE
NETSYM	PWR, GND
PASS	XTAL, XFMR, TEST, RES, IND, FUSE, CAP, BAT
PLD	PLD, EPLD, PAL, FPLA
PROC	COPROC, MPERIPH, MCTRL, PROC, MEMCTRL, MEMMAP
REG	REG, SHREG
THYR	UJT, TRIAC, THYR, SCR, DIAC
TRANS	TMOS, PNP, NPN, JFET, MOSFET
XCVR	XCVR

Description

This type attribute contains a short description of the symbol type. For instance "Quadruple 2-Input Positive-NAND Gates".

Package

This type attribute contains the package code for the part defined in the "Part" attribute. For a description of currently used package codes see the description elsewhere in this manual.

Package.List

The list of package codes that are used by this symbol type. The list is separated by commas where each element corresponds to the package used for the part that holds the same position in the Part.List string.

For example:

- if the Part.List = "74ALS00AN,74ALS00AD".
- then the Package.List = "TI14DPN,TI14SN",
- where TI14DPN is the package for the first element of the Part.List 74ALS00AN and so on.

Attributes on Pins

VisPin.List(*)

This type attribute is set for each pin on a symbol. It contains a list of pins numbers for each unit in a package. The list is separated by commas, e.g. "1,4,9,12". For instance if the symbol was for a 2-input quad nand gate then one of the input pins would have a VisPin.List attribute of "1,4,9,12". For the "a" unit the pin number is "1"; for the "b" unit the pin number is "4" and so on.

Pin Permutability

Note: The permutability of pins on a device is controlled exclusively by the packager contained within DesignWorks. The information below is provided for reference and information only.

Permutability of pins on a device is indicated (but not used) by an attribute in the library type record.

1. The attribute used is "Permutable".
2. The value of the Permutable attribute is a string of bracketed pin numbers.
3. There are two types of brackets; ’(’ ’)’ enclosing a list of permutable pins; ’[’ ’]’ enclosing a list of non permutable pins.
4. Pin lists are separated by ’,’ characters.
5. Pins that are not in the list are not permutable.
6. The permutable and non-permutable brackets are nested to indicate conditions where individual pins are not permutable but as a group they may be.

    - A list of pins inside a () are permutable. e.g. (1,2,3).
    - A list of pins inside a [] are not permutable. e.g. [1,2,3].
    - Lists of non permutable pins [1,2,3] inside a () indicates that the individual non permutable pins may be permuted as a group. e.g. ([1,2,3],[4,5,6])
    - Complex permutations as in Example 3 below may indicate that within a non permutable unit some of the pins may be permuted. Such as the A and B inputs on a NAND gate e.g. (1,2) but the output Y must not be swapped with another pin without swapping both A and B. e.g. [(1,2),3] so we get a list like ([(1,2),3],[(4,5),6]).

This means pins 1 and 2 may be swapped (they are functionally identical). Pins 4 and 5 may be swapped. If we swap pin 3 or 6 we must swap 1,2,3 or 4,5,6 at once.

Example 1

A 54LS74 has two D type flip flop units each identical. To permute one pin we must permute all pins on one FF with the ones on the other FF. This is indicated as follows:

Permutable = "([14,1,2,3,13,12],[8,7,6,5,9,10])";

pins 14,1,2,3,13,12 are non permutable with each other.

pins 8,7,6,5,9,10 are not permutable with each other.

But as a group pins [14,1,2,3,13,12] are permutable with pins [8,7,6,5,9,10].

Example 2

A 54LS04 Hex Inverter. The two pins on each device are not permutable since one is an input and the other is an output. But each of the six inverters can be exchanged as a whole with another.

Permutable = "([1,14],[3,2],[5,6],[7,8],[9,10],[13,12])";

Example 3

A 54LS00 Quad 2-Input Nand Gate. The 4 individual units of the NAND may be permuted as a whole as well the two inputs of each may be permuted. This is indicated as follows:

Permutable = "([(1,2),3],[(6,7),5],[(9,10),8],[(12,13),14])";

In this example each non permutable group "[(1,2),3]" is one of the NAND gate units for this unit pins 1 and 2 are permutable. Pin 3 is not. However each unit may be permuted as a whole.

