

# **EMTP<sup>®</sup> User Manual**

## **Version 4.4**

Jean Mahseredjian and Chris Dewhurst, Monday, January 15, 2024

## • Disclaimer

Information in this document is subject to change without notice.

No part of this manual may be copied or reproduced in any form or by any means without written permission from PGSTech.

PGSTech makes no representation or warranty with respect to the adequacy or accuracy of this documentation or the software which it describes. In no event will PGSTech or its direct or indirect suppliers be liable for any damages whatsoever including, but not limited to, direct, indirect, incidental, or consequential damages of any character including, without limitation, loss of business profits, data, business information, or any and all other commercial damages or losses, or for any damages in excess of the list price for the license to the software and documentation.

## • Copyrights

- EMTP® and EMTPWorks: © Copyright 2016-2023 Hydro-Québec, EDF and RTE.
- DesignWorks: © Copyright 1985-2011 Capilano Computing Systems Ltd., 2011-2023 by Flying Objects Software Inc.
- ScopeView: © Copyright 2010-2023 Hydro-Québec

[www.emtp.com](http://www.emtp.com)



## Table of Contents

1	Welcome to EMTP® .....	1
1.1	An intuitive and user-friendly Graphical User Interface (GUI) .....	1
1.2	A powerful new engine .....	1
2	New Users.....	3
3	Online Documentation.....	3
4	Other Documentation .....	3
5	Tutorials .....	4
5.1	Tutorial 1: Creating a simple schematic .....	4
5.1.1	Start the program and create your first design .....	4
5.1.2	Wire editing .....	5
5.1.3	Setting device data and running a simulation .....	5
5.2	Tutorial 2: Signal Connections.....	6
5.2.1	Setting the Line Type .....	6
5.2.2	Drawing a 3-phase bus .....	7
5.2.3	Splitting out Single Phases .....	7
5.2.4	Using Bundles .....	8
5.3	Tutorial 3: Creating a Subcircuit Bottom Up .....	9
5.3.1	Creating the Subcircuit.....	9
5.3.2	Opening the Subcircuit.....	11
5.3.3	Modifying the Block Symbol.....	12
5.3.4	Saving a Subcircuit Device to a Library .....	13
5.3.5	Creating a subcircuit using Port Connectors .....	14
5.4	Tutorial 4: Creating a Subcircuit Top Down.....	14
5.4.1	Creating a Block Symbol.....	14
5.4.2	Checking the Symbol .....	15
5.4.3	Saving the Symbol .....	15
5.4.4	Placing the Symbol in a Circuit .....	16
5.4.5	Creating the Subcircuit.....	16
5.4.6	Closing and Verifying the Subcircuit .....	17
5.4.7	Making a Unique Subcircuit .....	17
5.5	Tutorial 5: Advanced Subcircuit Topics .....	19
5.5.1	Creating a Bundle Port Connector .....	19
5.5.2	Connecting to the Bundle.....	19
5.5.3	Adding Circuit Elements.....	20
5.5.4	Creating the Subcircuit Block.....	20
5.5.5	Verifying the Subcircuit Block .....	21
5.5.6	Modifying Bundle Connections on a Subcircuit Block.....	21
5.5.7	Modifying the Symbol for a Block on a Schematic.....	23
5.5.8	Adding Pins to an Existing Block Symbol .....	23
5.5.9	Updating the Symbol on the Schematic.....	23

## Table of Contents

5.5.10	Updating the Port Interface.....	23
5.6	Changing phases in the port interface.....	24
5.7	Changing the signal type in the port interface.....	26
6	Basic Schematic Editing.....	29
6.1	Background information.....	29
6.1.1	Design Structure.....	29
6.1.2	What is a Circuit.....	29
6.1.3	Types of Objects in a Circuit.....	30
6.1.4	How Design Templates Work.....	30
6.1.5	EMTPWorks File Types.....	30
6.2	Top menus.....	31
6.2.1	File Ribbon.....	31
6.2.2	Home Ribbon.....	31
6.2.3	Options Ribbon.....	32
6.2.4	View Ribbon.....	33
6.2.5	Simulate Ribbon.....	33
6.2.6	Design Ribbon.....	34
6.3	Basic Operations.....	35
6.3.1	Creating a New Design.....	35
6.3.2	Choosing a Template.....	35
6.3.3	Opening a Design.....	35
6.3.4	Navigating Around a Schematic Page.....	36
6.3.4.1	Zooming.....	36
6.3.4.2	Panning method.....	37
6.3.5	Opening Circuit Page Windows.....	37
6.3.6	Locating Circuit Objects with the Find Tools.....	37
6.3.7	Locating Objects Using the Browser Tool.....	37
6.3.8	Saving a Design.....	37
6.3.9	Reverting to a Saved File.....	38
6.3.10	Saving a Circuit Page in WMF or DXF Graphics Formats.....	38
6.3.11	Printing.....	39
6.3.12	Backup Procedures.....	40
6.3.12.1	Enabling Auto-Backup on Save.....	40
6.3.12.2	Enabling Timed Auto-Save.....	40
6.3.13	Closing a Design.....	40
6.3.14	Disposing of a Design.....	41
6.3.15	Exiting EMTPWorks.....	41
6.4	General Editing.....	41
6.4.1	Undo and Redo.....	41
6.4.2	Selection.....	41
6.4.2.1	Selecting Groups of Objects.....	41
6.4.2.2	Deselecting a Selected Object.....	41

6.4.2.3	Multi-page Selection .....	42
6.4.3	Clipboard Commands .....	42
6.4.3.1	Cut .....	43
6.4.3.2	Copy .....	43
6.4.3.3	Paste.....	43
6.4.4	Signal Connection Checking on Paste.....	43
6.4.5	Paste Special .....	44
6.4.6	Zooming in on Selected Objects .....	46
6.4.7	Adjusting the Position of All Objects on a Page.....	46
6.4.8	Adding, Deleting and Titling Circuit Pages .....	46
6.4.9	Duplicating One or More Objects.....	47
6.4.10	Rotation on Paste and Duplicate .....	47
6.4.11	Making a Circuit Read Only .....	47
6.4.12	Setting Design Attributes .....	48
6.4.13	Showing Circuit and Design Statistics .....	48
6.5	Placing and Editing Devices .....	49
6.5.1	Devices: introduction.....	49
6.5.2	Placing a Device from a Library .....	51
6.5.2.1	Setting Device Orientation.....	51
6.5.3	Using the Parts Palette .....	53
6.5.4	Selecting a Device .....	55
6.5.5	Duplicating an Existing Device.....	55
6.5.6	Deleting a Device .....	55
6.5.7	Moving a Device.....	55
6.5.8	Flipping and Rotating a Device .....	55
6.5.9	Displaying and Setting Device Information .....	55
6.5.10	Pins .....	56
6.5.11	Device right-click menu.....	57
6.5.12	Device Data .....	58
6.5.13	Device info .....	60
6.5.14	Entering Device Attributes .....	62
6.5.15	Device Attribute probe .....	62
6.5.16	Device Naming.....	62
6.5.16.1	Device Names .....	62
6.5.16.2	Device Naming Options .....	63
6.5.16.3	Setting the Auto-Generated Name Format.....	64
6.5.16.4	Reassign Names .....	64
6.5.16.5	Setting the Name Prefix for a Device .....	64
6.5.16.6	Specifying that a Device Should be Unnamed When Placed.....	64
6.5.16.7	Overriding Default Name visibility.....	64
6.5.16.8	Device Token Values.....	65
6.5.16.9	Device Date Stamping .....	65

## Table of Contents

6.5.16.10	Disabling Date Stamping .....	65
6.6	Devices Recognized in EMTP .....	66
6.6.1	Devices Recognized in EMTP .....	66
6.6.2	Device pin types.....	67
6.6.3	Built-in libraries.....	68
6.7	Creating and Editing Signals .....	69
6.7.1	Signals .....	69
6.7.2	Interconnecting Signals.....	69
6.7.2.1	Drawing From an Existing Line or a Device Pin .....	69
6.7.2.2	Creating an Unconnected Signal Line .....	70
6.7.2.3	Editing a Signal Line .....	70
6.7.2.4	Checking connectivity.....	70
6.7.3	Setting Signal Color .....	72
6.7.4	Selecting a Signal .....	72
6.7.5	Selecting a Pin .....	72
6.7.6	Pin right-click menu.....	72
6.7.7	Signal right-click menu .....	72
6.7.8	Signal Info .....	73
6.7.9	Signal Parameters.....	74
6.7.10	Signal and Pin Attribute probe .....	74
6.7.11	Signal Line Type .....	74
6.7.12	Connecting Signals by Name .....	76
6.7.13	Naming Signals.....	78
6.7.13.1	Reserved signal names .....	79
6.7.13.2	Signal connector devices.....	79
6.7.13.3	Entering and editing a Signal Name .....	79
6.7.13.4	Moving a Signal Name.....	80
6.7.13.5	Multiple Naming of Signals .....	80
6.7.13.6	Signal Auto-Naming Options .....	81
6.7.13.7	Using the Auto-Naming Features .....	82
6.7.14	Bundles.....	82
6.7.14.1	Creating a Bundle .....	83
6.7.14.2	Bundle signal names .....	83
6.7.14.3	Getting Bundle Information: Bus (Bundle) Info .....	87
6.7.14.4	Getting Information on Signals Inside a Bundle .....	87
6.7.14.5	Using Bundle Breakouts: New Breakout .....	87
6.7.14.6	Using Bundle Pins .....	89
6.7.14.7	Bundles and connectivity by name .....	90
6.7.14.8	Changing Bundle Pin Connections: Bundle Pin Info .....	92
6.7.14.9	Bundle Connection menu .....	93
6.7.15	Inter-page Connections .....	94
6.7.15.1	Automatic Display of Page References .....	96

6.7.15.2	Connecting Bundles Across Pages .....	98
6.7.15.3	Changing the Page Connector Symbol .....	98
6.7.15.4	Tracing Connections Through Page Connectors .....	98
6.8	Adding Text and Pictures to the Drawing .....	99
6.8.1	Using Text and Picture Objects .....	99
6.8.2	Creating a Text Block.....	99
6.8.3	Editing a Text Block .....	99
6.8.4	Background and Border Objects.....	100
6.8.5	Border Objects .....	100
6.8.6	Background Objects.....	100
6.8.7	Using Text Variables .....	100
6.8.8	System Variables .....	100
6.8.9	Attribute Variables.....	101
6.8.10	Editing Text with Variables .....	102
6.8.11	Text Frame Size with Variables .....	102
6.8.12	Text Object Options .....	102
6.8.13	Creating a Picture .....	103
6.9	Simulation options .....	105
7	Attributes .....	107
7.1	Attributes Overview .....	107
7.2	Attribute Organization .....	107
7.2.1	Attribute Definition Table.....	107
7.2.2	Predefined Fields .....	107
7.2.3	User-defined Fields .....	108
7.2.4	Primary vs. Secondary Fields .....	108
7.2.5	Attribute Limitations.....	108
7.3	Entering and Editing Attribute Data - Basic Procedure .....	109
7.3.1	Entering Data Values .....	109
7.3.2	Restoring the Default Value .....	110
7.3.3	Using the Value List Pop-up Menu .....	110
7.3.4	Displaying an Attribute on the Schematic .....	110
7.4	Entering Design Attributes.....	110
7.5	Entering Pin Attributes.....	110
7.6	Attribute text right-click menu .....	111
7.7	Rotating Attribute Text.....	111
7.8	Hide a Visible Attribute Value .....	111
7.9	Setting Attribute Text Style .....	112
7.10	Setting Visible Attribute Justification .....	113
7.11	Displaying an Attribute Value in Multiple Locations .....	114
7.12	Showing the Field Name with an Attribute Value .....	114
7.13	Using Value List Fields.....	114
7.14	Using Script fields.....	115

## Table of Contents

7.15	Using Default Position Fields .....	115
7.16	Defining Attribute Fields .....	116
7.16.1	Setting Attribute Field Options .....	117
7.17	Duplicate, Merge and Delete options .....	119
7.18	Globally Duplicating Attribute Data .....	119
7.19	Merging Two Existing Attribute Fields .....	120
7.20	Deleting Attribute Field Definitions .....	120
7.21	Temporarily Displaying Attributes .....	121
7.22	Permanently Showing Data Throughout a Design .....	121
7.23	Merging Dissimilar Designs .....	121
7.24	Importing Attribute Definitions .....	121
7.25	Pasting from the Clipboard or Placing a Library Device .....	122
7.26	Predefined Attribute Fields .....	122
7.27	Property Browser .....	134
7.27.1	Browser for attributes .....	134
7.27.2	Changing Attribute Values .....	136
7.27.3	Saving and Printing Data in a Browser Window .....	137
7.27.4	Browser Clipboard Data Format .....	137
7.27.5	Showing Objects in the Schematic .....	137
7.27.6	Sorting Displayed Objects .....	137
7.27.7	Adjusting the Spreadsheet .....	138
8	Subcircuits and Hierarchy .....	139
8.1	Subcircuits and Hierarchy .....	139
8.1.1	What is Hierarchy? .....	139
8.2	Subcircuit Uniqueness and Encapsulation .....	140
8.2.1	Make Unique Type .....	141
8.2.2	PartTemp attribute .....	143
8.2.3	Subcircuits within subcircuits .....	144
8.3	Subcircuit Properties .....	145
8.4	Subcircuit creation methods .....	148
8.5	Saving Part Type to Library .....	149
8.6	Attaching Subcircuit .....	149
8.7	Detaching Subcircuit .....	150
8.8	Discarding a Subcircuit .....	151
8.9	Locking and Unlocking Subcircuits .....	151
8.10	Subcircuit decoding .....	151
8.11	Printing Hierarchical Designs .....	154
8.11.1	Determining Print Page Order .....	154
8.11.2	Design Preferences: Printing .....	154
8.11.3	Printing Sequential Page Numbers in a Hierarchical Design .....	155
8.12	Masking .....	155
8.12.1	Masking .....	155

8.12.2	Unmasking .....	156
9	Device Symbols and Libraries .....	157
9.1	About Device Symbols .....	157
9.2	Terminology Note .....	157
9.3	Working with Symbol Libraries .....	158
9.4	Designs and Libraries .....	158
9.5	Creating a New Library .....	158
9.6	Manually Opening a Library .....	158
9.7	Automatically Opening Libraries at Startup .....	159
9.8	Manually Closing a Library .....	159
9.9	Lib Maintenance .....	159
9.10	Update from Lib .....	160
9.11	Internal Circuit Options .....	163
9.12	Save to Lib .....	164
9.13	Circuit to Library .....	165
9.14	Library to Circuit .....	166
10	Symbol Editor .....	167
10.1	About the Symbol Editor .....	167
10.2	Using the Clipboard in Device Symbol Editor .....	167
10.3	Editing existing Parts (Devices) .....	167
10.3.1	Editing an existing part in a library .....	167
10.3.2	Editing an existing part in the design .....	168
10.4	Modifying subcircuits .....	168
10.5	Creating a new EMTP Part .....	168
10.6	Editor functions .....	168
10.7	Symbol Gallery .....	169
10.7.1	Using elements from the Symbol Gallery .....	169
10.7.2	Adding elements in the Symbol Gallery .....	169
10.8	Placing objects .....	170
10.9	Symbol Date Stamping .....	170
10.10	Setting Grids .....	171
10.11	Line and Fill color settings .....	172
10.12	Working with Pins .....	172
10.13	Pin Function .....	172
10.14	Deleting Pins .....	173
10.15	Adding sequential pin names .....	173
10.16	Placing a Bundle Pin .....	174
10.17	Pin right-click menu .....	174
10.18	Setting Pin Attributes .....	175
10.19	Setting Part Attributes .....	175
10.20	Attribute Field Definitions in Symbols and Designs .....	176
10.21	Auto Create Symbol .....	177

## Table of Contents

10.22	Subcircuit and Part Type command.....	178
10.23	Opening the Subcircuit Associated with a Symbol .....	179
10.24	Creating Pseudo-Device Symbols .....	179
10.24.1	Creating Pseudo-Device Symbols.....	179
10.24.2	Breakout primitive type .....	180
10.24.3	Creating a Signal Connector.....	180
10.24.4	Creating a Port Connector .....	181
10.24.5	Creating a Page Connector .....	184
10.25	Password protection.....	184
11	Templates .....	185
11.1	Design Templates .....	185
11.2	Creating Design Templates.....	185
11.2.1	Creating a Title Block.....	185
11.3	Sheet Templates .....	186
11.3.1	Setting Sheet Sizes and Borders.....	186
11.3.2	Importing Sheet Settings from another Design, Page or Template.....	187
11.3.3	Custom Sheet Size using Sheet Size Wizard.....	187
11.3.4	Border Wizard .....	190
11.3.5	Creating Custom Sheet Border Graphics .....	191
11.3.6	Creating Multipage Templates.....	192
11.3.7	Sheet Border Setup for Multi-Page Designs.....	192
12	Initialization file EMTP.INI.....	193
12.1	INI File General Format.....	193
12.2	Specifying File and Folder (Directory) Names .....	193
12.2.1	Use Environment and Registry Variables .....	193
12.3	[DevEditor] Section.....	194
12.4	[Drawing] section .....	195
12.5	[Export] Section .....	200
12.6	[Find] Section .....	201
12.7	[Jscript] Section .....	201
12.8	[Libraries] section .....	201
12.9	[System] Section .....	202
12.10	[TextEdit] Section .....	203
12.11	Creating Custom Menus .....	203
13	Keyboard Shortcuts .....	205
13.1	Keyboard Shortcuts.....	205
13.2	Other Key Usage.....	207
14	FAQ & Tech Notes.....	209
14.1	Installation issues .....	209
14.1.1	Can I install more than one version of EMTP on my computer? .....	209
14.2	Schematic Editing.....	209
14.2.1	Modifying a Bundle Pin on an Existing Device .....	209

14.2.2 Setting Device Color ..... 209

14.2.3 Attributes..... 210

    14.2.3.1 Making an Attribute Field Visible on All Devices ..... 210

    14.2.3.2 Preventing a Name from Being Made Visible..... 210

    14.2.3.3 Preventing a Name from Being Assigned..... 211

    14.2.3.4 Deleting Attribute Fields from a Symbol (Device)..... 211

14.3 Symbols (Devices) & Libraries ..... 211

    14.3.1 Deleting Attribute Fields from a Symbol (Device)..... 211

    14.3.2 Adding Internal Pins to a Bundle Pin ..... 212

    14.3.3 Opening Libraries Automatically ..... 213

        14.3.3.1 Placing Libraries in the standard "Libs" Folder ..... 213

        14.3.3.2 Placing a Shortcut in the "Libs" Folder ..... 213

    14.3.4 Adding an Item to the Symbol Gallery ..... 213

    14.3.5 Creating a Fixed-Phase Pin on a Symbol..... 214

14.4 Report Generation ..... 214

    14.4.1 Creating an Information Report ..... 214

14.5 Creating Scripts for the Find Tool ..... 216

    14.5.1 Find Data File Format ..... 216

14.6 Printing ..... 220

    14.6.1 Fitting a Printout to a Single Sheet ..... 220

    14.6.2 Printing in Black and White ..... 220

    14.6.3 Line Width Scaling When Printing ..... 221

    14.6.4 Notes on Editing the INI File ..... 221

14.7 Copying ..... 222

    14.7.1 Copying into Word ..... 222



# 1 Welcome to EMTP®

EMTP® is the Electromagnetic Transients Program.

EMTPWorks is the graphical user interface (GUI) of EMTP®.

EMTP® is used to study transients in large scale power systems and in electrical networks in general. It is a simulation software.

The famous EMTP software was initially developed by the Development Coordination Group (DCG) of the EMTP. The contributing DCG members were: American Electric Power corporation, CEA Technologies, CRIEPI of Japan, Électricité de France, EPRI, Hydro One Networks, Hydro-Québec, Réseau de Transport d'Électricité (France), US Bureau of Reclamation and Western Area Power Administration.

In 1998, the DCG, under the technical leadership of Jean Mahseredjian then working at Hydro-Québec (IREQ), started a major project for restructuring/rewriting the EMTP with the target being the complete rewriting of the software code using the latest software engineering techniques and the most recent and proven computational methods. The first new version of EMTP was released in 2003 and initially named EMTP-RV, with RV meaning Restructured Version. Only the original name (EMTP) without version number is currently used.

The new EMTP established a new standard in the simulation and analysis of power system transients. EMTP is a completely new code written from scratch using completely new computational methods and models.

The DCG was dismantled after the development of the restructured EMTP version. Several versions of EMTP have been released since 2003 on regular basis. The current owners of EMTP are Hydro-Québec (HQ), Électricité de France (EDF) and Réseau de Transport d'Électricité (RTE).

## 1.1 An intuitive and user-friendly Graphical User Interface (GUI)

EMTP has a powerful GUI called EMTPWorks which is also its simulation environment. It is designed to efficiently create and maintain small circuits as well as very large scale networks. EMTPWorks has an open-architecture for maximized user configurability from basic user-defined model assemblies to more advanced script based programming. EMTPWorks offers script methods for simple data management tasks to more advanced object attribute settings, library functions and network device symbol redrawing and updating. Each device is given its data web using DHTML. Device data web pages can reside on the user's computer or anywhere on the web. The script source code of all device data management functions is available to the user. EMTPWorks has automatic subcircuit creation methods with unlimited levels of hierarchy. Subcircuit masking options can accommodate user programming from simple data statements to more complex scripts and data capture panels.

## 1.2 A powerful new engine

EMTP uses a completely new approach for assembling network equations based on sparse modified-augmented-nodal analysis. A new Jacobian based nonlinear solver eliminates topological restrictions and allows solving very large scale nonlinear systems with a minimized number of iterations. In addition to a large library of electrical/electronic circuit and power system devices, EMTP can solve control systems using block-diagrams. A library of primitive devices is used to build libraries of specialized control functions. EMTP has a completely new control system solver and an optional Newton method for finding the ultimate simultaneous solution of all blocks.

## Table of Contents

Power electronics circuits can be solved using macro-models, such as ideal switches and detailed nonlinear functions. Simultaneous switching is achieved by treating the switches in an iterative loop.

In addition to its time-domain solution EMTP has a complex matrix based steady-state solution used primarily in automatic initialization of state-variables. Automatic initialization is for achieving quick harmonic steady-state in the first time-domain cycle. All devices, including the distributed parameter transmission line models and machines, can be initialized.

The steady-state solution of EMTP can be found from a load-flow solution. EMTP offers a unique and advanced multiphase load-flow solution option.

Equipped with a fully automatic and dynamic memory management, EMTP minimizes numerical limitations and topological restrictions for providing to the user an environment of unsurpassed expressive power. Although complex and powerful, it remains straightforward to harness for conducting studies at all user levels.

## **2 New Users**

New users should start with the Quick Overview document (see File>Help & Support>Help Documentation>Quick Overview of EMTP). Then it is recommended to read the Tutorials to quickly familiarize with basic program operation.

## **3 Online Documentation**

The documentation is organized into a set of generic documents for describing main functions and methods, and into a set of specific documents. An electrical network device, for example, has its own specific set of documentation and it can be accessed only from its help tab.

This online help system is divided into three sections:

- The Tutorial section will give you a quick introduction to basic operation of the software and bring you up to speed with frequently-used procedures.
- The Reference section provides detailed background information and procedures for all the commands and features of the package.
- The FAQ & Tech Notes section provides detailed answers to questions about specific application areas and problem-solving.

See the Table of Contents for direct links to these sections.

## **4 Other Documentation**

Additional documentation is provided for advanced users. It is available under the File>Help & Support menu of EMTPWorks.

## 5 Tutorials

### 5.1 Tutorial 1: Creating a simple schematic

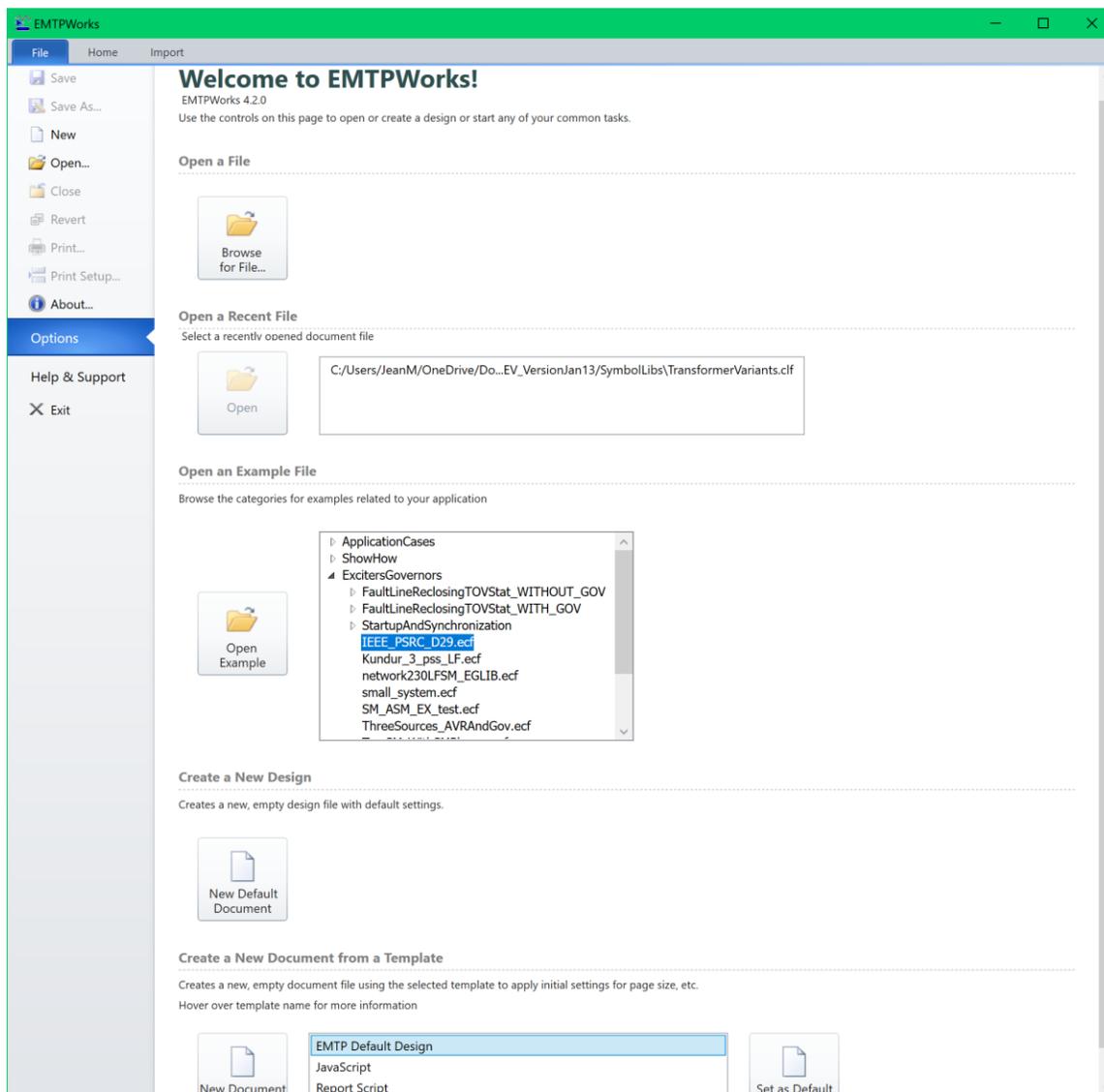
#### 5.1.1 Start the program and create your first design

- If it is not already running, double-click on the EMTWorks icon, or select the EMTWorks program in the Start menu to start the program.



EMTWorks.exe

After loading the program, the following main view appears.



The user can start by selecting the “EMTP Default Design” and clicking on the button “New Document from Template”.

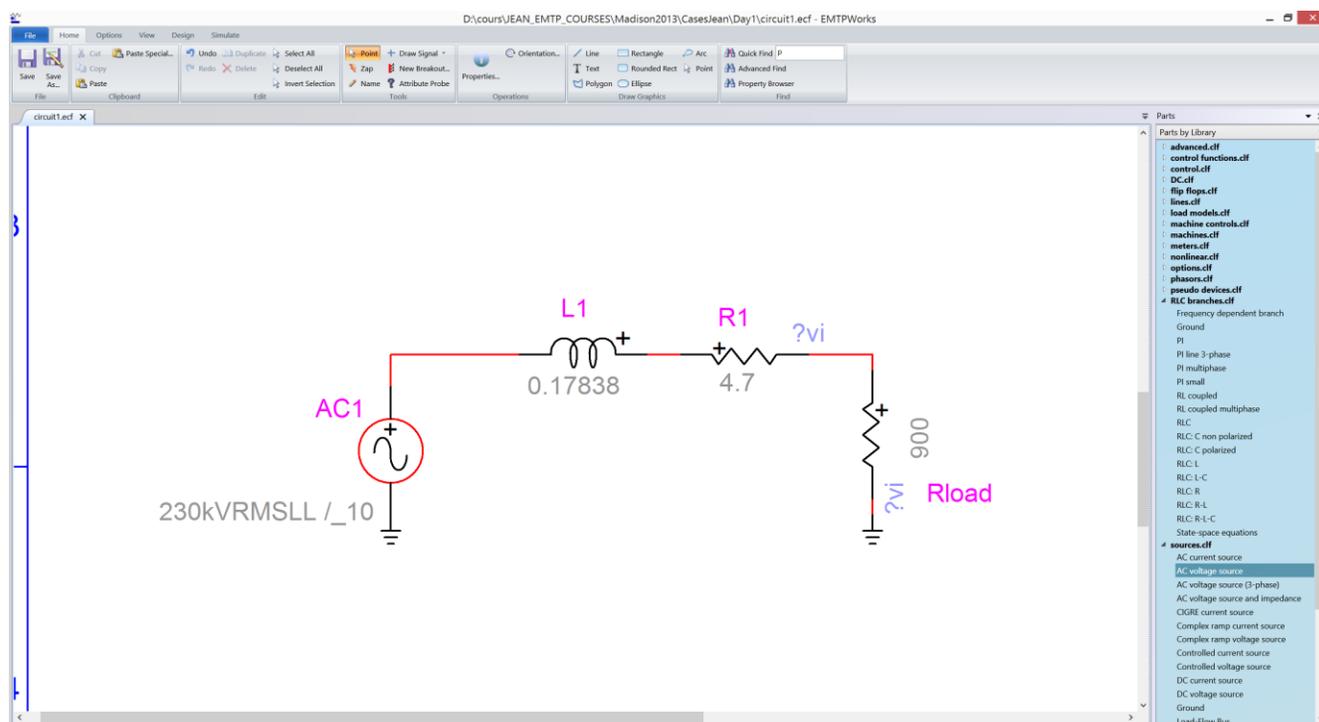
This box ("Create a New Document from Template") allows you to select a design template suitable for your application. Each selection enables a group of options most commonly used for the application type. All options can be changed later, if desired.

- ❑ Select the "EMTP" template.
- ❑ Click on the Create button or double-click on the selected template.

We have selected a template with options set up for a typical EMTP design. It is possible to create your own templates and add them to this list to customize the application to your needs.

You will now see an empty schematic drawing window ready for editing.

- ❑ The creation of the circuit is based on selecting devices from the library and connecting them with wires.



- ❑ Save the design (File>Save Design As) under the name circuit1.ecf (any name).

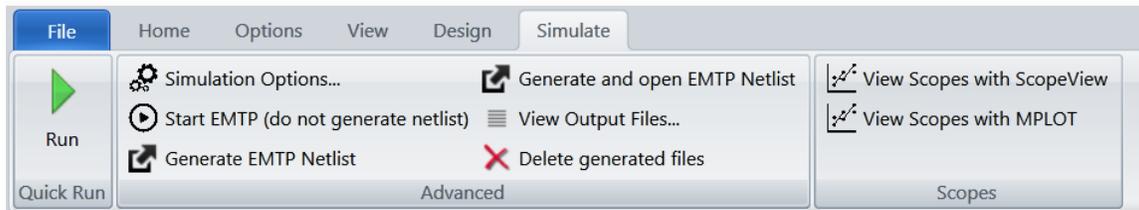
### 5.1.2 Wire editing

EMTPWorks allows you to edit wiring with some built-in "intelligent" rules. It is also feasible to delete part or a wire or the entire wire. You can right-click on a wire for various properties and options.

EMTPWorks keeps automatic internal names for all signals. The user can change any signal name, if needed. In general it is recommended to use device names and not signal names for building a design.

### 5.1.3 Setting device data and running a simulation

Entering device data is achieved by double-clicking on a given device and opening the device data web. After completing device data input it is necessary to set up simulations options before running the simulation. The simulation options can be found on the Simulate ribbon.

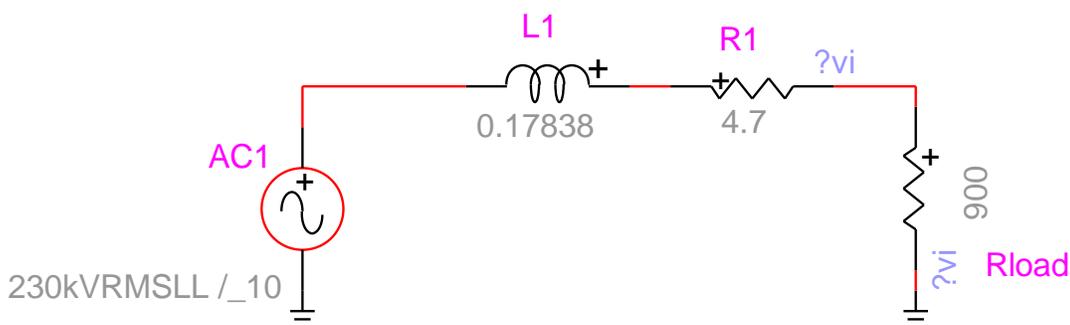


There are two separate packages for showing scopes (waveforms): ScopeView and MPLOT.

## 5.2 Tutorial 2: Signal Connections

EMTPWorks supports a number of different types of signal connections, such as control signals, single- and 3-phase power signals and bundles. As you are working with these items, the program enforces rules concerning what types of elements can be interconnected. In this tutorial section we will try out the various signal tools.

The previously created circuit had only default power signal type (General Signal):



The General Signal has a default red color.

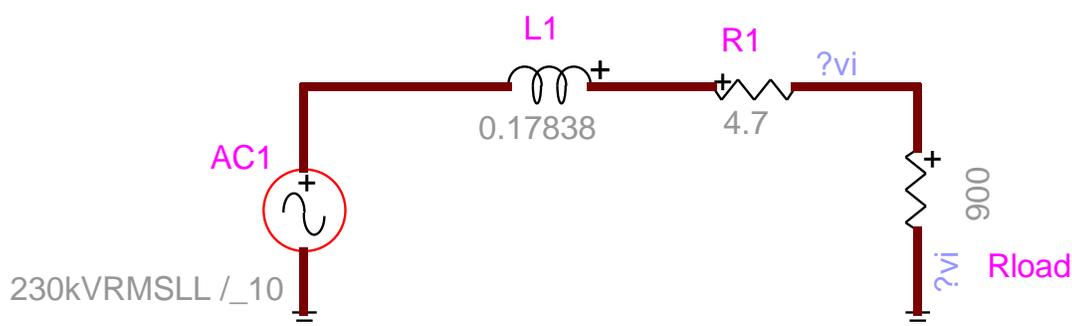
### 5.2.1 Setting the Line Type

- Click the right mouse button along any one of the signal lines, then slide down to the Line Type submenu.

Note that the item General Signal is checked. This indicates that this type of line can adapt to whatever devices are attached to it. In this case, all attached items have power pins and this line represents a single-phase power signal.

- Select the 3-Phase Signal item in the Line Type submenu. The circuit is now 3-phase. The 3-phase signal is shown in default dark red color and is thicker than the General Signal.

By default all circuit data is assumed to be balanced, but you can modify 3-phase device data by double-clicking on the device and opening its 3-phase data web.



Note that this setting not only changes the line we clicked on, but all the other lines (interconnected) in the design as well.

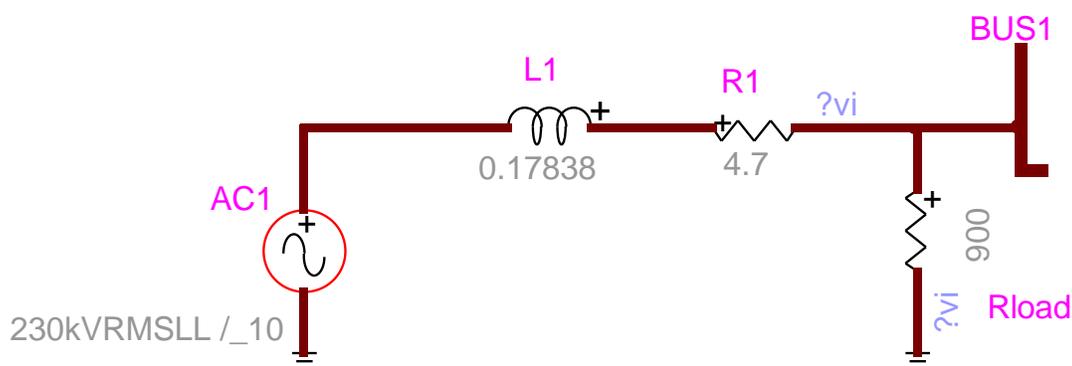
- Optional: Right-click on any signal line in the design and check its line type, verifying that all signals are now 3-phase.

Whenever the phase setting of a power signal is changed, EMTPWorks attempts to propagate this change through the design, adjusting all attached devices and signals as needed to maintain compatibility. In cases where some attached devices are not compatible with this change, a warning may be issued. It is strongly recommended to use the Home>Edit>Undo command (Ctrl+Z shortcut) to correct the problem, the simulation may encounter problems otherwise.

### 5.2.2 Drawing a 3-phase bus

- Move to the toolbar "Home>Tools>Draw Signal" and select the "Draw 3-phase Bus" tool
- Use this tool to draw a bus along one side of the existing circuit.

A "3-phase bus" is electrically the same as a "3-phase signal", but its usage in the diagram is different. A "3-phase signal" is used to simply interconnect components, while a "3-phase bus" is typically used as a connecting point for multiple 3-phase or single-phase lines.



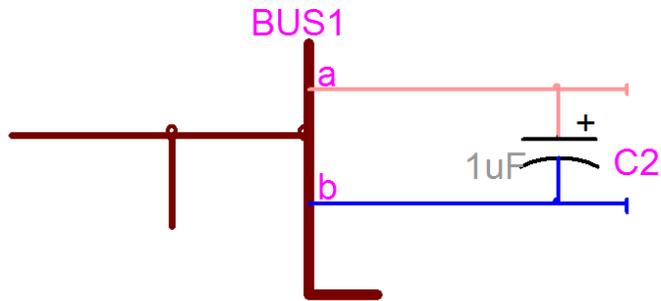
### 5.2.3 Splitting out Single Phases

EMTPWorks allows connecting to individual phases of the 3-phase bus.

- Select "Draw Phase A" tool in "Home>Tools>Draw Signal".
- Click anywhere and extend to connect to the bus or start from the bus.
- Similarly, use the "Draw Phase B" to create the phase b signal.

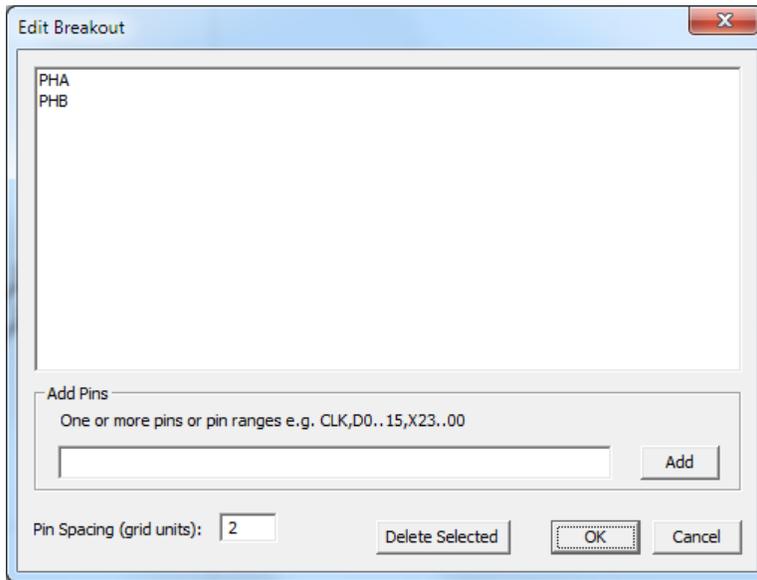
Note that the Phase a and b lines are color-coded to distinguish them. The device C2 is 1-phase.

The phase signal name is BUS1a. It is the same for the other phases.