Example 4

A 5475J 4-Bit Bistable Latch. There are 4 nearly identical latches in this package. Each has a D input, a Q and \Q output and an enable (C). However there are only 2 enables for the 4 bits. Bits 1 and 2 have a common enable (Enable 1-2) and bits 3 and 4 have a common enable (Enable 3-4). This differentiates bits 1 and 2 from bits 3 and 4. The D,Q,\Q pins on bit 1 may be exchanged with the same pins on bit 2, but if we exchange bit 1 for bit 3 or 4 we must also exchange the Enables. This is indicated as follows:

([13,([2,16,1],[3,15,14])],[4,([6,10,11],[7,9,8])])

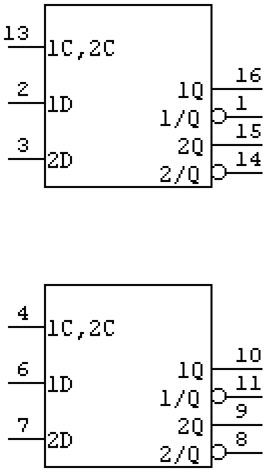


Figure 3 - Example Symbols

Package Codes

Introduction

Package codes have three parts:

- Manufacturer Name
- Number of Pins
- Package Style code

The manufacturer name may be any string of characters although it is advised that no numeric digits should be used since that may confuse the format, the first digits in the package name should be the number of pins used in the package.

Example

TI14DPN

This package is manufactured by Texas Instruments (TI) there are 14 pins on the device and the style is a narrow, plastic dual in line package.

In general the physical dimensions of the packages are not defined by these codes. The DIP package types are standard, others may vary with the manufacturer. The package codes define the sets of foot prints required for PCB layout, it is up to the PCB layout tool to define the actual physical dimensions for the package foot print.

The Package styles used by DesignWorks are as follows:

Package Type	**Code**
DIP, plastic narrow	DPN
DIP, plastic medium	DPM
DIP, plastic wide	DPW
DIP, plastic x-wide	DPX
DIP, ceramic narrow	DCN
DIP, ceramic medium	DCM
DIP, ceramic wide	DCW
DIP, ceramic x-wide	DPX
DIP, side brazed narrow	DBN
DIP, side brazed medium	DBM
DIP, side brazed wide	DBW

DIP, side brazed x-wide	DBX
DIP, windowed narrow	DVN
DIP, windowed medium	DVM
DIP, windowed wide	DVW
DIP, windowed x-wide	DVX
MODULE - Leadless/single side	M
MODULE - Leadless/double side	MD
MODULE - Single Sided with Edge Clipped Leads	E
MODULE - Double Sided with Edge Clipped Leads	ED
MODULE - Single Sided with Pin Centered	I
MODULE - Double Sided with Pin Centered	ID
ZIP	Z
Flatpack	F
Flatpack, windowed	FV
Flatpack, Sidebrazed	FB
Quad Flatpack	Q
SOIC narrow	SN
SOIC medium	SM
SOIC wide	SW
SOIC x-wide	SX
Quad SOIC, Gullwing	G
Quad SOIC, Gullwing/Rectangular	GR
SOJ (J-bend SOIC)	J
20/26 pin SOJ	J
SOIC, ceramic	JC
Pin Grid Array	A
Pin Grid Array, windowed	AV
Pin Grid Array, plastic	AP
Pin Grid Array, cavity down	AD
Pin Grid Array, with Heatsink	AH
Leadless chip carrier, ceramic Square	L
Leadless chip carrier, ceramic Rectangular	LR
Leadless chip carrier, sidebrazed Square	LB
Leadless chip carrier, sidebrazed Rectangular	LBR
Leadless chip carrier, windowed Square	LV
Leadless chip carrier, windowed Rectangular	LVR

Leadless chip carrier, plastic Square	P
Leadless chip carrier, plastic Rectangular	PR
Leadless chip carrier, ceramic Square	C
Leadless chip carrier, ceramic Rectangular	CR
Leadless chip carrier, ceramic windowed Square	CV
Leadless chip carrier, ceramic windowed Rectangular	CVR
Leadless chip carrier, plastic with bizarre attributes	PQ

Appendix A - Device Pin Types

Every device pin has a characteristic known as its *pin type*. The pin type is set when the part entry in the library is created and cannot be changed for individual device pins on the schematic.

See Chapter X - Device Symbol Editing for information on how to set the pin type while creating a device symbol.

What Pin Types are Used For

For many general schematic editing purposes, the pin type will be unimportant an can be ignored. However, pin type settings are important in the following cases:

- Pin type settings are crucial if you plan to use the DesignWorks Digital Simulation option. The pin type of each pin determines the direction of signal flow and must be appropriately set for each symbol.
- Pin type information is required in many netlist file formats for FPGA layout and digital simulation.
- Correct pin type settings allow the ErrorFind module to check for fanout and multiple drive situations.
- Other future analysis tools may use this information for timing and loading analysis.

Pin Types Table

The following table lists the function of each of the pin types available in DesignWorks.