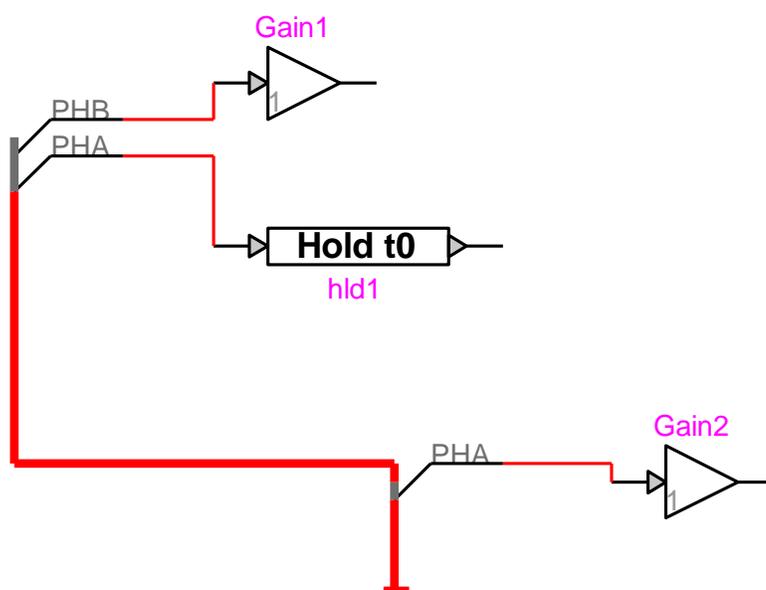
### 5.2.4 Using Bundles

- ❑ Select the New Breakout command in the “Home>Tools” menu.
- ❑ Enter two names PHA and PHB, separated by spaces.
- ❑ Click Add.
- ❑ Click OK.
- ❑ Place the resulting breakout symbol.
- ❑ Draw lines from bundle pins to connect to other pins.



A bundle is a way of grouping any collection of other signals into a single line. It can make a diagram much simpler by collecting lines into a bundle line. The signals in a bundle can be any combination of power and control signals, but a bundle cannot contain another bundle.

- ❑ Click on the bottom of the "spine" of the breakout symbol and extend a bundle line away and to the right to a clear area of the diagram.
- ❑ Right-click at the end of the bundle and select the Breakout command. The resulting box should contain a list of the existing signals in the bundle.
- ❑ Remove the name PHB from the list (Click on the name to select and push on Delete Selected) so only PHA remains in the name box, then click OK.
- ❑ Place the new breakout and attach it to the bundle.



A bundle contains the sum of all signals that are attached to it using breakouts. A given breakout can extract any or all of the signals in the bundle.

- ❑ Right-click along the line leading from the new Gain2 device to the PHA breakout pin and select the "Pin List" command.

Note that this list shows the other device connected through the PHA signal inside the bundle.

The pin list also shows the breakout connection points. This is for convenience in locating connections to the signal on the diagram, even if breakouts are not electrically significant in the circuit.

The notation after each item (e.g. "1-A3") is the grid location of the item on the drawing sheet. This can be helpful in locating items in larger designs.

- ❑ Double-click on any of the items in this list to highlight and use the "Go To" button.

### 5.3 Tutorial 3: Creating a Subcircuit Bottom Up

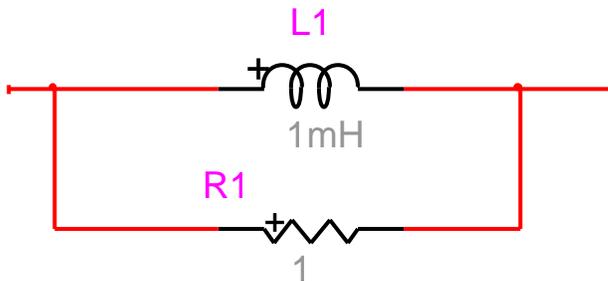
In this tutorial section we will look at how to use hierarchy to simplify the representation of complex designs. Using this feature, a symbol on a top-level circuit can itself represent a subcircuit of any desired size or complexity.

#### 5.3.1 Creating the Subcircuit

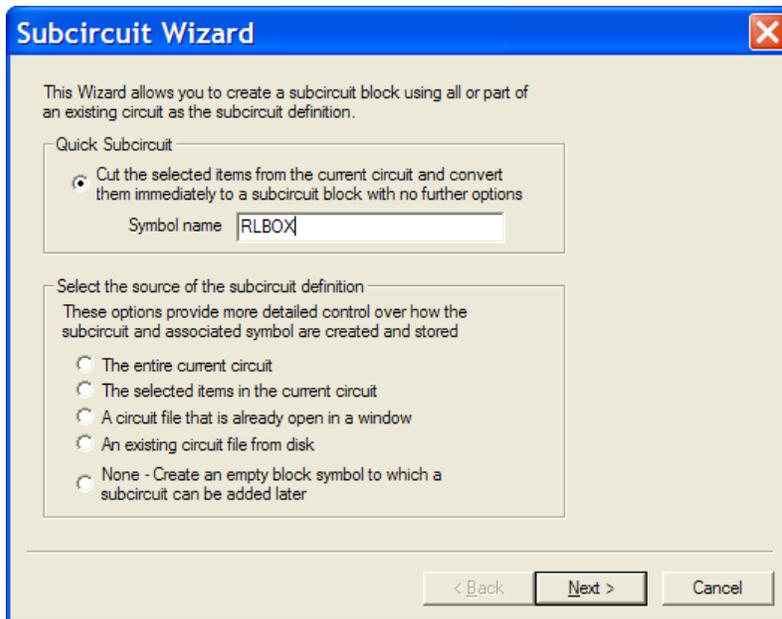
First, we will create a simple subcircuit, and then create the symbol that will be used to represent it in higher-level circuits.

## Tutorials

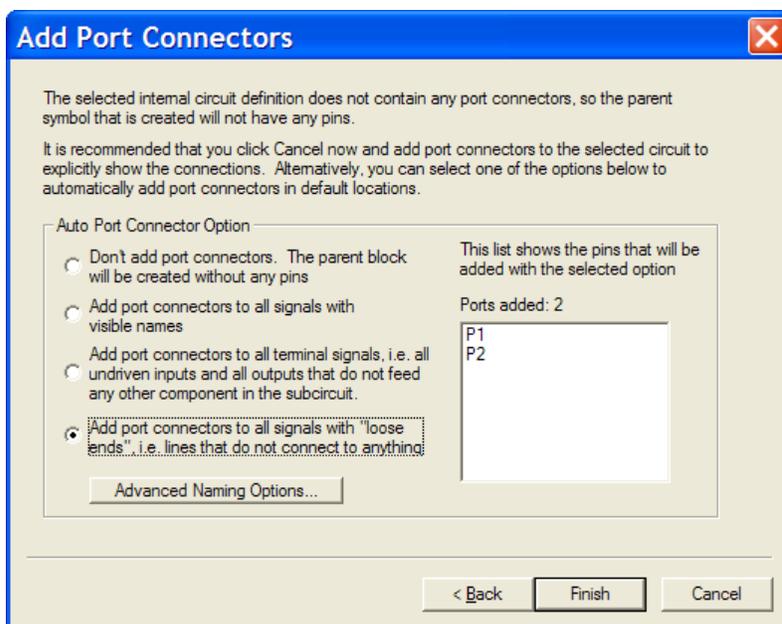
- ❑ Using the methods covered in previous tutorials, create the simple circuit shown below. You will need one each of the device types R and L in the RLC Branches library.



- ❑ Select the entire interconnected circuit. You can select the entire interconnected circuit section using:
  - the mouse pointer and selecting with a bounding box
  - hold the Ctrl button and double-click on any one of the devices, it will select the entire (physically) interconnected circuit
- ❑ Select the “Create Subcircuit Block” command in the “Options>Subcircuit” menu.



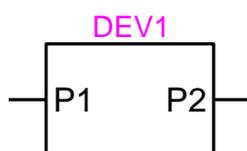
- ❑ Select the Quick Subcircuit option.
- ❑ Enter a name for the symbol, for example "RLBOX".
- ❑ Click the Next button on this panel.



Note that there are some restrictions on names and that the Next button will be disabled if the name contains blanks or other unacceptable characters.

A subcircuit needs to have special symbols called "Port Connectors" which indicate which signals in the subcircuit are available for connection through the parent symbol. Since the circuit we created does not contain any "Port Connectors", the program is offering to add them automatically.

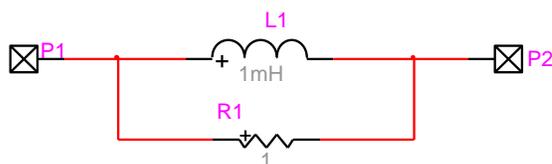
- ❑ Select the last option "Add port connectors to all signals with loose ends". The list at the right should show two entries.
- ❑ Click the Finish button.



The program has replaced our original circuit with a single symbol representing the circuit we created. Since we provided no hints about names for the ports, it has generated default names. Numerous options are available for editing and modifying the symbol and the internal circuit.

### 5.3.2 Opening the Subcircuit

- ❑ Double-click on the block symbol to open the subcircuit or select the subcircuit (block symbol) and use Ctrl+Shift+I ("Options>Subcircuit>Push Into").

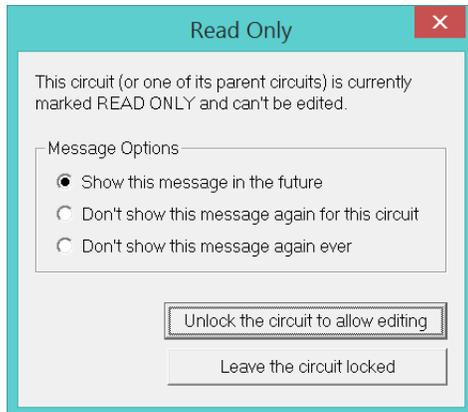


## Tutorials

You will see that the subcircuit contains our original design with port connectors added to the two signals. This works well for simple circuits, but for larger subcircuits, we recommend adding "Port Connectors" manually before coming to this step to ensure that they are named and placed appropriately.

As a default procedure all freshly created subcircuits are locked, you need to first unlock the subcircuit before making any changes to its contents.

- ❑ To unlock a subcircuit (if needed to make changes), right-click in an empty area of the subcircuit and select "Circuit Info". Then uncheck the checkbox "Read Only". You can also try to move a device (use the arrow keys or the mouse) which will trigger a menu asking permission to unlock the subcircuit.



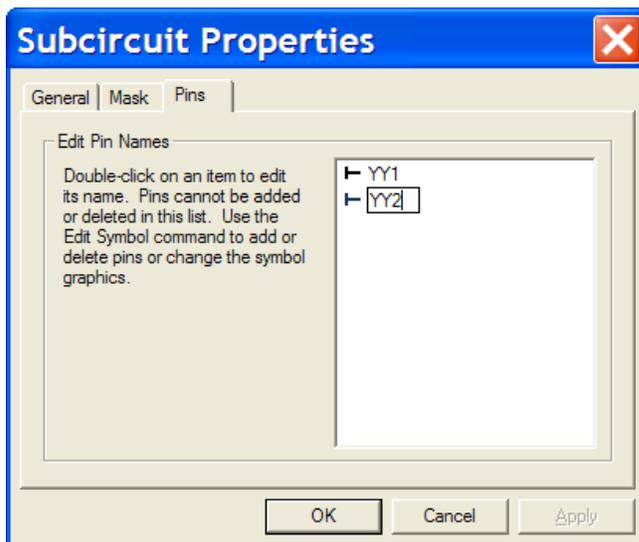
- ❑ Right-click in an empty area of the subcircuit and select the "Options>Subcircuit>Pop Up" (Ctrl+Shift+U) command. This will close the subcircuit and return to the parent circuit.

### 5.3.3 Modifying the Block Symbol

- ❑ Right-click on the block symbol and select the Subcircuit Info command.

This command allows us to change subcircuit settings and make limited changes to the symbol.

- ❑ Select the Pins tab.
- ❑ Double-click on the pin name P1 and enter a new name, for example YY1.
- ❑ Modify the other pin name as well, if desired.



NOTE: The size of the symbol and the positions of the text are not modified by this operation, so changing from a short name to a long name may not fit well on the symbol. Making manual changes to the symbol is covered later.

- ❑ Click OK to complete the changes.

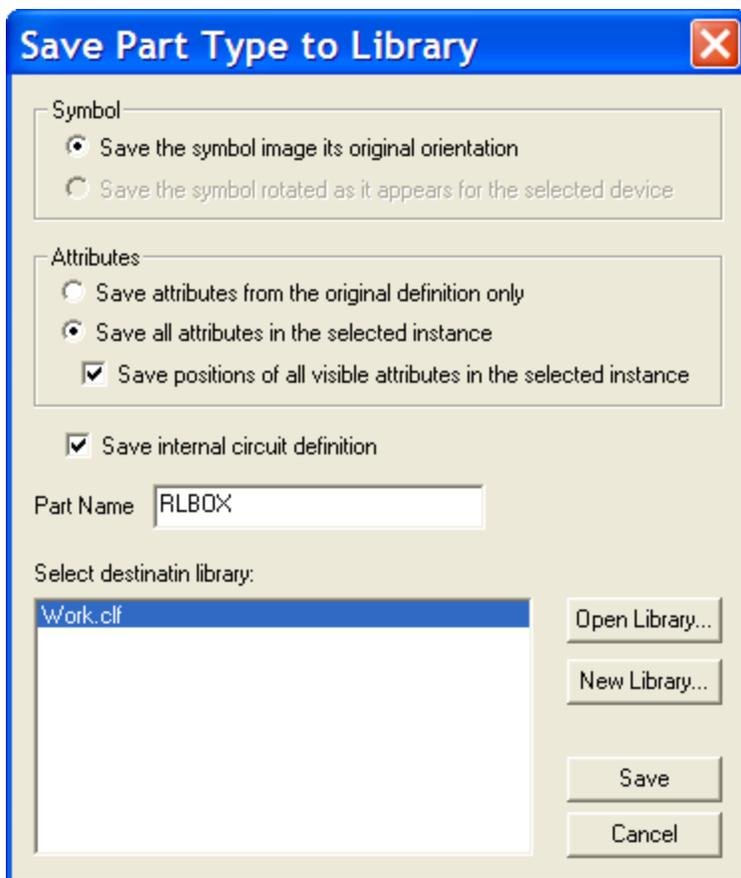


- ❑ Double-click on the symbol again to open the subcircuit and observe that the names on the port connectors have been changed to match the pin name changes on the parent.

The association between the port connectors in the subcircuit and the pins on the parent symbol is made by name. It is essential for each pin on the symbol to match exactly one port connector. In this case, the program is making both changes for us, but if you manually change either the circuit or the symbol, it is your responsibility to maintain this match.

#### 5.3.4 Saving a Subcircuit Device to a Library

- ❑ Close the subcircuit so that we are again looking at the top-level design.
- ❑ Click on the block symbol to select it.
- ❑ Go to the "Options" menu, "Part Type" submenu, and select the command "Save to Lib".
- ❑ If you already have a work library open, you can select it in the list. Otherwise, use the "New Library" button to create a temporary library in a convenient location. You can call it "Work" or any other desired name. Select the new library in the list. It is noticed that the builtin EMTPWorks libraries are locked.
- ❑ Ensure that the options "Save Symbol Image in its Original Orientation" and "Save Internal Circuit Definition" are selected. The other options are not critical.



- ❑ Click the Save button.

We have now saved the symbol to the library complete with the subcircuit definition. This entire entity can now be freely used in other designs. Note that the data stored in the library is complete of itself and there is no reference from the library back to this original design. You can modify or discard this design without affecting the library version or any other designs that use it.

### 5.3.5 Creating a subcircuit using Port Connectors

Instead of letting the program place the port connectors automatically, you can place them manually. This provides more flexibility and is especially useful for large circuits.

The port connectors are selected from "Options>Subcircuit>New Port Connector". Another approach is to select Port Connectors from the "Pseudo Devices.clf" library.

## 5.4 Tutorial 4: Creating a Subcircuit Top Down

In this tutorial, we will work from the top level of the design, first creating the symbol and then later describing its internal definition.

### 5.4.1 Creating a Block Symbol

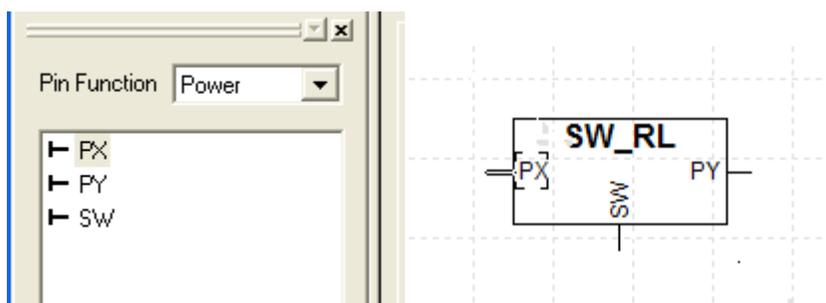
- ❑ Select the New command in the File menu.
- ❑ Double-click on the Device Symbol item.
- ❑ In the "Symbol Editor>Operations" menu, select the "Auto Create Symbol" command.
- ❑ In the Left Pins box, type the text "|P PX". The first character is the vertical bar and is used to force a specific pin type, in this case Power.

- ❑ In the Right Pins box, type "|P PY".
- ❑ In the Bottom Pins box, type "|I SW". This is a vertical bar, followed by the capital letter "I" and indicates an input pin.
- ❑ Enter "SW\_RL" for the part name.
- ❑ Click the Generate Symbol button.

#### 5.4.2 Checking the Symbol

You should now see a square symbol with the requested 3 pins.

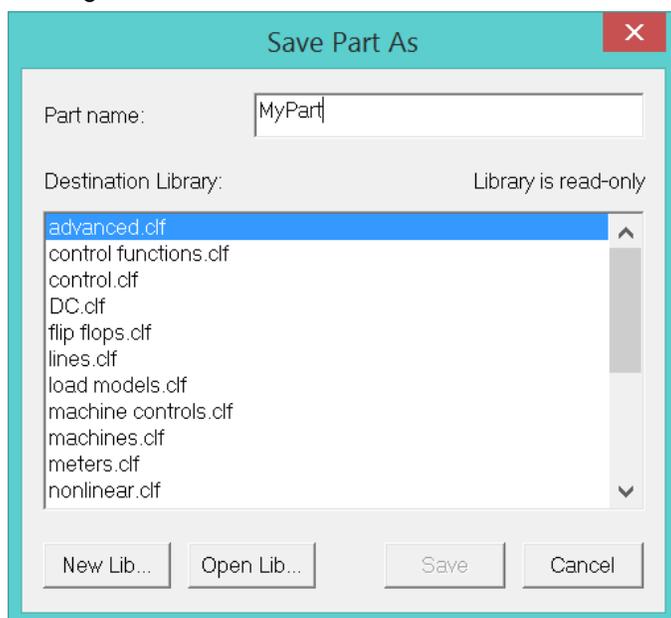
- ❑ Click on pin PX in the list and verify that it's function is Power.
- ❑ Click on PY and verify that it is also a power pin, then click SW and verify that it is an input pin.



The correct setting of the pin functions is essential as EMTPWorks checks to make sure only compatible connections are being made as you draw your diagram.

#### 5.4.3 Saving the Symbol

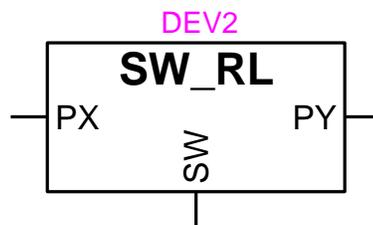
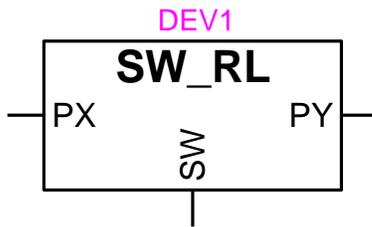
- ❑ Use the "Symbol Editor>Save As" command to save the symbol to a suitable symbol library. Do not save your own symbols to the standard libraries included with the package. You can use the "New Lib" or "Open Lib" buttons as needed to create a new library or locate an existing one.



- ❑ It is important to fix the Part attribute for this new device. Use the "Operations>Part Attributes" command (of the Symbol Editor), select the Part attribute and enter the name SW\_RL.
- ❑ Click on (X) at the top right corner of the window tab to close the Symbol Editor.

#### 5.4.4 Placing the Symbol in a Circuit

- ❑ If there is not already a blank design open, create one now, using the New command in the File menu.
- ❑ Locate the new item "SW\_RL" in the library parts list and drag one onto the diagram.
- ❑ Repeat the above step and place a second copy of the same part on the drawing.



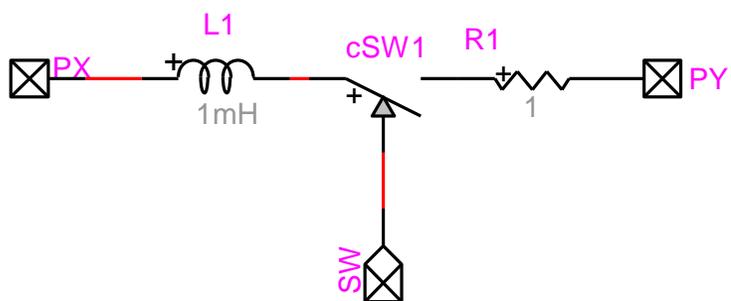
#### 5.4.5 Creating the Subcircuit

- ❑ Double-click on either one of the block symbols. You should see a box asking if you wish to create a subcircuit.
- ❑ Click OK



The program has now created a subcircuit automatically, with port connectors added matching the parent symbol. We need now only add the circuit elements needed to make the internal circuit definition.

- ❑ Draw the circuit shown (or any other circuit you like) using part types R, L and Controlled Switch.



At this stage, the actual internal circuit we create is not important. Note, however, that the pin type of the port connectors is complementary to the pin type of the parent symbol. This will restrict what type of connections you can make to each port. For example, only control input pins can be connected to the SW port and only power connections can be made to the PX and PY ports.

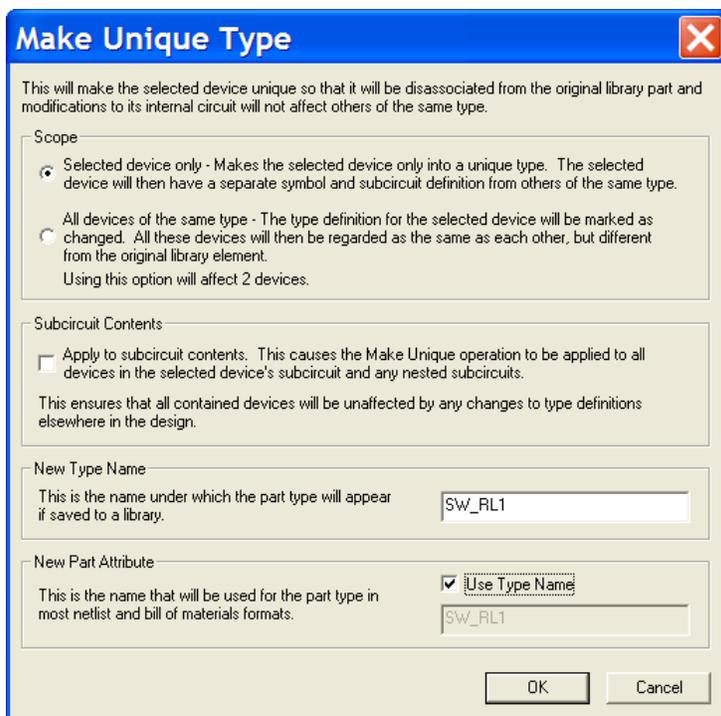
#### 5.4.6 Closing and Verifying the Subcircuit

- Close the subcircuit window to return to the top-level circuit.
- Double-click on each of the block symbols in turn to verify that they both refer to the same subcircuit definition.
- Make some visible modification to the subcircuit and note that it affects the subcircuit under both copies of the device.

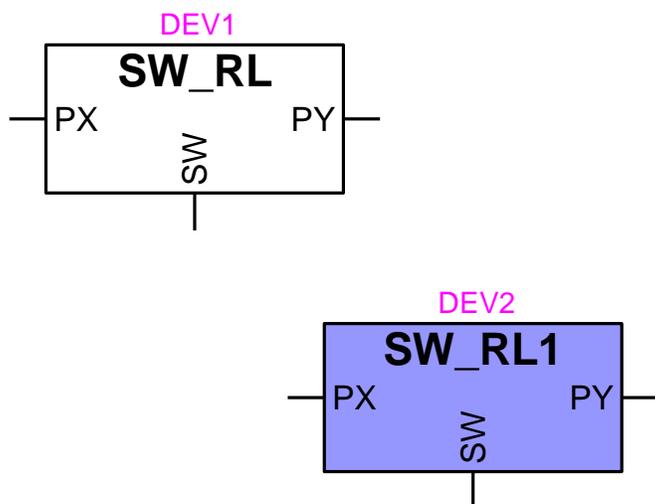
Only one copy of the subcircuit definition is stored for all instances of the same device type. Modifying the subcircuit anywhere in this design affects all instances.

#### 5.4.7 Making a Unique Subcircuit

- Make sure the subcircuit window is closed.
- Click on either one of the block symbols so that exactly one of them is selected.
- Go to the "Options ribbon", move to "Part Type" and select the "Make Unique Type" command.
- Under the Scope options, select the "Selected Devices Only" option. Leave the other options as their default values.
- Click the OK button.



We have now "separated" the selected device so that it now has a different definition than the other device. It is strongly recommended to select a significant "Type Name" and to make visual changes to the new device symbol using the "Edit Symbol" right-click menu. In this example we changed the name and the fill color of the new device.



- ❑ Double-click on either device to open it and make a visible modification to the internal circuit.
- ❑ Return to the top-level circuit and open the other device and verify that it has NOT changed, as it is now a distinct device type.

EMTPWorks uses an internal automatic device attribute PartTemp to distinguish subcircuits. It is however recommended that you select significant Part names (Part attribute). The Part attribute can be also made visible from the device right-click "Attributes" menu.

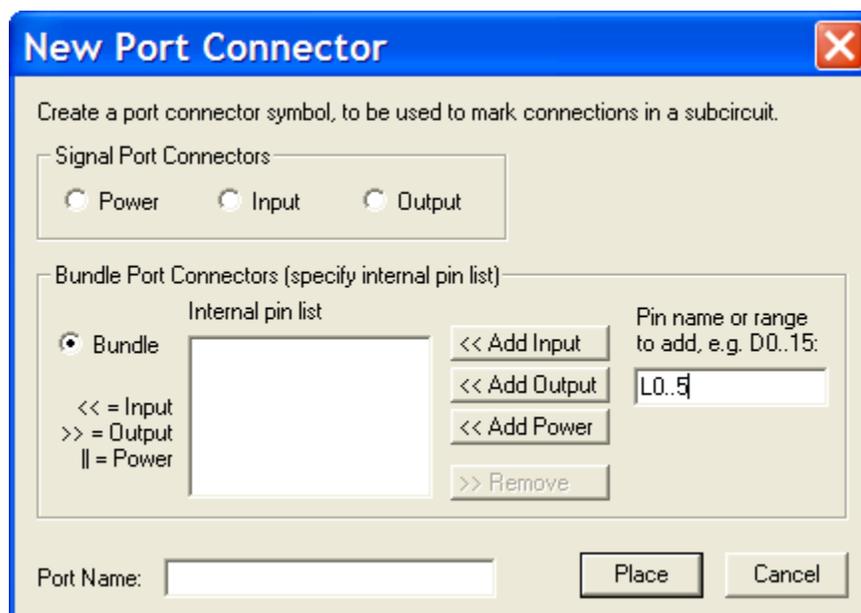
## 5.5 Tutorial 5: Advanced Subcircuit Topics

In this tutorial we will look at some more advanced operations on subcircuit blocks, such as the use of bundle pins and making custom modifications to block symbols. More information on subcircuits can be found in [Subcircuits and Hierarchy](#).

You must also read the section on [Bundles](#) for detailed description on bundle pin and signal usage rules.

### 5.5.1 Creating a Bundle Port Connector

- ❑ Create a new, empty design using the "New" command in the "File" menu.
- ❑ In the Options ribbon, select the Subcircuit tools and then the "New Port Connector" command.
- ❑ Click on the Bundle option, then enter the text "L0..5" in the Pin Name or Range box. Make sure this is L, zero, two dots and 5. This specifies a range of numbered pins.
- ❑ Click the Add Power button to add the pins to the list.
- ❑ In the Port Name box, type "L".



- ❑ Click the Place button. This returns us to the schematic diagram in device placing mode.
- ❑ Use the left and right arrow keys on the keyboard, if necessary, to orient the port connector as shown, and place it to the left on the diagram.
- ❑ Repeat the above procedure, using the New Port Connector command to create a bundle port with pins "R0..5" and named "R".
- ❑ Place the new port connector to the right on the diagram, as shown.



### 5.5.2 Connecting to the Bundle

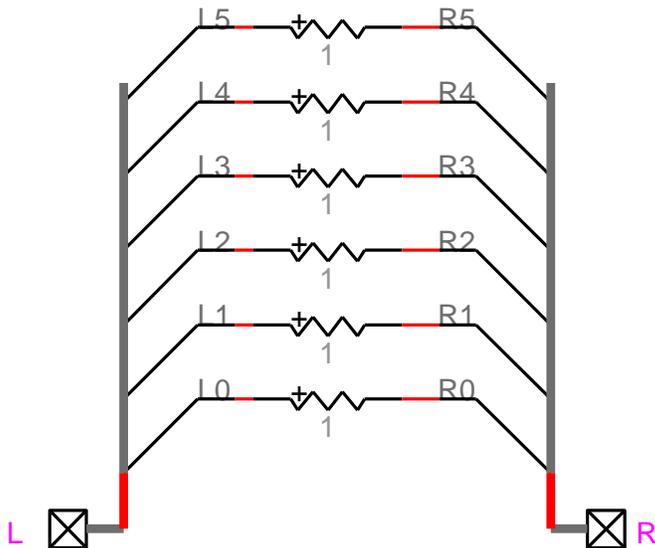
- ❑ Click at the end of the pin on the L port connector and extend a short bundle line, as shown.

## Tutorials

- ❑ Right-click anywhere along the bundle line and select the Breakout command. You should see the text "L0.5" already placed in the Pins List box.
- ❑ Click OK to create the breakout symbol.
- ❑ Place the breakout as shown, so that it connects with the bundle.
- ❑ This new breakout symbol will give us access to the individual signals inside the bundle.
- ❑ Repeat the above procedure for the R port connector, using the Breakout command to create circuit as shown at left. As before, you may need to use the arrow keys on the keyboard to orient the breakout symbol in the right direction.

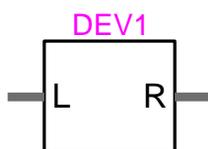
### 5.5.3 Adding Circuit Elements

- ❑ Place circuit elements connecting the two bundles. In this case, we have used the part "R small" for all six lines, but the exact circuit is not important for this tutorial.
- ❑ Use the Save command in the File menu to save the file as "Tutorial Bundle Subcircuit" in any convenient location. This is only for safekeeping, it is not a necessary part of this procedure, but is useful in case we discover errors later and wish to return.



### 5.5.4 Creating the Subcircuit Block

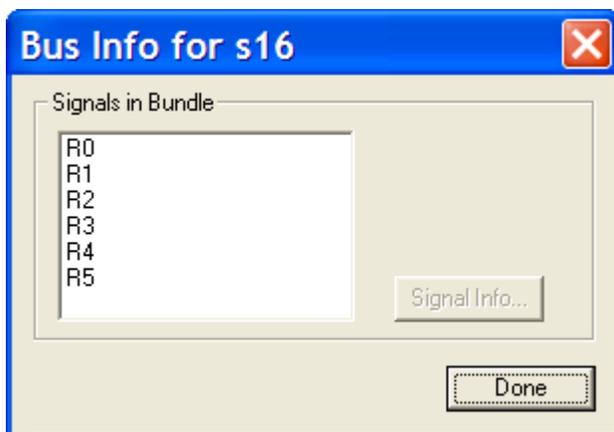
- ❑ Use the "Select All" command in the "Home>Edit" menu to select all of the items in the circuit we have just created. The keyboard shortcut is Ctrl+A.
- ❑ Select the "Create Subcircuit Block" command in the Subcircuit submenu of the Options ribbon.
- ❑ Select the "Quick Subcircuit" option and enter the name "BU" or any other valid name.
- ❑ Click the Finish button.



The “Quick Subcircuit” option has cut the selected elements from the circuit and placed them in a subcircuit block with a default symbol.

### 5.5.5 Verifying the Subcircuit Block

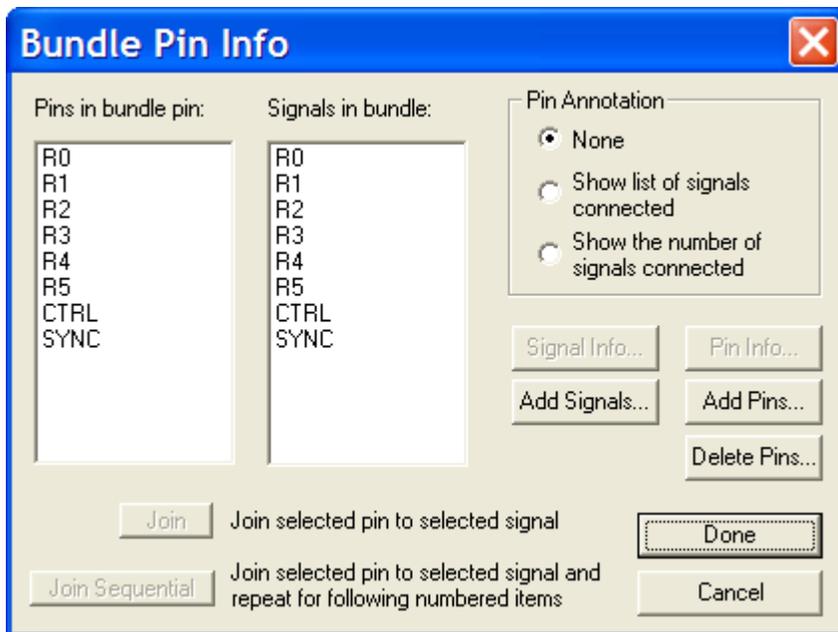
- ❑ Double-click on the block symbol to see its internal circuit and verify our original circuit is there as desired.
- ❑ Close the subcircuit window again to return to the parent circuit.
- ❑ Right-click near the end of the R pin on the block symbol and select the "Bundle Info" command.



Verify that the bundle contains the signals R0 to R5 that we specified.

### 5.5.6 Modifying Bundle Connections on a Subcircuit Block

- ❑ Right-click on the R pin, but this time, click close to the symbol body so that we get the pin information menu, as shown.
- ❑ Select the “Bundle Pin” Info command.
- ❑ Click the “Add Pins” button.
- ❑ Enter two names, Ctrl and SYNC, separated by a blank space, as shown.
- ❑ Select the Pin Type as Input. NOTE: This selection is important as it determines what type of signals can be attached to this pin.
- ❑ Click OK.



- ❑ Click Done on the bundle pin info box.
- ❑ You will receive a warning that you have made a structural modification to the device. Click OK to confirm the change. If you have not unlocked the subcircuit, it will give you a warning message, but the program will actually automatically unlock the subcircuit.

This operation has actually had a number of effects on the design:

- Two new internal pins have been added to the select bundle pin.
- Two new signals with the same names have been added to the attached bundle.
- The port connector in the internal circuit has had the same two pins added.
- The two corresponding signals have been added to the attached bundle in the internal circuit.

The above operation has made the new internal pins available inside the bundle and it is now our job to make whatever connections are needed to these pins inside the internal circuit.

- ❑ Double-click on the block symbol to open the subcircuit. You will notice that the internal circuit looks exactly the same as before. The above changes have only affected the list of internal signals in the R bundle and have not made any visible circuit change.
- ❑ Right-click anywhere along the R bundle (but not too close to either the port connect or the breakout symbols) and select the Breakout command. You will see that the default entries in this box now include the Ctrl and SYNC signals that we added above.
- ❑ Delete the other signals "R0..5" so that we are creating a breakout only for the new signals, then click OK.
- ❑ Place the new breakout as shown so that it attaches to the R bundle.

NOTE: It is important that the new breakout symbol touches the existing R bundle so that it connects to it. If you place the new breakout separately, you have created another, unconnected bundle containing two signals.

- ❑ If desired, place some circuit elements in the subcircuit and connect to the breakout. In this case, we have added the device PI Control (constant). The exact circuit connections are not important for this tutorial.

Note that since we selected the Input pin type when we added the bundle internal pins, these signals are control inputs to the subcircuit and can only be connected to control inputs on the attached devices.

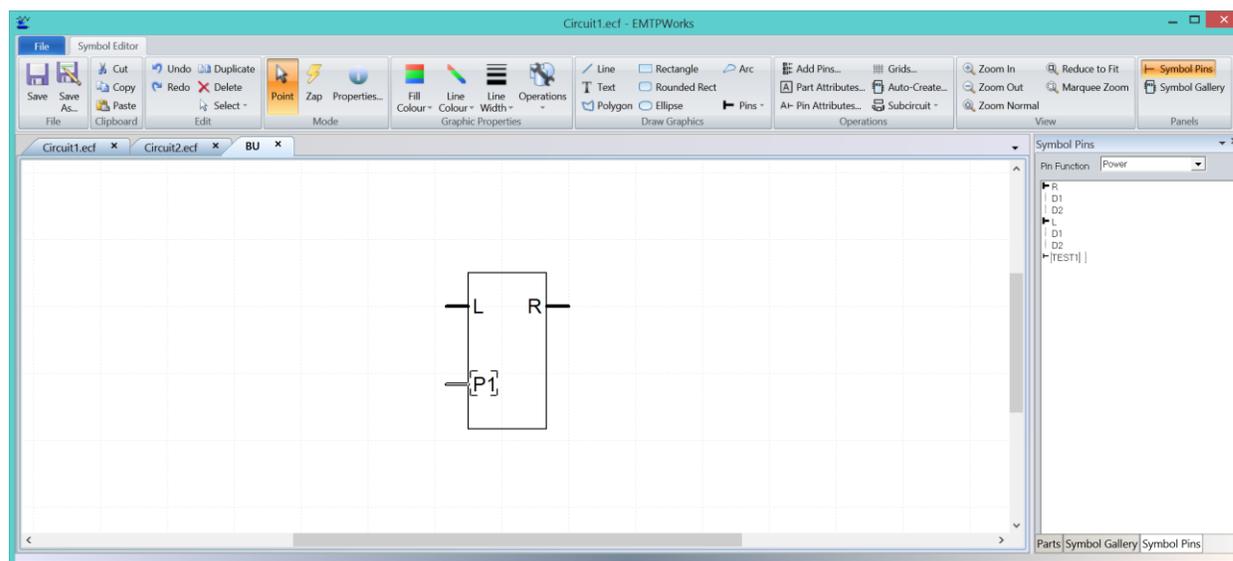
- ❑ Close the subcircuit window to return to the parent design.

### 5.5.7 Modifying the Symbol for a Block on a Schematic

- ❑ Right-click on the block symbol in the top-level circuit and select the “Edit Symbol” command. This will open the symbol in the symbol editor.
- ❑ Move the R pin and expand the box so that it appears roughly as shown.

### 5.5.8 Adding Pins to an Existing Block Symbol

- ❑ Click on the Pins menu in “Symbol Editor>Draw Graphics” and select left-facing pin.
- ❑ Place a new pin on the left-hand side of the symbol, as shown.
- ❑ Press the spacebar to terminate pin placement.
- ❑ Double-click on the new pin in the pin list and change the name to TEST1.
- ❑ Set the Pin Function to Power, if it isn't already.



### 5.5.9 Updating the Symbol on the Schematic

- ❑ Click the close box (X) in the “Symbol Editor” window tab and click Yes to confirm that you wish to save your changes to the symbol. The symbol on the schematic will be updated with the new version.
- ❑ Double-click on the new symbol to open the internal circuit. Note that there is no change to the internal circuit.
- ❑ Close the internal circuit again. You should receive an error message indicating that the port TEST1 has no matching port connector.

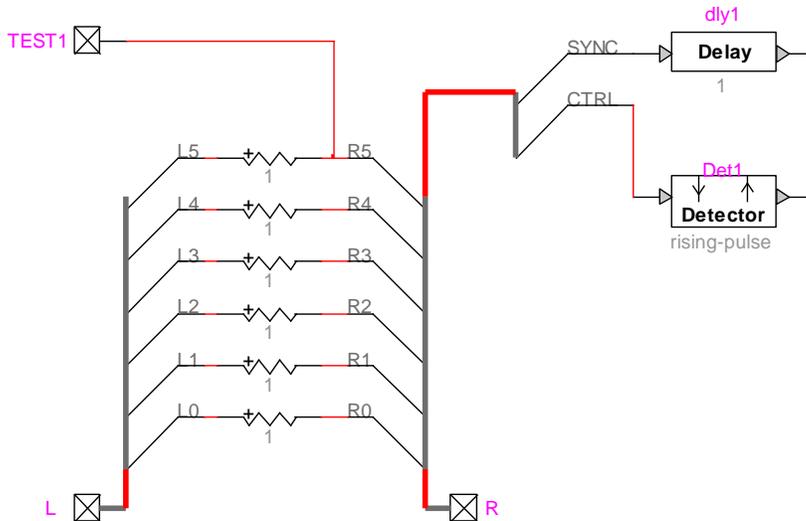
**IMPORTANT:** Whenever you make manual changes to a block symbol such as this, it is your responsibility to update the internal circuit, adding any necessary port connectors and circuitry needed to make it valid.