Pin Type	**Description**
IN	Input - this is the default for pins created using DevEditor. This setting is used for all pins on discretes except those with some digital function.
OUT	Output - always enabled.
3STATE	Output - can be disabled (i.e. high-Z)
BIDIR	Bidirectional
OC	Open collector output - i.e. pulls down but not up

BUS	Bus pin - This does not represent a physical signal but is a graphical representation of a group of internal pins, each having its own type.
LOW	Output - always driving low
HIGH	Output - always driving high
LTCHIN	Input to a transparent latch - this is used for calculating cumulative setup and hold times.
LTCHOUT	Output from a transparent latch - this is used for calculating cumulative setup and hold times.
CLKIN	Input to an edge-triggered latch - this is used for calculating cumulative setup and hold times.
CLKOUT	Output from an edge-triggered latch - this is used for calculating cumulative setup and hold times.
CLOCK	Clock input - this is used for calculating cumulative setup and hold times.
OE	Open emitter output - i.e. can pull up but not down.
NC	A no-connect pin

Appendix B - Primitive Device Types

Every device on a DesignWorks schematic has a characteristic known as its *primitive type*. The primitive type is set when the part entry in the library is created and cannot be changed for individual devices on the schematic.

Primitive types fall into three general groups:

• **Schematic symbols:** The two primitive types SUBCIRCUIT and SYMBOL fall into this category and are the normal primitive types used for creating schematic symbols. SUBCIRCUIT is the default type for symbols created using the DevEditor. There are no restrictions on the ordering or type of pins on these symbols.

• **Pseudo-device types:** These are the symbols used for page connectors, power and ground symbols, etc.

IMPORTANT: DesignWorks has very specific requirements for the order and type of pins on pseudo-devices. Refer to the following table for information. These rules ARE NOT CHECKED by the DevEditor.

• **Simulation types:** The majority of the primitive types defined in the following table are simulation primitives and are intended for use with the DesignWorks Digital Simulator option.

IMPORTANT: The simulation primitive types should not be used for user-created symbols without a clear understanding of their function. See the Digital Simulator manual for more information.

Schematic Symbol Primitive Types

Primitive Type	**Pin Requirements**	**Comments**
SUBCIRCUIT	No restrictions	Symbol having an optional internal circuit. This is the default for symbols created using DevEditor.
SYMBOL	ENo restrictions	Symbol with no internal circuit.

Pseudo-Device Primitive Types

IMPORTANT: The pin requirements listed in the following table must be followed when creating pseudo-device symbols. These rules ARE NOT CHECKED by the DevEditor.

Primitive Type	**Pin Requirements**	**Comments**
BREAKOUT	Pin 1 is Bus Pin, followed by N Normal Pins, set to Input	Splits signals out of or into a bus.
SIGNAL CONNECTOR	Exactly 1 normal pin, normally set to Input	Used for power and ground connections.
PORT CONNECTOR	Signals - exactly 1 pin, Busses-exactly 1 bus pin with any number of internal pins	Makes a connection between the signal it is connected to and a like named pin on the parent device.
PAGE CONNECTOR	Signals - exactly 1 pin, Busses-exactly 1 bus pin and no internal pins	Exports the signal or bus globally across all pages at one level of a schematic.

Simulation Primitive Types

WARNING: Do not use these without referring first to the Digital Simulation manual for more information!

Primitive Type	**Pin Requirements**	**Comments**
NOT	1 Input, 1 Output, ...	Output complement of Input. Same for each gate.
AND	N Inputs, 1 Output	Output <- AND of all Inputs
OR	N Inputs, 1 Output	Output <- OR of all Inputs
XOR	N Inputs, 1 Output	Output <- XOR of all Input
XNOR	N Inputs, 1 Output	Output <- XNOR of all Input
D-FF	/Set, Data, Clock, /Reset, Q, /Q	
JK-FF	/Set, J, /Clock, K, /Reset, Q, /Q	
D-FF (non inv)	Set, Data, Clock, Reset, Q, /Q	Set and Reset are active high.
JK-FF (non inv)	Set, J, /Clock, K, Reset, Q, /Q	Set and Reset are active high.
BUFFER	1 Input, 1 Output, ...	Output is Input. Same for each gate.
MULTI-PLEXER	N Data, M Select, /E, Output	Routes 1 of N inputs to the output. M is the smallest value such that 2M _ N
DECODER	N Select, M Output, /E,	Selects 1 of M outputs. N is the smallest value such that 2N _ M
REGISTER	N Inputs, N Outputs, Clock, Clear	Inputs are latched when clock is high, clear is async.
SHIFT REG	N Inputs, N Outputs, Clock, Shift/~Load, Shift-In	Shifts or loads data on clock.
ADDER	N A-Inputs, N B-Inputs, N Outputs, Carry In, Carry Out	Adds A and B w/Carry
COUNTER	N Inputs, N Outputs, Clock, Count/~Load	Counts up or loads on clock.
X-GATE	2 Bidirectional, Control, ...	The 2 bidirectional pins are connected if control is active. Repeated for each gate.
RESISTOR	2 Bidirectional, ...	Converts high drive to low drive for each resistor.
PULL_UP	1 Bidirectional, ...	Outputs a high with a low drive level for each pin.
ONE SHOT	Clock, /Clear, Q, /Q	Retriggerable. Delay is 1 by default. Delay setting is used as pulse width.