### 5.5.10 Updating the Port Interface

- ❑ Double-click again on the symbol to reopen the internal circuit.

## Tutorials

- ❑ Select the “New Port Connector” command in the Subcircuit sub-menu of the Options ribbon.
- ❑ Select the Power port type and enter TEST1 for the port name.
- ❑ Click the Place button and then place the port connector anywhere on the diagram.

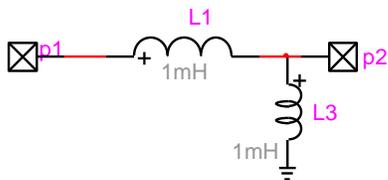


Close the internal circuit window again. This time you should get no error message.

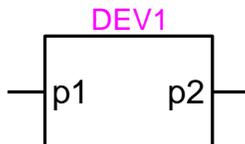
This procedure could have been also started by adding the port TEST1 in the subcircuit and then modifying the device symbol to account for it.

## 5.6 Changing phases in the port interface

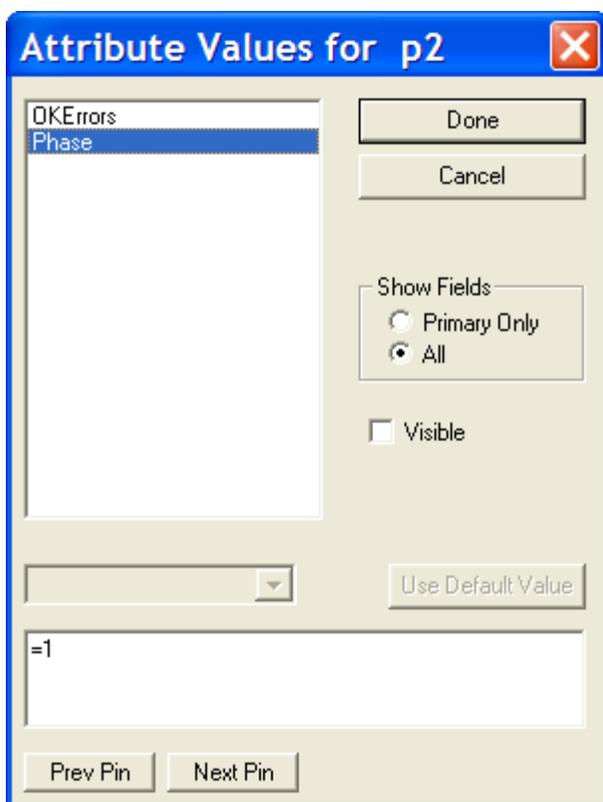
When a subcircuit is first created, its symbol pins inherit the phase numbers of the underlying port pins. The signal (Line Type) type of each pin is blocked through the pin Phase attribute. In this simple example the subcircuit is given by:



and the subcircuit symbol is given by:



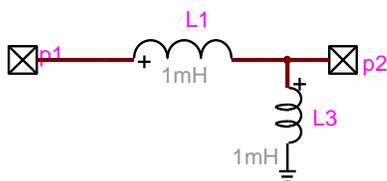
If you right-click on pin p1 or p2, and select the Attributes item, the Phase attribute is given the value "=1":



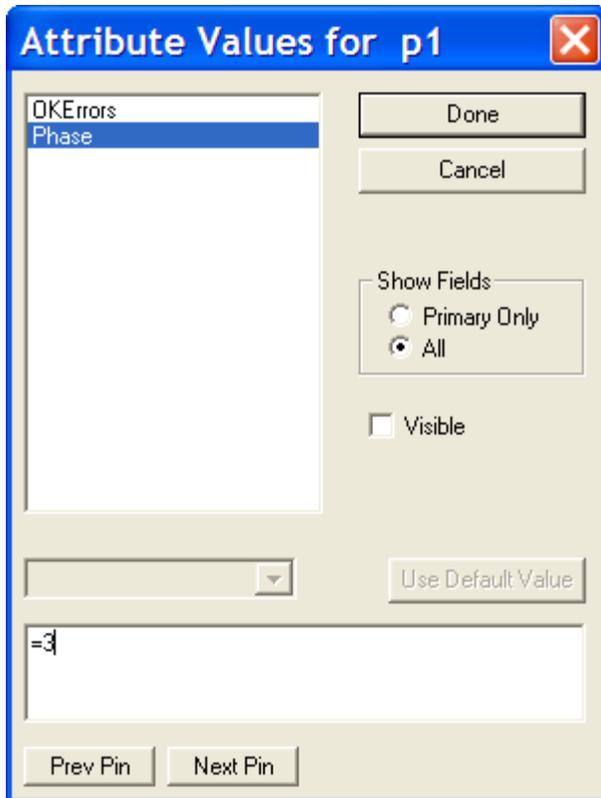
This is done automatically to avoid conflicts between the line types in the subcircuit circuit and its parent.

If it is desired to change the number of phases in the parent or in the subcircuit circuit, then you must manually adjust the Phase attribute for the related pins. If you forget to make the adjustment, the contents of the subcircuit may become disconnected or phase corruption may occur and remain undetected throughout the simulation. The possible complexity of hierarchical designs makes it almost impossible to automatically propagate phases or the "Line Type" more generally speaking.

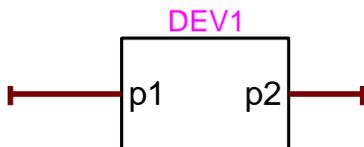
If in the example above, the subcircuit circuit becomes 3-phase:



then the Phase attributes of the parent symbol pins p1 and p2 must be set to "=3":



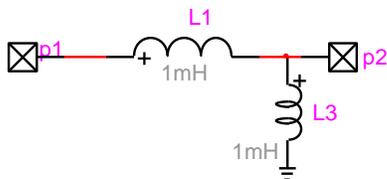
The signals connected to pins p1 and p2 must be manually set to the appropriate "Line Type":



The same procedure is applicable backwards when going from 3-phase to 1-phase.

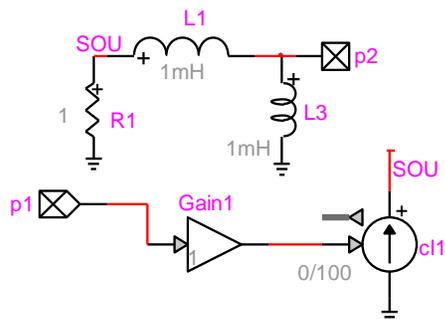
### 5.7 Changing the signal type in the port interface

It is feasible to change the type of the signal entering the subcircuit from power to control. There are several methods for changing the signal type in the port interface. The simplest method is given by these steps (applicable to the previous example):



- Move into the subcircuit and change the power port by an Input (or Output if another case) port. This can be achieved through "Options>Subcircuit>New Port Connector".

- The old port p1 must be deleted and the new port must be named also p1.



- Exit the subcircuit (go up to the parent circuit).
- Edit the subcircuit symbol: right-click and select "Edit Symbol".
- Select the pin p1 in the Symbol Editor pin list and change its "Pin Function" to Input.
- Save the changes to the symbol.

More information on signal types can be found in Signal Line Type and Device pin types.



# Reference

## 6 Basic Schematic Editing

### 6.1 Background information

#### 6.1.1 Design Structure

In EMTPWorks the term "design" refers to a complete, independent logical entity which is saved in a single file. 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. More information on the relationship between symbol libraries and designs can be found in About Device Symbols.
- 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 and increasing the memory allocated to the program will increase the size of the designs you can work with.
- A number of user-settable parameters are stored with the design and affect the entire design when changed. These include:
  - o attribute and pin number text style settings
  - o the attribute definition table
  - o signal and device auto-naming settings
  - o display options, such as crosshairs and printed page breaks
  - o page reference format
  - o hierarchy mode
- In a hierarchical design (that is, one in which a symbol can represent a nested circuit block), all hierarchy levels are stored in a single file.

#### 6.1.2 What is a Circuit

A circuit in EMTPWorks has the following characteristics:

- A circuit can be drawn on one or more pages (up to 1000 in this version). You 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 (tab) 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 a page connector device. Three versions of this device can be found in the built-in "pseudo devices.clf" library. The "Page Conn output signal" device is used to interconnect control signals across pages. The "Page Conn Power Signal" is used to interconnect power signals across pages. The "Page Conn Bundle" is used to interconnect bundles across pages.

### 6.1.3 Types of Objects in a Circuit

An EMTPWorks circuit is made up of five types of entities: devices, pins, signals, text objects and picture objects.

- A device is an object having a symbol, signal connection points called "pins", and optional text attributes, internal circuit and simulation information. A device in EMTPWorks 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.
- 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. A bus (bundle) 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.
- A text object is used to place a title block or other notation on the diagram. Text can be typed and edited directly within EMTPWorks, 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.
- A picture object is used to place any graphics item imported from another program or the symbol editor 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 symbol editor (or other application) window, edited using the device symbol tools, then copied and pasted back into the schematic.

### 6.1.4 How Design Templates Work

A design template is simply a normal design file that has its sheet size, attribute fields, hierarchy mode and other settings pre-defined for the application at hand. The simplest way to create a template may be to take an existing design that is set up the way you like it, delete all the circuit elements and extra pages out of it and save it in the appropriate template directory.

When you create a new design and select one of the template files listed, EMTPWorks just reads the file in the normal way, then renames it "Circuit1" (or the next available number), and disassociates it from the original file so that it cannot accidentally be Saved on top of the template. In all other respects, creating a new design using the New command is the same as doing an Open on the template design. All the settings and contents of the design template file become part of the new design.

More information on "Design Templates" can be found in [Design Templates](#).

### 6.1.5 EMTPWorks File Types

EMTPWorks deals with a number of different types of files. The most important type is the design file, which contains all the stored data of your schematics. Operations on design files are discussed in this chapter.

The following table summarizes the types of files used in EMTPWorks:

Description	File Extension	More Information
Design Files	ECF	Design Structure
Symbol Library Files	CLF	About Device Symbols
Report Script Files	RFM	EMTPWorks Export Script Language files. (see <a href="#">Export Script Language</a> )
JavaScript Files	JS	Scripting method in EMTPWorks based on JavaScript and EMTPWorks extensions to JavaScript. (see <a href="#">JavaScript based scripting</a> )
Design and Sheet Template Files	ECF	How Design Templates Work
Netlist file for EMTP	NET	Netlist file for EMTP
Initialization Files	INI	INI File General Format

In addition to the above files an extra set of files is created during the simulation process of EMTP. More documentation on Input/Output files is available through the Help tab of the dialog box "Input output files" opened from the menu "Simulate>View Output Files".

## 6.2 Top menus

The first ribbon is the File ribbon or File menu. The ">" character is used to denote submenus. Each time a menu item (submenu) is highlighted a tooltip appears for explaining its function.

The following bullets provide documentation links to the *non-obvious* options.

### 6.2.1 File Ribbon

- File>Revert: Revert to the last saved version of this design. See [Reverting to a Saved File](#).
- File>Options: Various options for opening files and creating new designs.
- File>Help & Support: Top Help and support section of EMTPWorks.

### 6.2.2 Home Ribbon

- Home>Clipboard>Paste Special: Paste from the clipboard with options. See [Paste Special](#).
- Home>Tools>Point: Returns to standard pointer.
- Home>Tools>Zap: Selects the Zap (delete) cursor. See [Deleting a Device](#), [Editing a Signal Line](#) and [Hide a Visible Attribute Value](#). You can use ESC button or spacebar to return to the standard pointer.
- Home>Tools>Name: This is the same as the Naming tool used for device attributes, pins and signal attributes. Selecting and clicking on a device, signal or pin gives access

## Reference

to the name attribute. Selecting and clicking on a visible attribute gives access to that attribute contents. This tool can be also used for editing text created with “Home>Draw Graphics>Text”.

- Home>Tools>Draw Signal: creates signals (see [Signals](#)). For “Draw Bundle” see [Bundles](#), for “Draw 3-phase Bus” (power system bus) see [Signal Line Type](#) and for Draw Phase A (B or C) see [Signal Line Type](#).
- Home>Tools>New Breakout: see also [Using Bundle Breakouts: New Breakout](#).
- Home>Tools>Attribute Probe: Displays attributes and information on the selected (clicked) object. See [Displaying and Setting Device Information](#).
- Home>Operations>Properties: shows the properties of the selected device or of the circuit (when no selected device). This is the same as the Ctrl+I shortcut and this menu is available from actual object (device) right-click menu or circuit right-click menu.
- Home>Operations>Orientation: Sets device orientation for the next device dragged in from a library. See also [Placing and Editing Devices](#).
- Home>Draw Graphics: Various tools for drawing graphics objects on the schematic.
- Home>Find>Quick Find: search for entered object name (signal or device). Only partial information can be entered. You can use the Ctrl+G shortcut to zoom on found signal or device.
- Home>Find>Advanced Find: Opens the Find tool tab. See [Locating Circuit Objects with the Find Tool](#). You can use the Ctrl+G shortcut to zoom on found signal or device.
- Home>Find>Property Browser: opens the Attribute Browser tool. See [Property Browser](#).

### 6.2.3 Options Ribbon

- Options>Design>Pages: Create, delete and modify circuit pages. See [Adding, Deleting and Titling Circuit Pages](#).
- Options>Design>Import Sheet Info: Used to set the sheet border and background style. See also [Importing Sheet Settings from another Design, Page or Template](#).
- Options>Design>Sheet Size Wizard: View and change sheet size. See also [Setting Sheet Sizes and Borders](#) and [Custom Sheet Size using Sheet Size Wizard](#).
- Options>Design>Border Wizard: View and change border settings. See [Border Wizard](#).
- Options>Design>Design Attributes: View and change design attributes. See [Entering Design Attributes](#) and [Predefined Attribute Fields](#).
- Options>Design>Define Attribute Fields: Create or modify attribute field definitions. See [Defining Attribute Fields](#).
- Options>Naming Options: Various naming options for devices and signals. See [Using the Auto-Naming Features](#), [Signal Auto-Naming Options](#) and [Device Naming Options](#).
- Options>Part Type>Update from Lib: Update the selected device from a library. See [Update from Lib](#).
- Options>Part Type>Save to Lib: Save the part definition of the selected device to a library. See [Save to Lib](#).

- Options>Part Type>Make Unique Type: Make a new part type definition for the selected device. See [Make Unique Type](#).
- Options>Part Type>Library to Circuit: Places all devices from the library into the current circuit. See [Library to Circuit](#).
- Options>Part Type>Circuit to Library: Save all symbol definitions from the current design to a library. See [Circuit to Library](#).
  
- Options>Subcircuit>Create Subcircuit Block: This is the top dialog for creating subcircuits. See [Subcircuit creation methods](#).
- Options>Subcircuit>New Port Connector: Creates a port connector for the subcircuit interface pins. See also [Creating Pseudo-Device Symbols](#).
- Options>Subcircuit>Attach Subcircuit: Attach a subcircuit to the selected symbol. See [Attaching Subcircuit](#).
- Options>Subcircuit>Detach Subcircuit: Detach the subcircuit associated with the selected symbol, making it a separate design. See [Detaching Subcircuit](#).
- Options>Subcircuit>Discard Subcircuit: Discard the subcircuit associated with the selected symbol. See [Discarding a Subcircuit](#).
- Options>Subcircuit>Push Into: To get into a selected subcircuit contents (same as Ctrl+Shift+I).
- Options>Subcircuit>Pop Up: Close the current subcircuit and go to the parent level (same as Ctrl+Shift+U).
  
- Options>Export: export functions.
  
- Options>Libraries>New Part: Opens the Symbol Editor for creating a new part.
- Options>Libraries>Library Operations: Various library maintenance functions. See the section [Working with Symbol Libraries](#) for detailed information on this option.

#### 6.2.4 View Ribbon

- View>Zoom>Zoom to Selection: Same as Ctrl+G, can be used for zooming onto a selection or a device found with find functions.
- View>Zoom>Pan: Allows panning the circuit diagram. Same as holding Ctrl button and using left mouse button.
- View>Zoom>Redraw: Refresh the circuit drawing.
  
- View>Panels>Parts Library: Shows or hides the Parts Palette. See [Placing and Editing Devices](#).
- View>Panels>Console: Shows or hides the Console window.

#### 6.2.5 Simulate Ribbon

This menu is created in the EMTP.INI file (see [Creating Custom Menus](#)).

- Simulate>Quick Run>Run: Generates the EMTP Netslist and starts the simulation. This command opens the Progress Panel and shows messages from the EMTP simulation engine.

## Reference

- Simulate>Advanced>Simulation Options: Set EMTP simulation options. The appearing dialog has help sections for explaining various options. This command is usually the first step before running a simulation.
- Simulate>Advanced>Start EMTP (do not generate Netlist): Same as above, only now the Netlist is not regenerated and the previous Netlist is used if available.
- Simulate>Advanced>Generate EMTP Netlist: The Netlist is just generated, but no simulation is performed.
- Simulate>Advanced>Generate and open EMTP Netlist: Same as above, only now the Netlist is opened in the file editor.
- Simulate>Advanced>View Output Files: Top level access to various output files resulting from an EMTP simulation.
- Simulate>Advanced>Delete generated files: Eliminates some of the files created during the EMTP simulation procedures.
  
- Simulate>View Scopes with ScopeView: Starts a separate application for visualizing EMTP simulation waveforms.
- Simulate>View Scopes with MPLOT: Starts a separate application for visualizing EMTP simulation waveforms.

### 6.2.6 Design Ribbon

This menu is scripted. It is created in the EMTP.INI file (see [Creating Custom Menus](#)) and uses the EMTPWorks scripting techniques for programming various tasks. You can open the scripts associated to each item and learn on [script programming methods](#). Documentation on the used scripting languages is available in [Other Documentation](#).

- Design>Device Operations>Exclude Devices: Exclude the selected devices from the Netlist. This command is similar to the Exclude/Include available from device right-click menu. The difference is that in this case it can act on one or more selected devices in the design. You can select entire circuit sections using the Shift and Ctrl keys while clicking on devices, or using a bounding box. See [Device right-click menu](#).
- Design>Device Operations>Include Devices: Include the selected devices back into the Netlist. This command is similar to the Exclude/Include available from device right-click menu. There is no change if the selected devices are already included. The difference is that in this case it can act on one or more selected devices in the design. You can select entire circuit sections using the Shift and Ctrl keys while clicking on devices, or using a bounding box. See [Device right-click menu](#).
- Design>Device Operations>Show/Hide: Commands for showing or hiding various device attributes.
  
- Design>Signal Operations>Signal Parameters: Allows setting properties for selected signals. This is similar to the item "Signal Parameters" available from individual signal right-click menu (see [Signal right-click menu](#)). The difference is that you can now selected one or more signals and apply simultaneous changes.
- Design>Signal Operations>Show Hide: Commands for showing or hiding signal names.
  
- Design>Steady-State View>Load Load-Flow solution data: Used for the Load-Flow option.
- Design>Steady-State View>Hide: Hide Load-Flow solution data on the schematic.
- Design>Steady-State View>Show data: Show steady-state solution results for selections made on the schematic.

- Design>Scopes>Control signal scopes: allows to select or cancel control signal scopes (“Control signal scope” device).
- Design>Utilities>Define Global Variables: See documentation link available after selecting this option. This option allows creating Global Variables to be used in various locations in EMTPWorks mask scripts.
- Design>Utilities>Run a script: Allows to select and run a script on the current design.
- Design>Patch: Special patch function applied to the design. This is provided for upgrading design templates and/or updating devices. You may be asked to perform this command when a simulation results into an error or when a new option became available. The design patch level is saved into the attribute PatchLevel. It is not harmful to run this command on any design, since the patch level will be automatically detected and is applicable for the current version of the program.
- Design>Patch Devices: Special patch function applied to devices. This is provided for upgrading devices or fixing device bugs. You will be asked to perform this command when a simulation results into an error or when a new option became available. The device patch level is saved into the attribute MPLevel.

## 6.3 Basic Operations

### 6.3.1 Creating a New Design

A new design is created by selecting the New command in the File menu, then selecting Design, and then choosing from one of the available templates. The new design can be also created using the “Create a New Design” or “Create a New Document from Template” option in the menu “File>Options”.

The new design will consist of an empty circuit with a single page. This page will appear in a circuit window and given a default name. The New command does not create a disk file. The design exists only in memory until you do a "Save As" from the File menu or Home ribbon.

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

### 6.3.2 Choosing a Template

EMTPWorks does not have a fixed set of templates, but rather displays the ones that are available in the designated Templates folder. The default EMTP template file "EMTP Default Design.ecf" is found under the program directory Templates.

A template may contain a variety of initial settings for options and items such as a sheet border and title block, that make it easier for you to get started on a design. You can create your own templates that suit your needs, once you are comfortable with the basic operation of EMTPWorks. The templates are part of the open architecture methods of EMTPWorks and should be used with care not to corrupt simulation data and built-in essential attribute data.

It is important to remember that program updates may change the built-in template files. If you have created your own template files from the existing template file, you may need to adjust it for upgrading to include new attributes and settings. The changes are usually available through patch scripts, but some changes maybe more complicated, so that the recommended approach is to recreate your template file by reapplying your modifications to the latest template file “EMTP Default Design.ecf”.

See [Design Templates](#) for more information on creating design templates.

### 6.3.3 Opening a Design

A design file can be opened in one of two ways:

## Reference

- Select the Open command in the File menu. This command allows a design to be opened from a disk file using the standard Windows file open box.
- Locate the file using the Windows Explorer and double-click on the file's icon. If EMTWorks is not already running, it will be started up. If it is already running, it will come to the front and the selected file will be opened.

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.

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.

### 6.3.4 Navigating Around a Schematic Page

In addition to the standard scroll bars, EMTWorks has these convenient features for moving around a diagram.

#### 6.3.4.1 Zooming

##### Controlling Zoom by Menu Commands

The View>Zoom menu contains four menu items for controlling the zoom factor of a window. These commands control screen display only and have no effect on the stored circuit information, printed output, or graphics files.

Zoom Normal	Sets the screen scale to 100%.
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.
Zoom in	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. It is recommended to remember the keyboard shortcut Ctrl+Shift+E
Zoom out	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%. It is recommended to remember the keyboard shortcut Ctrl+Shift+R.

##### Controlling Zoom with the “Marquee Zoom” Tool

The “Marquee Zoom” in the “View>Zoom” toolbar 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.

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

- 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. Moving most of the way across the window is equivalent to doing a "Reduce to Fit".

#### 6.3.4.2 Panning method

Another convenient method for moving around a design is to hold down the Ctrl key, click and hold down the left mouse button while moving the mouse. This allows moving the design page in any direction. The Pan command is also available in the "View>Zoom" toolbar.

#### 6.3.5 Opening Circuit Page Windows

In EMTPWorks, each circuit page is displayed in a separate window that can be opened, closed, scrolled and zoomed independently. The Pages command can be used to display any existing page, as well as adding new pages and setting page order and title. The Pages command can be found in "Options>Design>Pages", or in the circuit popup menu which is displayed by right-clicking in an empty location in the diagram.

See more information on the Pages command in [Adding, Deleting and Titling Circuit Pages](#).

#### 6.3.6 Locating Circuit Objects with the Find Tools

The find tools allow you to locate any device or signal by name. There are two options: "Home>Find>Quick Find" and "Home>Find>Advanced Find".

The "Quick Find" command is a basic search method. The object name can be entered completely or partially before clicking on the "Quick Find" button.

To use the "Advanced Find" method:

- Click on the "Advanced Find" button to open the Find palette.
- Select the type of object you wish to search for, devices or signals.
- Enter the name of the object to search for, or other search criteria.
- Click the Search button.

The Find tool automatically displays the first item found. The list of found items can be used to look at any other objects matching the search criteria. You can also use wildcards as explained in the help menu "?" of the Find tool window. The command "View>Zoom>Zoom Selection" can be used to zoom on the found object.

The Find tool is scriptable and can be customized to find objects or scan the design for many different purposes. See [Creating Scripts for the Find Tool](#).

#### 6.3.7 Locating Objects Using the Browser Tool

The Browser tool ("Home>Find>Property Browser") is a convenient way of viewing all objects in the design in tabular format. Clicking on any item in the Browser's spreadsheet area automatically displays the corresponding item in the schematic.

See also [Browser for attributes](#).

#### 6.3.8 Saving a Design

An open design can be saved to disk using the standard Save and "Save As" commands in the File menu (or "Home>File" toolbar). The "Save" command saves the circuit back into the file that was most-recently opened. It will be disabled if no file has been opened or if you've created a

new design that has never been saved. If you select "Save As", a standard file save box will be displayed requesting the name of 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.

### 6.3.9 Reverting to a Saved File

The Revert command in the File menu 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.

### 6.3.10 Saving a Circuit Page in WMF or DXF Graphics Formats

EMTPWorks can save diagrams in the standard WMF (Windows Metafile) and DXF (AutoCAD) graphics formats. This capability allows you to pass graphics to other programs for plotting, enhancement, or incorporation into other documentation. Here is some more information on these formats:

- WMF is a standard format under Windows for representing graphics objects such as lines, circles, text, etc. This format can be read by many Windows-based graphics and word processing packages. Note that EMTPWorks cannot read these files, it can only create them. A WMF file is strictly a graphics format and does not contain information about signal connectivity, pin functions, attributes, etc.
- DXF is a format developed by AutoDesk for use with their AutoCAD series of general CAD packages. In addition to AutoCAD, most general drafting packages can read and write DXF. Like WMF, a DXF file is strictly a graphical representation of a diagram and does not contain all the information necessary to construct a logical schematic. EMTPWorks has no ability to read DXF files. Note that Enhanced Metafile and bitmap pictures that have been pasted onto a diagram (e.g. for a logo in a title block) are not exported to DXF.

#### Generating a WMF File

- ❑ Open the design file you wish to export. If your design has multiple pages and you only want to export one of them, make that the current page (i.e. the topmost window).
- ❑ Select the Export command in the "Home>Export>Export" menu.
- ❑ Select the WMF option and click Next
- ❑ Select either "Export all Pages" or "Export Current Page Only", as appropriate.
- ❑ If desired, click the Browse button to enter a different filename. The default will be the same name as the design with the extension changed to "WMF".
- ❑ Click the Finish button to complete the operation.

#### Generating a DXF File

- ❑ Open the design file you wish to export. If your design has multiple pages and you only want to export one of them, make that the current page (i.e. the topmost window).
- ❑ Select the Export command in the "Home>Export>Export" menu.
- ❑ Select the DXF option and click Next
- ❑ Select the appropriate color option. If you choose the "Export Color Information" option, EMTPWorks will choose the closest standard DXF color for each object on the schematic.
- ❑ Choose whether you want to use Release 12 or Release 13 format. Release 13 provides better representation of curves and ellipses, but is not compatible with all readers. You should try a test run with the desired target application (making sure your test includes a variety of symbols) before doing any major conversions.
- ❑ Click the Next button.
- ❑ Select either "Export all Pages" or "Export Current Page Only", as appropriate.

- ❑ If desired, click the Browse button to enter a different filename. The default will be the same name as the design with the extension changed to "WMF".
- ❑ Click the Finish button to complete the operation.

For some applications it is feasible to gain advanced editing capabilities by simply copying and pasting an EMTWorks circuit into a desired application window.

### 6.3.11 Printing

The "File>Print" command allows you to print all or part of your circuit diagram to any printer or other output device that is selectable as a printing device.

#### Specifying the Page Number Range

For purposes of specifying a range to print, printed sheets are numbered from top to bottom, then left to right, starting on circuit page 1. The page range settings in the standard Print dialog box correspond to printed sheet numbers, not to circuit pages.

In multi-page circuits the print 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 they are explicitly placed there using a text variable. See [Using Text Variables](#).

In hierarchical designs, pages are numbered separately for each circuit level. The Design Preferences command is used to determine printing extent. See more information on printing in hierarchical designs in [Printing Hierarchical Designs](#).

#### Setting the Printer Page Setup

The Print Setup command in the File ribbon allows 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 "Options>Design>Design Preferences" command. Depending on the settings in the "Design>Sheet Size Wizard", the page setup may affect the displayed page border in your schematic

#### Fitting the Diagram to the Available Paper

EMTWorks has a very flexible ability to set the size of a circuit page, either independently of the current printer, or derived from the printer settings. The "Sheet Size Wizard" command in the "Options>Design" toolbar is used to select from a number of size options. If, as a result of these settings, 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 "Print Setup". You can preview the page breaks by using the "Show Printed Page Breaks" option in the "Options>Design>Design Preferences" command.

#### Fitting a Printout to a Single Sheet

If the current sheet size settings are such that your printout does not fit to a single sheet on the current printer, you can force it to be scaled by taking these steps:

- ❑ Select the "Sheet Size Wizard" command in "Options>Design>".
- ❑ If your design has multiple pages or hierarchy levels, you will be asked which of these you want to modify, so make the appropriate selection, and click Next.
- ❑ Select the "Fixed Border Size" option and click Next
- ❑ The Wizard will have filled in the sheet size setting with the current sheet size, which you should not change, so just click Next.
- ❑ Select the "Scale to Fit Printed Page" option and adjust the controls to print on 1 wide by 1 high.
- ❑ Click Finish.

## Reference

When the design is printed, each page will be scaled as needed to fit on a single sheet of paper according to the current printer settings.

More information on sheet size settings is given in Custom Sheet Size using Sheet Size Wizard.

### 6.3.12 Backup Procedures

EMTPWorks has several backup features to allow you to fine-tune your backup strategy to suit your requirements. These features are controlled by inserting keywords in the INI file, as described in the following sections.

#### 6.3.12.1 Enabling Auto-Backup on Save

When auto-backup is enabled, the following sequence of actions is performed every time you select the Save command in the File menu when a circuit file is current:

- The existing design file is renamed to the same name with ".bak" appended. If a file with the ".bak" extension already exists, it is discarded.
- The design is saved to a new file with the original design name in the same directory.

Note that the "created date" of the new file is set to be the same as the old file so it will retain its historical created date.

The auto-backup feature (on by default) is enabled by placing the following line in the INI (EMTP.INI) file:

```
Backup = On
```

See [\[Drawing\]](#) section section of INI file.

If a disk failure occurs (such as disk full) while the file is being saved, the existing file will already have been renamed to "design.bak" and will not be automatically switched back to its original name by the program.

#### 6.3.12.2 Enabling Timed Auto-Save

Timed auto-save allows you to request a prompt to save the design every time a specified number of minutes have elapsed. At the appointed time, a box will give you a choice of:

- Saving the file. This will also create an auto-backup file, as described above, if enabled.
- Postponing until another interval elapses.
- Disabling timed auto-save completely.

Note that if you choose the last option, timed auto-save will be disabled until you quit and restart the program.

Timed auto-save is an extension of the auto-backup feature and is enabled by placing the following line in the EMTP.INI file:

```
AutoBackup = 30
```

where "30" can be replaced by any desired number of minutes.

- ❖ See [\[Drawing\]](#) section of INI file for more information.

### 6.3.13 Closing a Design

The Close command in the File menu closes all the circuit windows associated with the current design and removes all data from memory. If the design is hierarchical, all hierarchy levels will be closed. If any changes have been made to your design since the last Open or Save, then you will

be asked if you wish to save those changes. The same effect is achieved by clicking the "X" box in the upper left corner of the last circuit window tab.

### 6.3.14 Disposing of a Design

EMTPWorks has no built-in command to dispose of a design file. All information about a design is stored in a single file. Deleting the file using the Windows Explorer (or any similar system command) has the effect of erasing all data concerning that design. In addition to the design file, there maybe one or more files and directories related to the simulation. More information on EMTP related files can be found in the Help tab of the menu "Simulate>Advanced>View Output Files". A specific menu "Simulate>Advanced>Delete generated files" is used to eliminated some of the files created during the EMTP simulation procedures.

### 6.3.15 Exiting EMTPWorks

The Exit command in the file menu closes the EMTPWorks application. If any designs are open and have been modified, EMTPWorks will ask if you wish to save them.

## 6.4 General Editing

### 6.4.1 Undo and Redo

The Undo command ("Home>Edit" or Ctrl+Z) 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", adding pages, or any menu commands involving a dialog box are usually not undoable. EMTPWorks stores up to 10 levels of Undo information.

Undo never changes the contents of the Clipboard. After a Cut operation, for example, Undo will restore the schematic, but leave the Cut objects on the Clipboard.

The Redo command ("Home>Edit" or Ctrl+Shift+Z) redoes the last Undo command. It will only be enabled immediately after an Undo operation. Any other editing operation will disable this item.

Data entered into a device cannot be undone if you click OK.

### 6.4.2 Selection

#### 6.4.2.1 Selecting Groups of Objects

Several methods are available for selecting multiple objects:

- 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).
- A group of interconnected devices and signals is selected by double-clicking on any device in the group while holding the Ctrl key pressed. If a circuit is completely interconnected, this will select the entire circuit.
- The "Select All" command in the "Home>Edit" menu selects all items on the current page except background items.
- 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 clicked on. Thus you can add to the selected group until the desired collection of items is selected.
- ❖ See [Background Objects](#) for more information on background objects and how to select them

#### 6.4.2.2 Deselecting a Selected Object

## Reference

All currently selected objects are deselected by clicking in an empty area of the schematic window (see other options in "Home>Edit"). A single item can be deselected by holding the Shift key while clicking on it.

### 6.4.2.3 Multi-page Selection

The EMTPWorks commands that operate on selected objects normally only affect objects on the current page. However, some external tools, such as the Property Browser and Find, 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 the Property 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.

### 6.4.3 Clipboard Commands

The standard clipboard commands Cut, Copy, Paste and Paste Special can be used to move or copy circuit fragments, graphical and text information within a single circuit window, between multiple windows, and between different programs (e.g. word processing or drafting).

#### Using Clipboard Data From Other Programs

When you enter EMTPWorks, the clipboard may contain graphical or text information Cut or Copied from a document in another program.

#### Picture (Graphical) Clipboard Objects

A picture can be Pasted directly onto your diagram in any desired position. Once placed, it can be dragged, Zapped, Cut, Copied, Duplicated, etc. using the various editing commands available for other circuit elements. Here are some notes on pasting pictures onto a schematic.

- When a picture item is selected, the Properties command can be used to select an optional border and to indicate that the picture should be considered to be part of the sheet background or border. See [Creating a Picture](#) for more information.
- A graphics object is considered to be a monolithic item on a schematic and can only be moved or deleted in its entirety.
- EMTPWorks accepts pictures from the clipboard in either Bitmap or Enhanced Metafile format. The Bitmap format is typically produced by "paint" programs. Bitmap format is not generally recommended as it does not scale evenly and gives relatively poor printing results. Enhanced Metafile Format (EMF) is produced by object-oriented drawing programs, such as the drawing tools in Microsoft Word. EMF is much better suited for creating mechanical drawings or logos since it will reproduce much better on a variety of output devices. In some cases, programs may place both types of images on the clipboard. You can use the "Home>Clipboard>Paste Special" command to select which type of data is used.

#### Text Information on the Clipboard

Text information from a word processor or text editor can be pasted into a text block. See [Creating a Text Block](#) for more information.

### Using Clipboard Data from EMTPWorks

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

- A picture of the selected items, which can be pasted into a graphics document using most drawing programs. Both bitmap and Enhanced Metafile Format versions of the picture are placed on the clipboard.
- The EMTPWorks circuit info for the selected items. This data is in a format that only EMTPWorks can understand and is discarded when you exit the program.

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 simultaneous EMTPWorks 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.

#### 6.4.3.1 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

As well as the standard location in the Edit menu, there are also Cut, Copy, Paste, Clear and Duplicate commands in the Device and Signal pop-up menus. These operate on the single object that was clicked on to display the menu.

#### 6.4.3.2 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.

#### 6.4.3.3 Paste

Paste replaces the cursor with an image of the contents of the Clipboard in the circuit window. The data being pasted may be a circuit group copied from within EMTPWorks, or it may be text or picture information created by another program. 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. Paste will be disabled if there is no information on the clipboard of a recognized type.

#### 6.4.4 Signal Connection Checking on Paste

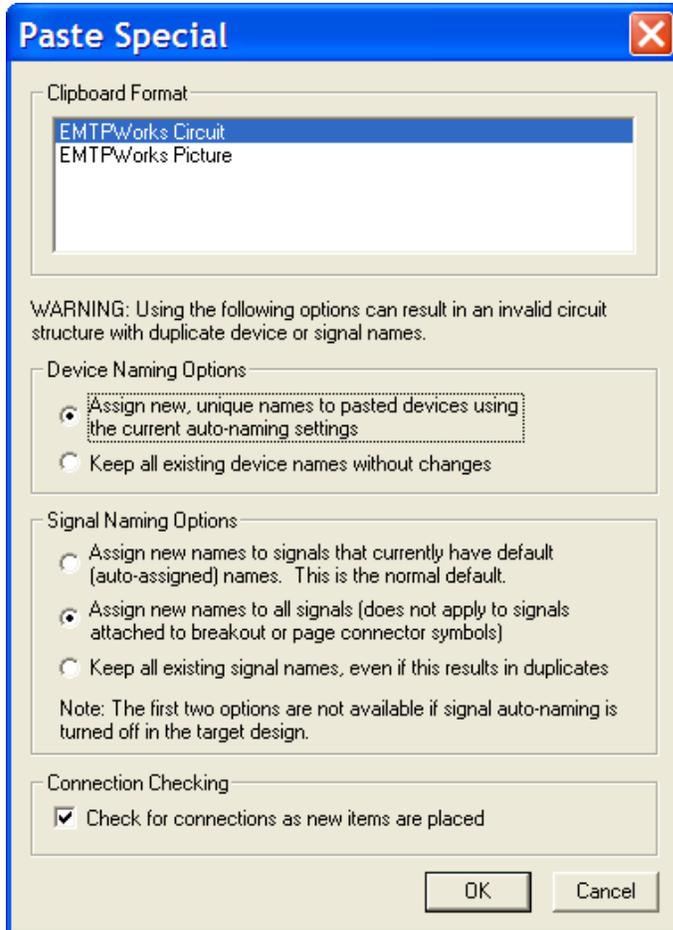
EMTPWorks checks for signal connections only at "loose ends" in the signal lines being pasted, i.e. ends of line segments which do not touch devices or other line segments.

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

The Ctrl key can be also depressed when moving circuit groups with the mouse or arrow keys.

### 6.4.5 Paste Special

The “Home>Clipboard>Paste Special” command performs the same operation as the Paste command described above, but it allows you to select a number of options for how the data is used. Here is a typical box that will be displayed in response to this command:



#### Clipboard Format

The clipboard format selection provides a list of the data formats that are currently on the clipboard and are acceptable to EMTPWorks. The following table summarizes the common formats that you will see in this list.

Bitmap Picture	This is the most common graphics format created and used among Windows applications. In general, this format is not recommended as it does not print and scale well.
Enhanced Metafile Picture	This is the Windows "object graphics" format, i.e. it describes images as collections of lines, circles, etc. For this reason, it is better suited as a format for

printed logos, mechanical drawings, etc.

EMTPWorks Picture	This is the internal graphics format for picture data (i.e. without circuit information) used in EMTPWorks. This is the best format for copying images within the program as it preserves the image information exactly as it appears on a drawing. <b>WARNING:</b> This is strictly a graphics image format and contains no circuit connection data. If you copy a collection of circuit elements, then paste it as a picture you are losing all the device and signal connection information.
EMTPWorks Circuit	This is the format used internally for normal clipboard operations involving schematic data. Unless you have a good reason for choosing another format, this should be the default choice for all Paste operations on a schematic.
Unformatted Text	This is a plain text format, accepted by most text processing and drawing programs. Pasting text will place a "miscellaneous text block" on the schematic.

Obviously, the exact contents of this list will depend on what Cut or Copy operations have preceded the selection of the Paste Special command.

### Device Naming Options

This selection allows you to choose how names are applied to any devices contained in the circuit data on the clipboard. You can choose to apply a new name based on the current settings for naming options (the default operation), or to keep all names as they are.

Warning: Pasting with the "keep existing names" option may create duplicate device names that will render the name table invalid. It will then be your responsibility to make any corrections needed.

See also [Device Names](#).