CLOCK	1 Output	The delay setting is used as the high and low period of the clock.
PROBE	1 Input	Displays Boolean input value on the schematic
SPST-SWITCH	2 Bidirectional Pins	Makes a connection between the two pins when switch is closed, otherwise both pins are undriven.
SPDT-SWITCH	3 Bidirectional Pins	Has pins A, B, and C; A and C are connected or B and C are connected.
HEX KEYPAD	Q0..3 Outputs, Strobe Output	Outputs a 4 bit hex value, strobe goes high when key pressed.
PUSH BUTTON	3 Bidirectional Pins	A, B, and C Pins; A and C are connected or B and C are connected. Switch is momentary.
SETUP & HOLD	Clock, 1 Data (Input), Error (Output)	Error = X if Data changed after Setup or before Hold time, Error = Z otherwise.
GLITCH DETECTOR	Sense (Input), Error (Output)	Error = X if sense changes faster then delay setting, Error = Z otherwise.
UNKNOWN DETECTOR	Sense, Output	If Sense = 0 or 1 then Output = 0, Output = 1 otherwise.
SIMULATOR STOP	1 Input	Performs requested task (stop simulator, generate report, ...) on rising edge.
LOGICBOX		Represents LogicBox hardware interface. See LogicBox Manual.

Appendix C - Predefined Attribute Fields

The following tables list fields that are predefined for each new design that is created. These fields cannot be removed because many of them are used by specific DesignWorks features.

For convenience, the fields marked as "primary" by default are listed first since they are most commonly entered directly by the user.

The following information is provided in the table:

Field Name The name of the field as it appears in the Define Attributes box.

Used In The types of objects this field is used in.

Instance/ Definition Specifies whether the field data is kept with the definition or instance or a sub-circuit.

Description How the field is used.

NOTE: Many of the predefined fields are used for specific DesignWorks features. More details on the usage of these fields can be found in the manual sections covering those features.

Primary Fields

The following fields are the ones most commonly edited directly by the user.

Field Name	**Used In**	**Inst/Def**	**Description**
CctName	Design	Def	Design file name. Sets the window title and name of next saved file.
Description	Devices	Def	A short description of the device type. This is included in all standard DesignWorks libraries. Not used internally.
Ground	Devices	Def	List of pins to be hooked to ground net in netlist, separated by commas. Format example: 7,10,13

InstName	Devices, signals	Inst	Name applied to physical instance. This is normally used for the package name in hierarchical designs. Normally not used in Flat or Pure mode designs.
Name	Devices, signals	Def	Definition name, i.e. reference designator. This is the field set using the text tool on the schematic.
Package	Devices	Def	Device package code. This is included in all standard DesignWorks libraries. Not used internally.
Part	Devices	Def	Part type code. Used by Packager to assign gates to packages.
Power	Devices	Def	List of pins to be hooked to power net in netlist, separated by commas. Format example: 1,2,14
PrefixField	Design	Def	Contains the name of the field to use as a name prefix for device packaging and device auto-naming.
Spice	Devices, Design	Def	Holds simulation parameters for SPICE-based simulators. Not used internally.
Unit	Devices	Inst	Gate unit, e.g. a, b, c, etc. Set and checked by Packager or can be set manually.
Value	Devices	Def	Component value to appear in netlists. Not used internally.

Secondary Fields

In general, values in the secondary fields should not be edited directly by the user without a thorough understanding of their effect.

Field Name	Used In	Inst/Def	Description
AutoSym.Bottom	Devices	Def	The string used in the "bottom pins" box in the Auto Create Symbol command in the DevEditor.
AutoSym.Left	Devices	Def	The string used in the "left pins" box in the Auto Create Symbol command in the DevEditor.
AutoSym.Right	Devices	Def	The string used in the "right pins" box in the Auto Create Symbol command in the DevEditor.
AutoSym.Top	Devices	Def	The string used in the "top pins" box in the Auto Create Symbol command in the DevEditor.
BusInfo	Pins	Def	Used only on bus breakouts to notate the names of the attached signals. Normally only updated using the Bus Pin Info command.
Category	Devices	Def	Contains a terse device category code. Can be used as an alternate device prefix by entering the name of this field in PrefixField. Not used internally.
CctPath	Design	Def	The directory path name for the design file. This is used only for creating external circuit references for sub-circuit parts. Changing it will not affect the current design file.
Depth	Devices	Def	Indicates nesting depth of device sub-circuit. Only used temporarily by the Report tool while generating hierarchical reports. This value is not maintained during editing operations.
DevPrefix	Design	Def	Default prefix for auto-assigned device names. Normally set using the Design Preferences command.