### Signal Naming Options

The Signal Naming Options give you more control over the process of assigning new names to signals when circuit scraps are pasted back into a design. The available options are:

- Assign new names to signals that currently have default names. This is the normal operation performed by Paste. This option also renames signals that have a non-default name, but whose name is no longer visible because of the particular combination of elements that was copied.
- Assign new names to all signal. All signals will have a new name assigned except those that have a hard connection to a pseudo-device such as a bundle breakout or ground connector that forces a name. This guarantees that no connections by name are made with the new elements.
- Keep all existing signal names. This option guarantees that all names are pasted exactly as they appear in the original. This option is dangerous since hidden signal names may result in undesired virtual connections. This option is only recommended when pasting visible signals and connectivity through these visible signals must be maintained.

Reference

See also [Naming Signals](#) before using signal naming options.

### Connection Checking

This checkbox allows you to turn off the normal connection checking (see [Signal Connection Checking on Paste](#)) that occurs when circuit elements are pasted.

This option can result in diagrams that look connected, but are not. It is the user's responsibility to make sure the diagram is cleaned up to produce appropriate connection information for Netlist purposes.

#### 6.4.6 Zooming in on Selected Objects

The "View>Zoom>Zoom to Selection" 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 100% maximum. This command works even if the selected objects are on a page other than the current one.

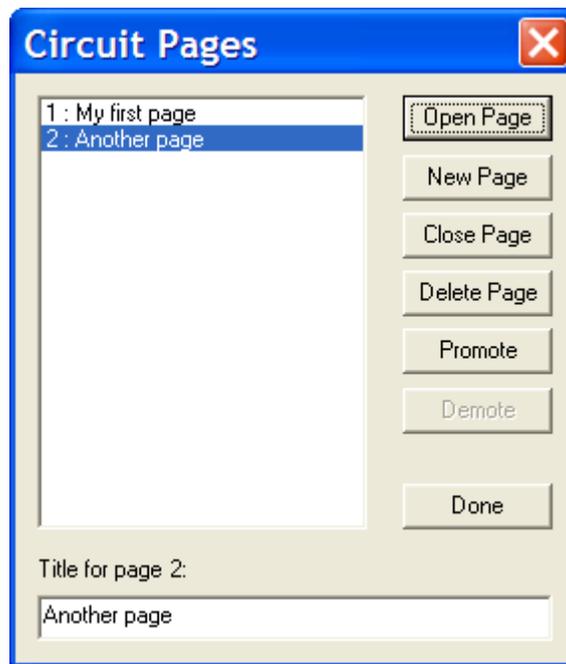
#### 6.4.7 Adjusting the Position of All Objects on a Page

If you wish to adjust all the objects on a page to make room for editing or to re-center an edited drawing, you can:

- Use the "Select All" command in the "Home>Edit" menu to select all the objects on the page, then click and drag inside any device symbol to move the entire group.

#### 6.4.8 Adding, Deleting and Titling Circuit Pages

The Pages command (Options>Design toolbar) 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:



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 Ctrl or Shift keys pressed while clicking in the list.

You can also right-click in an empty area of the circuit (i.e. not on a device or signal) and select the Pages command as a shortcut to display this box.

Operations in this box cannot be undone.

### **Opening a 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.

### **Creating a New Page**

The "New Page" button causes a new, empty page to be added to the circuit and opens it in a new window.

### **Deleting a 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 "Home>Edit>Select All" and "Home>Edit>Delete" commands) before this will be allowed. Higher numbered pages will be renumbered when this page is removed.

### **Changing Page Order**

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.

### **Setting a 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 displayed in text notations and title blocks using the \$PAGETITLE variable. See [Using Text Variables](#).

## **6.4.9 Duplicating One or More Objects**

The Duplicate command (Ctrl+D or Home>Edit>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 portion 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 arrow keys on the keyboard. You can rotate objects and entire circuits.

### **6.4.10 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 (Ctrl+Q) command in "Home>Operations" toolbar.
- 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.

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

### **6.4.11 Making a Circuit Read Only**

You can make a circuit read-only to prevent accidental modification to it. This is done by following these steps:

## Reference

- ❑ Click in an empty area of the schematic to ensure that no objects are selected.
- ❑ Choose the Properties command in the “Home>Operations” menu. This will display the circuit info box already described.
- ❑ In this box, turn on the "Read Only" switch. Note that this switch is enabled only when the "Current Circuit" button is selected since it only applies to one circuit level.

In a hierarchical design, the "Read Only" status applies only to the current circuit level in a hierarchical design and to all levels below it. Note that "Read Only" is not the same as "Locked" in that it allows you to view the circuit but not change it.

### 6.4.12 Setting Design Attributes

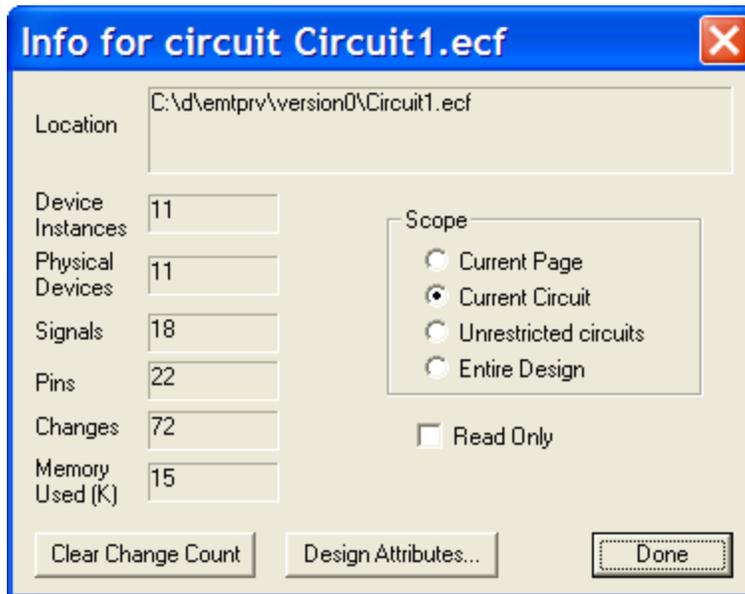
To edit text attributes associated with the design, you need to display the general design attribute editing box. This can be done in either of these ways:

- Right click in an empty area of the schematic, and then choose the "Circuit Info" command. This will display the circuit info box already described. In this box, click on the "Design Attributes" button.
- Select the "Design Attributes" command in the “Options>Design” toolbar.

For more information on the usage of attributes, see [Entering Design Attributes](#).

### 6.4.13 Showing Circuit and Design Statistics

If no objects are selected in the circuit (i.e. if you have clicked in an empty portion of the diagram) then the right-click "Circuit Info" command (or Ctrl+I) will display this general design information box:



The buttons on the right hand side of this box allow you to choose 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 subcircuits 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:

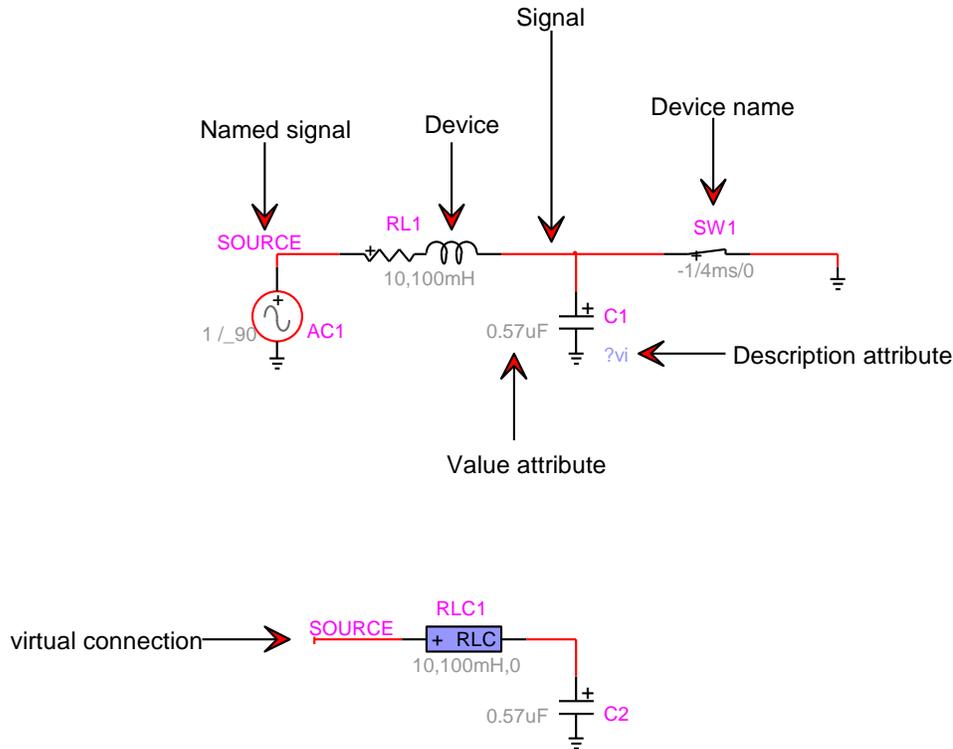
nnn devices	This is a count of devices in the selected scope. Pseudo-devices, such as page connectors and breakouts are not included. In "Current Page" or "Current Circuit" mode, this count includes devices that have subcircuits. In "Unlocked Circuits" and "Entire Design" mode, this count includes only "bottom-level" devices, i.e. those without a subcircuit or whose subcircuit 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. This value can be cleared using the "Clear Change Count" button and can be extracted in reports using the \$CHANGECOUNT script keyword.
nnnK memory used	This is a count of the amount of main memory occupied by the selected part of the design, in Kbytes.

## 6.5 Placing and Editing Devices

### 6.5.1 Devices: introduction

Devices are the building blocks of a design. There are electrical (interconnected) devices and other special devices for setting simulation options or documentation. This generic example shows interconnected devices:

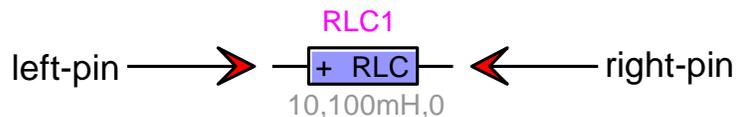
## Reference



Devices are dragged in from the Parts Palette (can be turned on using "View>Parts Library"). Each device has a default orientation in its library. The orientation can be changed after placement in the circuit. Another technique is to double-click on the part name in the library and then drag into the design. The device becomes stuck to the mouse pointer and the arrow keys can be used to rotate before a single click for placement. Use the Spacebar or the Pointer tool to stop the sticking. Instead of using the arrow keys, the user can also hit Ctrl+Q and select the orientation. The orientation of the next dragged in device can be pre-selected or reset to default using Ctrl+Q.

Once a device is positioned in a circuit it can be moved around using the mouse key or the keyboard arrow keys. When a device moves it tries to maintain existing connectivity, but this is a complex task and the user may have to redo connectivity manually under some conditions.

A device may have no pins, one or more pins. A two-pin device is shown below. The left-pin or the pin close to the "+" sign is also called a k-pin. The right-pin or the other pin, is also called an m-pin. The plus "+" sign is used to provide polarity for power devices.



Pins can be clicked on to select. There is also a right-click menu.

The design may have network devices: electrical devices or simulated devices recognized in EMTP. There are also various other types of devices. The options.clf library has devices for selecting EMTP simulation options. The symbols.clf library has several devices for symbol editing. Subcircuits are considered as black box devices.

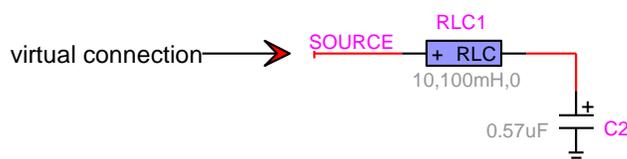
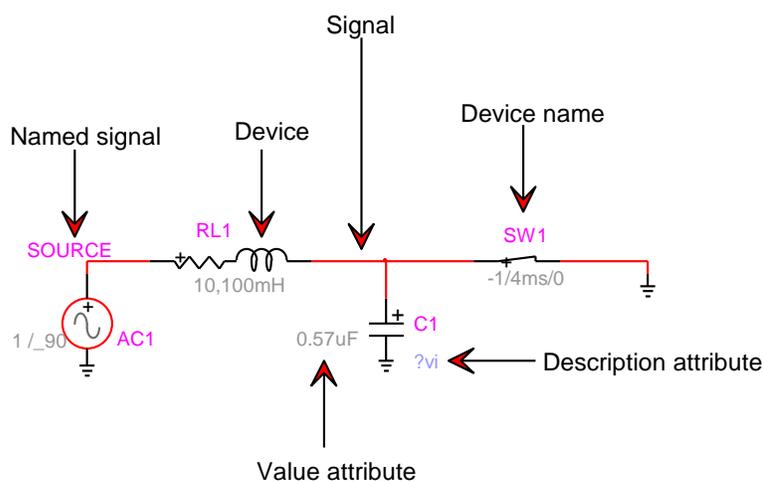
A Symbol Editor function invoked through the device right-click menu "Edit Symbol", can be used to edit (modify) a device's symbol (drawing).

Devices can be given extra functions through customized scripting.

Devices and pin types used in EMTP are described in [Devices Recognized in EMTP](#).

## 6.5.2 Placing a Device from a Library

Devices are objects appearing in a design drawing. Some devices are recognized by EMTP computational engine and some other devices are used for manipulating data or only for drawing purposes. Although generally speaking everything on the screen is a placed object, a distinction must be made between devices, pins, signals and text or picture objects. The following diagram presents a visual summary on the most important objects appearing in a circuit drawing:



Devices have attributes that can be made visible. Devices can be selected from built-in or user-define libraries.

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

- ❑ Select the desired library using the Parts palette. The palette displays only the contents of only one library at a time. If the library file you need is not open, you can open it using the "Open Library" command (right-click on any library or "Options>Libraries>Library Operations").
- ❑ Expand the desired library.
- ❑ Click on the desired part name in the list, hold down the left mouse button and drag it into the current schematic window.

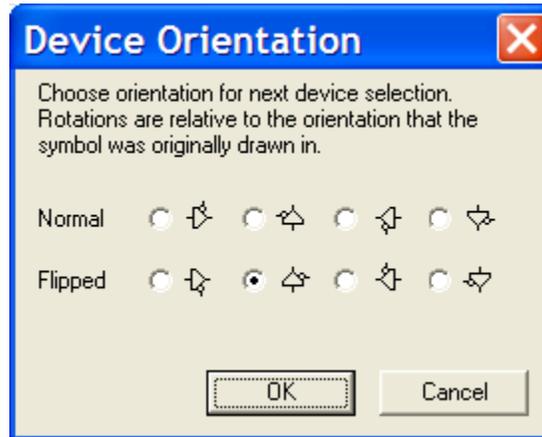
Holding the Ctrl key while clicking will inhibit checking for pin connections. This allows you to select the device again and drag it to a new position without affecting any existing connections.

### 6.5.2.1 Setting Device Orientation

## Reference

Device symbols can be placed in any of 8 different orientations, that is, the 4 major compass points, plus a 180 degree flip around each of those axes. The orientation that a symbol will be placed in is determined by the current orientation setting. This can be controlled in any of the following ways:

- The Orientation command (Ctrl+Q or Home>Operations>Orientation) 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:

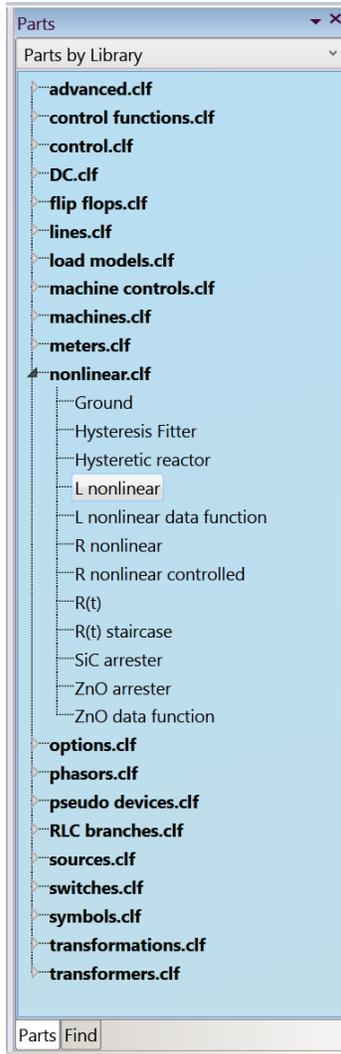


- The orientation can be also set using the arrow keys on the keyboard when a Cut and Paste operation is performed. At the Paste stage, the device is sticking to the mouse and its orientation is modifiable using the arrow keys.

You can also flip and rotate a device symbol after it has been placed in the schematic. See [Flipping and Rotating a Device](#).

### 6.5.3 Using the Parts Palette

Library contents are displayed in a floating palette window which looks like this:



The Parts palette and the Find palette are both turned on in this example and the Parts palette is selected.

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

#### Using the Part Name Filter

The Filter text box becomes available when the user clicks on the “Parts by Library” dropdown menu and selects “All Parts”.



## Reference

The Filter 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. To return to the full selection, delete all characters from this box.

### **Creating or Editing a Part Symbol**

To create or edit a part symbol, right-click in the parts list area and select the New Part or Edit Part commands. The Symbol editor is documented in [Symbol Editor](#) section. It is not allowed to edit parts or add parts into the existing libraries installed with the software, but only to user-defined libraries.

### **Hiding the Parts Palette**

The Parts palette can be removed from the screen by clicking in its close box. To re-display the palette, select the "View>Panels>Parts Library" command.

#### 6.5.4 Selecting a Device

To select a device, the cursor must be in the normal pointer mode. 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.

#### 6.5.5 Duplicating an Existing Device

To duplicate an existing device, either:

- Select a similar device anywhere on the current circuit page and use the Duplicate command (either in the "Home>Edit" menu or in the device pop-up menu or Ctrl+D). The device pop-up menu is available by right-clicking on the device. Use the Spacebar to exit the sticky mouse condition.
- 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 an image of the selected device which can be placed by clicking in the schematic, as discussed above.

#### 6.5.6 Deleting a Device

Devices can be removed by either of these two methods:

- Select the device by clicking on it and then hit the "Delete" key on the keyboard, or select the Delete command from the "Home>Edit" menu.
- Enter Zap mode by selecting the Zap command ("Home>Tools" toolbar) or clicking on the zap tool, then click on the device in question. You can use this tool to delete several devices. To go back to the pointer mode, hit the Spacebar.

#### 6.5.7 Moving a Device

Devices can be moved by either:

- Clicking and dragging to the desired new position, or,
- Using the arrow cursor keys on the keyboard to "nudge" the selected items one grid step at a time.

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.

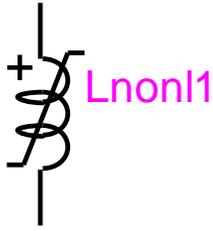
#### 6.5.8 Flipping and Rotating a Device

A set of four commands for flipping and rotating an existing device symbol can be found in the device pop-up menu, i.e. by right-clicking on the device. The Rotate Left/Right and Flip Vertical/Horizontal commands are equivalent to deleting the selected device and replacing it in the selected new orientation.

#### 6.5.9 Displaying and Setting Device Information

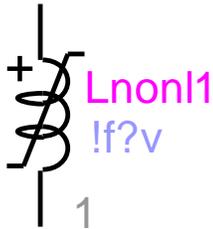
When a device is placed into a drawing it has in addition to its symbol several hotspots (separately selectable sections).

## Reference



Some non-empty sections (attributes), such as the device name, are visible by default.

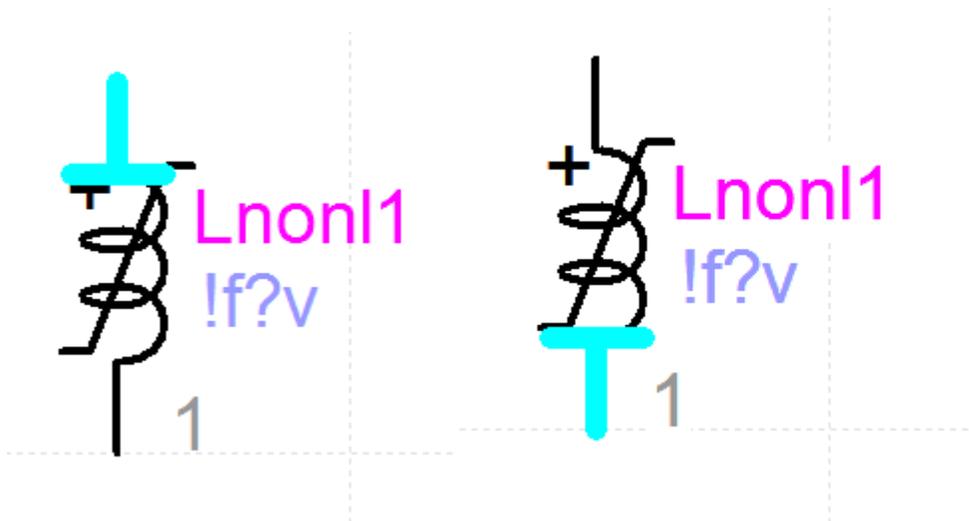
Each visible attribute can be separately clicked, right-clicked and moved. You can also double-click on an attribute and modify its data if it is not a read-only attribute. Visible attributes are used to provide visual information to the user on some device data. In the following example, the Description attribute is given the data "!f?v" which means for this device that it has initial flux and a voltage scope. For this device the Value attribute ("1" in this case) shows the initial flux data.