ExtCctName	Devices	Def	Name of external design file containing device internal circuit. Normally set by the DevEditor.
ExtCctPath	Devices	Def	Directory path for design file containing device internal circuit. Normally set by the DevEditor.
Function	Devices	Def	Contains a terse device function code. Can be used as an alternate device prefix by entering the name of this field in PrefixField. Not used internally.
HierNameSep	Design	Def	Contains the separator string used in generating hierarchical device and signal names for Report output. Normally set only by commands in a report form file.
LibDate	Devices	Def	"Last Modified Date" of the library the device was read or updated from, stored in encoded form. This is used by the Update from Lib command.
LibName	Devices	Def	Name of the library the device was read or updated from. This is used by the Update from Lib command.
LibPath	Devices	Def	Directory path of the library the device was read or updated from. This is used by the Update from Lib command.
LibType	Devices	Def	The original name of the part when it was read from a library. This is used by the Update from Lib command.
Name.Prefix	Devices	Def	Prefix used to create default Name value. Used by Packager and Auto-name.
Name.Pt	Devices	Def	Default position of Name attribute relative to device.
Name.Spice	Devices	Def	Alternate prefix for SPICE-based simulators.
OKErrors	Devices, Signals, Pins	Def	Used by ErrorFind tool to mark errors set as OK by user. Should not be set manually.

Package.List	Devices	Def	List of package types that apply to this part type.
PageRef	Devices	Def	Used in Page Connector pseudo-devices only. Contains the auto-generated page references.
PageRefFormat	Design	Def	Format string for page references. Normally modified only using the Design Preferences command.
PageRefWidth	Design	Def	Max. number of items per line in page references. Normally modified only using the Design Preferences command.
Part.List	Devices	Def	List of part types possible for this symbol.
Part.Pt	Devices	Def	Default position of Part attribute relative to device.
Permutable	Devices	Def	This field contains a list of the swappable pins and gate sections in this part type. Parentheses "()" enclose items that can be swapped, while brackets "[]" enclose items that cannot be swapped. This is intended for use by external PCB layout packages. Not used internally.
PinSequence	Devices	Def	This field is used to determine pin order in netlist output formats that require specific device pin ordering, e.g. SPICE. This is not defined in the standard DesignWorks libraries. Not used internally.
PkgLevel	Devices	Def	An integer 0 to 3 specifying packaging action: 0 = normal 1 = lock and check 2 = lock and don't check 3 = ignore Normally set using the Device Info or Get Info commands.
PkgPrefix	Design	Def	Default prefix used by Packager to create a package name. Normally set using the Default Prefix command.

Restrict	Devices	Def	An integer value from 0 to 7 used to set access restrictions for the device's internal circuit. The value is the sum of the following: 1 = Don't report 2 = Don't package 4 = Don't Push Into Normally set using the Get Info or Device Info command.
SigPrefix	Design	Def	Prefix used to generate default signal names. Normally set using the Design Preferences command.
Unit.All	Devices	Def	List of all units available in this part type. Used by Packager.
Unit.List	Devices	Def	List of all units in this part type having this symbol. Used by Packager.
UnusedPins	Devices	Def	List of unused pins in this part type. Not used internally.
VisPin.List	Pins	Def	List of pin numbers for this pin corresponding to the gate units in "Unit.List". Used by Packager.

Appendix D - Ini File Format

The "DW.ini" file is divided into sections. Each section starts with a section heading contained within square braces, eg. [Drawing]. Within a section each non-blank line is either a statement or a comment. A statement is a keyword, which specifies an option, followed by an equal sign, which is a separator, followed by the option’s value. Statements are terminated by a carriage return. A comment is any line that starts with "//".

Section [Drawing]

Initial Directory Settings

```
Directory = dir_name
```

This statement specifies the initial working directory. The default working directory if this statement does not appear in the Ini file is the same as set by Window’s Program Manager.

Font Settings

```
XXX_Font = font_name font_size [BOLD ITALIC]
```

This statement specifies the font for text items appearing in a schematic document.. "font_name" is the name of a TrueType font, only TrueType fonts are supported. "font_size" is the point size to use. There are two optional style keywords which may be applied, "BOLD" and "ITALIC". The possible items which may have there font specified are:

Default_Font

Attribute_Font, Border_Font, MiscText_Font, Pin_Font, Symbol_Font

Color Settings

```
XXX_Col = RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW
          BLACK, DKGRAY, GRAY, LTGRAY, or WHITE
```

This statement specifies the color for a item(s) appearing in a schematic document. All items in a schematic, except the page background, default to black. The page background defaults to white. The possible values for XXX_Col are:

Default_Col

DeviceAttrs_Col, SignalAttrs_Col, BusAttr_Col, PinNumber_Col, PinNumber_Selected_Col

Device_Col, Signal_Col, Signal_Selected_Col, Pin_Col, Pin_Selected_Col, Bus_Col, Bus_Selected_Col, BusPin_Col, BusPin_Selected_Col

Page_Col, Boundary_Col, GridMajor_Col, GridMinor_Col, RandowText_Col, RandowTextFrame_Col

Adding Default Attribute Field Definitions

The setup file keyword ATTRFIELD allows you to specify attribute field definitions that will be included in every new design that is created. The format of this keyword is as follows:

```
ATTRFIELD = fieldName
ATTRFIELD = fieldName/options;
```

"fieldName" can be any valid attribute field name, i.e. consisting of letters, digits and the period and underscore characters "." and "_". The maximum field name length is 16 characters and spaces are not allowed.

If no options are specified, the field will be a device field with default maximum length and switch settings.

A number of switch setting options can be specified as letters and digits after a "/" following the field name. If any digits follow the "/", they are taken to be the maximum data length for this field. The default is 32000. Any combination of other option letters can be selected from the following table. The order of the option letters is not important.

Any number of new fields can be added by using multiple ATTRFIELD lines. Any line specifying the name of an existing field will be ignored.

Option Letter	**Meaning**
A	Allow Carriage Return character in data
C	Allowed in design
D	Allowed in devices
F	Fixed, i.e. user cannot remove field definition
G	Group with Name
I	Instance data, i.e. can have different value inside different instances of the same hierarchical block
L	Location fixed on schematic
N	Only allow name characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
P	Allowed in pins
R	Rotate with object
S	Allowed in signals
T	Do not display value list, even if a .List field is present
U	Only allow numeric characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
V	Visible by default
W	Show Field Name by default
X	Read only value
Y	In Primary list by default

ATTRFIELD examples:

```
ATTRFIELD = Aperture/SY10
ATTRFIELD = InputDelay/PYU10
ATTRFIELD = ModelText
```

Setting New Design Command Action

The NEWDESIGN Ini file keyword allows you to control the function of the New Design menu command. If this keyword does not appear in the setup file, the New Design command will display the mode selection box described earlier in these notes.

NOTE: If you hold the Shift key pressed on the keyboard while selecting the New Design command, the New Design mode box will be displayed regardless of the settings in the NEWDESIGN setup file keyword.

The NEWDESIGN keyword can be followed by arguments in one of the following two forms:

```
NEWDESIGN = designMode pinMode
NEWDESIGN = fileName
```

In the first form, NEWDESIGN is followed by two integers, separated by blanks. These numbers in effect select the option buttons in the New Design mode selection dialog. The first integer can be from 1 to 7 and selects one of the modes described in the New Design command in the order that the buttons appear in the box. I.e. 1 = Flat - PCB and 7 = Pure Hierarchy - General. The second integer selects the pin space mode and can be 1 (DesignWorks 1/8") or 2 (ANSI 1/4").

The second form of the NEWDESIGN keyword allows you to specify the name of a file to open when the program is started or the New Design command is selected. This file can be an empty template containing your standard attribute fields and sheet settings. The format for the file name is the same as described for the DESIGN keyword. This keyword is equivalent to performing an Open on the file except that the window will be titled "Designxxx" and the Save command will be disabled to avoid accidentally overwriting the template file.

NOTES: The name of the file specified after NEWDESIGN cannot start with a digit or the first form will be assumed.

NEWDESIGN examples:

```
NEWDESIGN = 2 1
NEWDESIGN = Template\Design.cct
```

Default Design

```
DESIGN = circuitName
```

This statement allows you to specify a circuit file to open when the program starts. This can be used to open a file that is being repeatedly edited, or to open a default "template" file with a standard title block or border. NOTE: If the circuit is in any directory other than the Working Directory a "pathname" to must specified.

Disabling Untitled Design at Startup

This statement controls the creation of a new, untitled design when the program is first started. If the NOUNTITLED statement has a value of On, the program will start up and just display its menus, with no default window opened. The default is Off. This entry will appear as follows:

```
NOUNTITLED = on
```

Solid Grid Lines

The SOLIDGRID Ini file keyword determines if grid lines are drawing with solid lines. If not on then the grid is drawn with the default dotted lines. On some platforms dotted lines are not supported correctly or may be slow.

```
SOLIDGRID = on
```

Zoom Factors

```
SCALES = n1..n11
NORMALSCALE = index
```

The SCALES statement is used to specify the magnification levels used by the Reduce and Enlarge commands. The keyword is followed by 11 decimal integers separated by blanks and sorted in ascending order. The 1:1 scale level (at which externally created pictures appear in their original size) is 14. Enlargements are specified by smaller numbers (e.g. 7 gives 200%) and reductions by larger numbers. The default values are:

```
SCALES = 4 7 10 14 18 24 28 42 63 98 140
NORMALSCALE = 3
```

The NORMALSCALE statement is used to specify which of the scale steps specified in the SCALES line will be used as the "Normal Size" setting. The index must have a value in the range of 1 to 11.

Pin Spacing

```
PINSPACE = n;
```

The PINSPACE keyword is used to specify the spacing between adjacent pins when breakout symbols are created. This can also serve as the default for symbols created by other tools. The value must be a single decimal integer which is used as a multiple of the standard grid space of 5 dots. The default value is 2.

Breakout Parameters

```
BREAKOUT = dth dtv;
```

The BREAKOUT keyword lets you control the creation of bus breakout symbols generated by the program. This does not affect any breakouts in existing files as these symbols are already created and stored with the file.

The BREAKOUT keyword is followed by two numbers for the following parameters:

dth the horizontal offset (in screen dots at 100% scaling) for placement of text names on a breakout.

dtv the vertical offset (in screen dots at 100% scaling) for placement of text names on a breakout.

Disabling "Loose End" Markers on Signal Lines

The NOLOOSEENDS Ini file keyword disables the cross-markers that are normally displayed on the screen at the ends of unconnected line segments. The format of the command is:

```
NOLOOSEENDS = off
```

Undo Levels

The UNDO Ini file keyword indicates the number of levels of undo which should be maintained. A value of zero means that there is no undo.

```
UNDO = n
```

Fine-Tuning Pin Number Text Display

The PINTEXT keyword allows you to adjust the display position of pin numbers on devices. The format of this keyword is as follows:

```
PINTEXT = dth dtv
```

"dth" and "dtv" are positive or negative integer offsets which adjust the text position by the given amount in screen dots at Normal Size screen scale. All devices in the design are equally affected.

> IMPORTANT: This adjustment should not be required in normal use and should be used with caution. No checking is done on the range of these settings.

NOTE: Changing these numbers in the Ini file will not automatically recalculate the positions of pin numbers in existing designs. You can force a recalculate by using the Design Preferences command to change the pin text font or size, then change it back to the original setting.

Internal Error Checking

```
NOERRORCHECK = off
MSGCHECK = on
```

The first statement disables the internal error checking described elsewhere in the Appendix. The second statement increases what is checked.

NOTE: This should only be done in consultation with our Technical Support department as it will prevent warnings from being issued for internal or unusual system errors. In many cases these errors can be easily recovered if corrected immediately but may cause data corruption if left undetected.

Section [Libraries]

Library Folder

```
FOLDER = directory_path
```

This specifies the folder/directory that will contain the libraries to be specified in following LIBRARY statements. This statement can be omitted if the libraries are located in the same directory as the DesignWorks executable, or if you prefer to specify a complete file "library_path" in each library statement.

Single Library

```
LIBRARY = library_path
```

This specifies a single library to open. The "library_path" can be simply the name of the library if the library is in the current directory, or a relative path to the library, or a fully qualified path from the root. For example:

```
LIBRARY = Lib1.clf
LIBRARY = lib\74LS00.clf
LIBRARY = \mylibs\blocks\controls.clf
```

All Libraries in a Folder

```
LIBRARYFOLDER = directory_path
```

This names a folder/directory to be searched for libraries. All libraries in this folder will be opened. Folders inside this folder are not checked. The format of the folder name is the same as for the FOLDER keyword above.

Section [DevEditor]

Default Font

```
Font = "font_name" font_size [BOLD ITALIC]
```

This statement specifies the default font for text items appearing in a DevEditor document.. "font_name" is the name of a TrueType font, only TrueType fonts are supported. "font_size" is the point size to use. There are two optional style keywords which may be applied, "BOLD" and "ITALIC".

Grid Settings

```
GridColor =            RED, GREEN, BLUE, CYAN, MAGENTA, 
YELLOW
           BLACK, DKGRAY, GRAY, LTGRAY, or WHITE
GridSize = grid
SnapSize = snap
PinSnapSize = pinsnap
```

The GridColor statement specifies what color to use when drawing the DevEditors Grid. GridSize, SnapSize and PinSnapSize are all expressed as a 5 pixel multiplier. GridSize specifies the number of 5 pixel spaces between displayed grid lines. SnapSize sets the grid snap for all graphical objects, except pins, and PinSnapSize sets the snap used when positioning pins.

Section [Report]

Default Form File

```
DefaultFormFile=rptforms\spice_f.rfm
```

The DefaultFormFile statement specifies what report form should be selected by default when the report generator is first selected.

Section [System]

MEDA Modules Directory

```
ToolFolder = c:\dw\medatools
```

This statement defines where to look for the external code MEDA Modules. If this value is not specified then MEDA Modules are loaded from the tools subdirectory under wherever the DesignWorks executable is located.

Default System Font

```
Font = "font_name" font_size [BOLD ITALIC]
```

This statement specifies the default font which the DesignWorks system will use when no other font has been specified. Certain MEDA modules may by default display text using this default font. If no font is specified an attempt is made to use a Courier type face. If no font size is specified then 10 point is used.

"font_name" is the name of a TrueType font, only TrueType fonts are supported. "font_size" is the point size to use. There are two optional style keywords which may be applied, "BOLD" and "ITALIC".

```
Example:
Font = "Courier New" 10 Bold
```

Scale Printer Lines

```
ScalePrinterLines = None, All, or OverOnePixel
```

This allows the user to specify whether lines are to be scaled when printed.

"None" is the default, it indicates that no scaling will occur. With this setting a line’s width is printed with the same number of pixels as on the screen. When printing to high resolution printer, ie: >= 300 dots per inch, this will cause thick screen lines (busses) to be reproduced as printed lines which do not appear to be much thicker then thin screen lines (signals). This setting is most useful when printing to dot-matrix printers where the printer’s resolution is similar to the screens resolution.

"All" indicates that every lines will be scaled so that its printed width appears the same as on the screen.

"OverOnePixel" indicates that lines that have a screen width of greater then 1 pixel will be scaled when printed. The result is signal lines being drawn very finely, but busses appearing as thick lines.

Section [System Font Translations]

```
Old_Font_Name = Replacement_Font_Name
```

Font translations are used when the fonts imbedded in a file are not available on the current platform. This section allows the user to define which fonts (available on the current platform) are to be used instead of the fonts specified. The replacement font must be a TrueType font.

Each line in this section specifies a font mapping. For example:

```
Bookman = Courier New
Times = Times New Roman
```

specifies that anytime the font Bookman is displayed then Courier New should be used as its replacement, and when Times is displayed or requested that Times New Roman should be used.

Appendix E - Internal Error Detection

DesignWorks makes use of a number of complex internal data structures in order to maintain an up-to-date image of your circuit at all times. To assist in detecting problems due to hardware failures, program errors or operating system errors, a code module has been added which checks these structures for consistency. This is done in the "background" while the program is idling and should normally be invisible to the user. Should a problem be detected a warning box will be displayed:

In the warning box the "State" value is a code that specifies the type of problem detected, and the "Address" value is the memory address of the object that was in error. It is beyond the scope of this manual to discuss the meaning of all possible error codes and an estimate of their severity. In general, a State < 100 indicates a structural problem that is likely to cause a serious program malfunction if you proceed with editing. The warning box will only appear once if this type of error is detected, even if other errors occur later. The error detection mechanism is reset when all design files are closed. States > 100 indicate unexpected situations detected in connection with some specific function and may or may not be serious. If the error disappears after a later check (due to an offending object being deleted, for example) you will be notified. This may occur if you delete the corrupted part of the circuit or if some other internal check succeeds in correcting the problem.

If you see this box in the course of normal program operation, then save your design file immediately under a different name (so as not to wipe out your last good backup). Quitting the program and rereading the saved file may result in the problem being corrected. If you can isolate the problem down to one specific object, then try deleting that object and recreating it. In any case, please contact our Technical Support department and provide as much information as you can about the situation which created the problem. We will help you in any way we can to recover any lost data.

Error checking can be disabled using the NOERRORCHECK setup file option, described elsewhere in the Appendix. This should not be used under normal circumstances.

Credits

DesignWorks, LogicWorks, and MEDA are trademarks of Capilano Computing Systems Ltd.

DesignWorks was designed by Chris Dewhurst and coded by Chris Dewhurst, Scott Farquharson, Don Gamble, Geoff Harris, Addam Smith, David Taylor and Neil MacKenzie of Capilano Computing Systems Ltd.

This manual was written by Chris Dewhurst and Don Gamble using Microsoft Word (a trademark of Microsoft Corporation of Bellevue, Washington).

DesignWorks is written using Borland C++ 3.0 of Scotts Valley, California.

DesignWorks is implimented using the XVT Portability Toolkit. XVT is a trademark of XVT Software Inc. of Boulder, Colorado.

Windows is a trademark of Microsoft Corporation of Bellevue, Washington.

Other trademarks used in this manual are property of their respective holders.

Technical Support

Any problems or questions can be referred to:

Technical Support,
Capilano Computing Systems Ltd.,
360 Edworthy Way, Unit 206,
New Westminster, B. C.
Canada, V3L 5G2

Voice (604) 522-6200
Fax (604) 522-3972
email: tech@capilano.com
WWW: www.capilano.com

Index

Page numbers in **BOLD** face indicate major references to a topic.