You can modify visible attribute position by first clicking on the device attribute and then clicking and holding the attribute while moving the mouse. You can also right-click on a given visible attribute to set other properties. In this version of EMTPWorks, empty attributes do not keep their visibility bit.

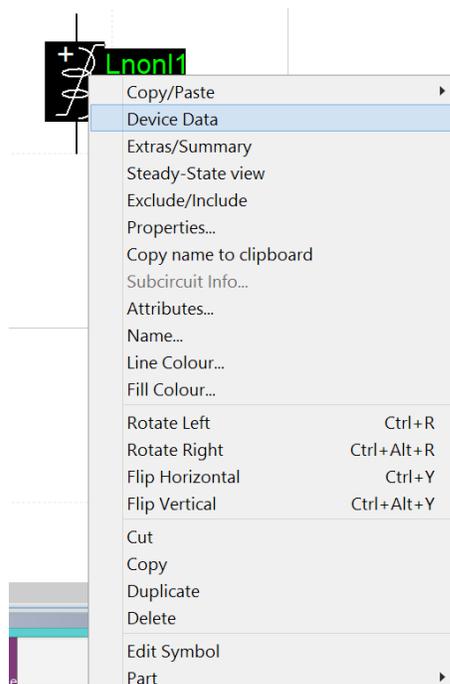
### 6.5.10 Pins

Devices may also have pins, which are also separately selectable by clicking. In this example there are two pins. You can also right-click on a pin for pin data.



### 6.5.11 Device right-click menu

Devices have a right-click menu (right-click on the device symbol):



Most of the above menu items are self-explanatory:

- The Copy/Paste data menu allows copying and pasting device data between identical devices. This is a script that copies data attributes from one device to another.
- The "Device Data" function opens a data input system (web) for entering device model data. It is the same as double-clicking on the device symbol. Each device data web has its own Help section for providing information on the device data, model and/or other functionalities in relation to EMTP or to basic drawing options.
- The "Extras/Summary" function performs various actions according to device documentation. In most cases it opens a panel and presents data summary on the device. This function could be useful in some cases for comparing device data.
- The "Steady-State view" function allows to make selection for visualization of device steady-state solution.
- Exclude/Include: will exclude the selected device from the Netlist, it will not be seen by EMTP and will not be part of the simulation. It becomes an open circuit. Excluded devices have a color code.
- Another information box is opened by selecting the "Properties" command (see section below on Device info).
- Copy name to clipboard: copies the device name to the clipboard. It can be used at a later stage for pasting (using Ctrl+V) into other name input boxes. It is practical to use this function for creating name variations among devices using a root name. It can be also used to name a signal with the name of a connected device.
- If this device is a subcircuit it will have a separate menu item "Subcircuit Info".
- Attributes: Provides access to device attributes.
- Name: Another method for entering the device name.

## Reference

- Line Colour: Changes the line color of the device.
- Fill Colour: Changes the fill color of the device.
- Edit Symbol: invokes the Symbol Editor for this device
- Part: optional item that provides information on the device Part name. This item will be used in a future version of EMTP-EMTPWorks.

The remaining options are self-explanatory.

### 6.5.12 Device Data

The available scripting methods allow modifying and customizing this option for using alternate data input methods. The default method created for EMTP devices is using a data web. This is called the device data web. The data web is organized using data tabs. Data tabs have data input fields. The data web is programmed using DHTML. The appearing web can be located on the user's computer or anywhere on the internet.

If, for example, the RLC device taken from the "RLC branches.clf" library is double-clicked, the following pre-programmed data web is opened.

Properties for RLC RLC1

Values | IC | Scopes | Attributes | Help

RLC branch

R 1 Ω

L 0 H

C 0 F

Select nominal frequency  $f_n$  60 Hz

Frequency scan option

$R = R + R * A * (f/f_n)^B$   $f_n$  is the nominal frequency

R is a function of frequency

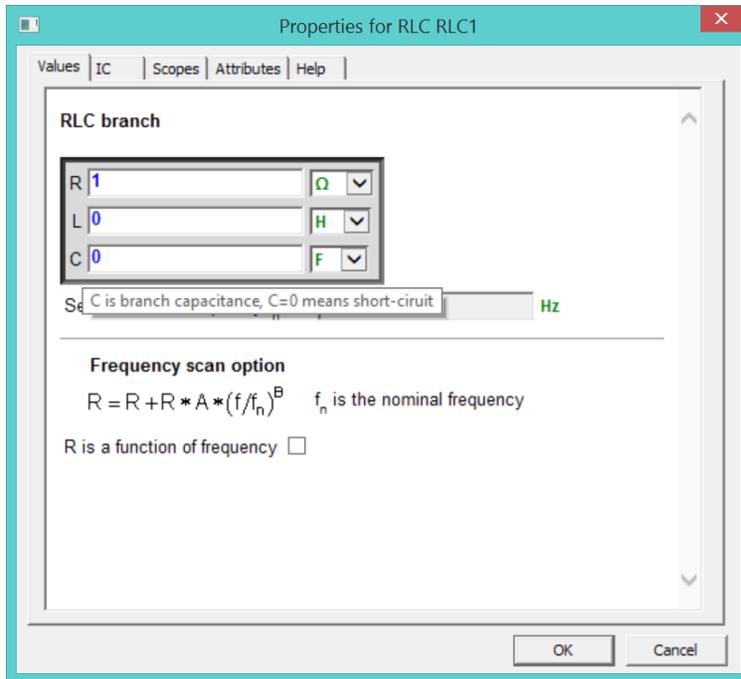
OK Cancel

Data entered in data fields is tested when the user unclicks the field. Data is also tested when moving between tabs and clicking on the OK button. The blue color is for existing data. When a data field is unclicked, the olive color indicates accepted data and the red color signals a corrected problem.

The OK button registers all data changes.

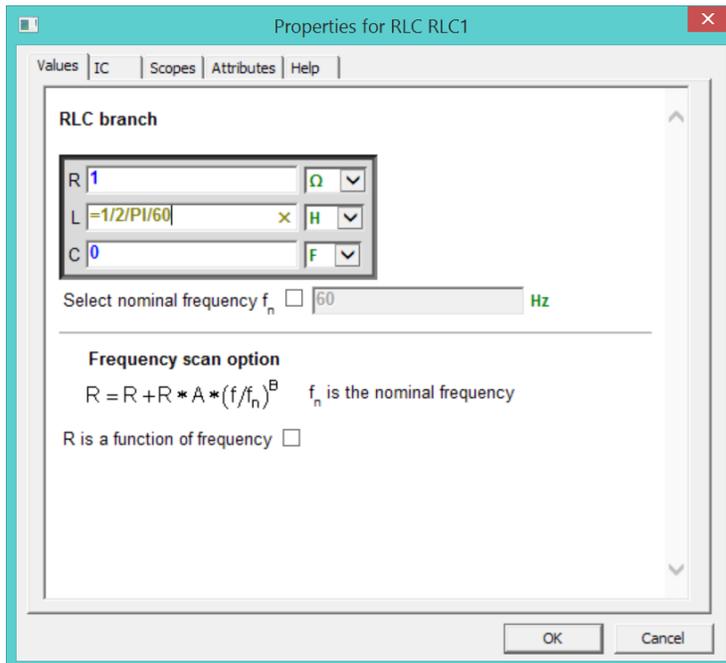
Almost all data fields have default data. Data errors can revert to default data.

Tooltips and hyperlinks are provided everywhere in data tabs. To learn about a given device model parameter (such as L) the user can move the mouse pointer over the parameter's name:



More help is available through hyperlinks and/or in the Help tab or button. Each device is given a Help tab or a Help button for informing on the device data and behavior.

The EMTP data fields have programming options. If the equality sign is the first character in a data field, it can be followed by a mathematical expression. The JavaScript syntax is used. An example is given by entering:



The expression is evaluated when the data field is unclicked.

## Reference

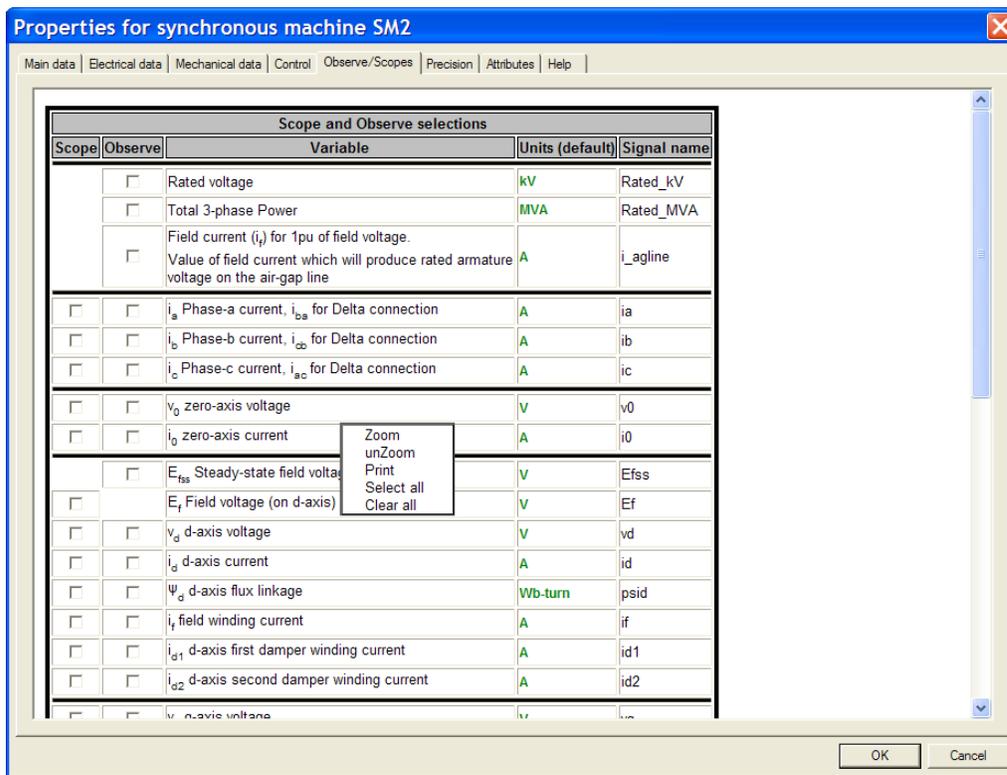
Another programming method is the usage of named values (programmed value or variable data). A named value is a string enclosed between two '#' characters and entered in device data fields instead of entering numeric values. Example:

#Rfault#

This is a data programming method. In this version of EMTP-EMTPWorks it is not allowed to use named values for devices appearing in the top circuit. More details on this approach can be found in the documentation of subnetwork masking methods.

Usage of named values is part of the open architecture methods available in EMTPWorks. Entered data is being determined only after making data substitutions and no testing is available in data tabs. Usage of such options can result in major data errors if the user does not follow the data rules and limitations. Only limited testing is available on EMTP side for the final device data.

Some data forms have a right-click menu for extra options. The right-click menu in the synchronous machine model Observe/Scopes tab is shown below.



### 6.5.13 Device info

When a normal device symbol is selected on the schematic (i.e. not a pseudo- device), then the following information box is displayed using Ctrl+I or the "Properties" command:



The following table lists the information and options available in this box.

Part 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. This is normally only of interest in specialized applications and to the programmer.
Token Number	This is a permanent number that is assigned to this device instance for use in internal EMTPWorks operations and some back annotation and other interface operations. More information on tokens can be found in <a href="#">Device Token Values</a>
Subcircuit Size	If the selected device has a subcircuit, 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 subcircuit (if any) of this device block from being opened for editing.
Pin Info	This button displays the "Pin Info" box for the first pin on the device. Buttons on the Pin Info box allow you to sequence through the other pins on the same device.
Attributes	This button displays the standard attribute edit box for the device. See <a href="#">Entering and Editing Attribute Data -</a>

[Basic Procedure](#) for more information.

NOTE: Clicking Cancel on the Device Info box does not cancel changes that were made in other boxes displayed using "Device Info" option buttons.

#### 6.5.14 Entering Device Attributes

Device attributes are part of the open architecture feature of EMTPWorks. They can be manipulated for advanced usage functions. Device attributes are normally maintained by high-level data input functions available to the user. It could be however convenient to modify them manually in some cases.

To enter device attributes right-click on the device, then select the Attributes command in the pop-up menu. See more information on attributes in [Attributes](#).

#### 6.5.15 Device Attribute probe

The Attribute Probe tool (also called sniffer) can be used to provide quick info on the device. To activate this option, it is needed to click on the "Attribute Probe" tool (Home>Tools) and then single-click on any device. The spacebar deactivates.

This is a script based tool; the scripts are named in the initialization file EMTP.INI (see Attribute Probe Scripts in [\[Drawing\] section](#)).

#### 6.5.16 Device Naming

##### 6.5.16.1 Device Names

Note: 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 "RLC", "TLM", etc.

The name associated with an object can be placed anywhere on the diagram and will be removed if the object is removed.

#### Adding a Device Name by Typing on the Schematic

Enter Name mode either by clicking on the "Name" tool ("Home>Tools") in or by double clicking on the shown name.

Note that the cursor changes to a pencil icon with the "Name" tool. Two techniques are available to determine the position of the name:

- If you click the pencil cursor on a device and release it immediately, the flashing text cursor will jump to the default name location for that symbol. You can then type the name as desired on the keyboard, ending with the Enter key. You can also click directly on the already visible name. If the name is not visible, this procedure will make it visible.

Note: A default position can be specified for any attribute using the methods described in [Using Default Position Fields](#). If no position is specified, the program will calculate one.

- If the name is not visible and if you wish to determine the starting position of the text before you type it, 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 anywhere. Note that if the device name is already visible this method will create another copy of the name on the screen. You can hide the extra copy by right-clicking on it and selecting Hide.

### Adding a Device Name Using a Menu Command

The name can be set by double-clicking on the already visible name or by right-clicking on the device symbol and selecting the Name option. This command displays a simple edit box allowing you to enter or edit the device name. The "Visible" option in this box allows you to select whether the name is made visible on the schematic or simply kept as an invisible text attribute of the device.

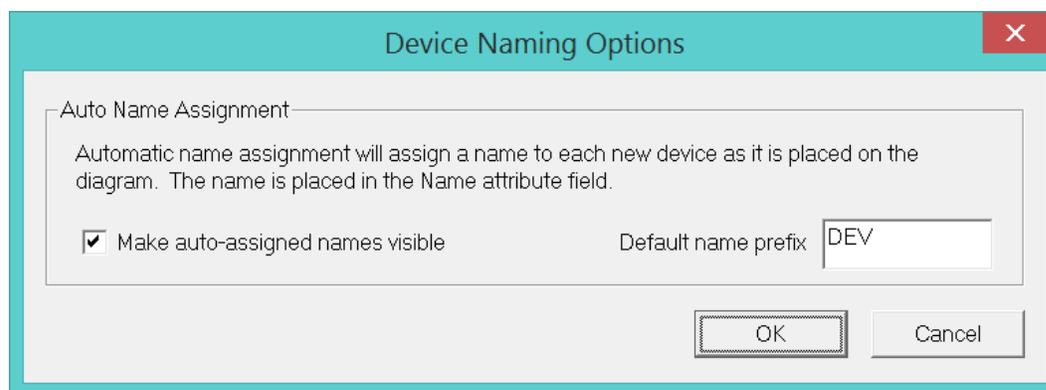
### Repositioning or Removing a Name

Once a name is placed, it can be repositioned by dragging it using the Point cursor or removed using the Zap (Zap tool, "Home>Tools>Zap" or Ctrl+H) cursor. The device name will be removed automatically if the device is removed. Holding the Shift+Ctrl keys depressed while moving a name (or any attribute value) disables the grid snap, allowing you to micro-position the name for alignment with other graphic items on the schematic.

NOTE: The "Name" is actually an attribute field, so all attribute editing techniques can be used on it. In particular, you can right-click on the name and the attribute pop-up menu will appear, giving you a variety of editing commands, described in Attribute text right-click menu.

### 6.5.16.2 Device Naming Options

Selecting the "Device Naming Options" command from "Options>Naming Options" will display this box for device Auto-name options:



- |                                  |  |
|----------------------------------|--|
| Make auto-assigned names visible | If this box is checked, the auto-assigned name will be displayed on the schematic next to the symbol. If this box is not checked, the name will be stored with the device, but not displayed on the schematic. |
| Default name prefix              | The text in this box will be the default prefix used to generate a device name, if no prefix is provided in the device attributes.   |

### 6.5.16.3 Setting the Auto-Generated Name Format

An auto-generated device name consists of two parts, the fixed prefix and the numeric suffix. The prefix portion is derived from one of two sources:

- The device's Name.Prefix attribute field, or if that is empty,
- the value set using the "Device Naming Options" menu command shown above.

The numeric suffix is assigned sequentially for each different prefix found in the design. The names "U1", "U2", "U3", etc. will be assigned to devices with the value "U" in their prefix field, and the names "Res1", "Res2", "Res3", etc. for devices having prefix "Res".

The default device prefix field is Name.Prefix. This can be changed by entering the name of any other field in the design's PrefixField field (Design attributes, use right-click menu on the design). As an advanced option, you can change the PrefixField field and use a different device prefix field when changing the design type.

### 6.5.16.4 Reassign Names

The "Reassign Names" command is available from "Options>Naming Options".

The Reassign Device Names command 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 the following rules:

- Only the current circuit is affected. Other circuit levels higher or lower in the hierarchy are 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.
- Names that have been edited by the user are no longer considered "default" names and will not be changed.
- Devices that are explicitly set to be Unnamed will also become named using the \$NONAME keyword (becomes used as a prefix) discussed in [Specifying that a Device Should be Unnamed When Placed](#).

### 6.5.16.5 Setting the Name Prefix for a Device

The name prefix is set by filling in the appropriate device attribute field Name.Prefix. It can be done after placing the device in the design or by editing the device symbol when it is in its library. In the first case the name will start changing after changing Name.Prefix, in the second case it will appear with the desired prefix immediately after dragging in from the library.

- ❖ See [Setting Part Attributes](#) for more information on using this command.

### 6.5.16.6 Specifying that a Device Should be Unnamed When Placed

In some cases it may be desirable to have a device remain unnamed when it is placed on a schematic. An example might be a simulation option device that is used as data input function for some global information on design parameters.

The keyword "\$NONAME" can be placed in the Name.Prefix field in a device symbol to indicate that no name should be assigned when the symbol is placed in the diagram.

### 6.5.16.7 Overriding Default Name visibility

Normally, when a device is placed on the schematic, its name will be made visible when the values are automatically assigned. This can be turned off globally using the Device Naming Options command. In some cases, it may be desirable to override this for specific devices.

To override the default visibility for a device:

- ❑ Open the device in the symbol editor: right-click on the device name in the “Parts” palette and select “Edit Part” or right-click on the device in the schematic and select “Edit Symbol”.
- ❑ Select the “Part Attributes” command in the Operations toolbar.
- ❑ Select the attribute field in question: Name.
- ❑ Enter a dummy value in the value text box. This value will be overwritten by the packager in any case, but a non-empty value must be placed here to force the attribute field settings to be stored with the symbol.
- ❑ Select the “Always Visible” or “Never Visible” option, as desired.
- ❑ Click OK.
- ❑ Resave the part.

#### 6.5.16.8 Device Token Values

Every time a device is created in an EMTPWorks circuit, it is assigned an integer value known as its “token” (See “Token number” from device right-click menu Properties). The token number stays with the device for its lifetime and numbers are not re-used. This ensures that a given device can always be recognized despite duplicate names or name changes. The token is used for a number of internal operations in EMTPWorks, but can also be seen by the user in the following circumstances:

- The token number is displayed in the Device\_info box.
- The token number can be written out in Netlists 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.
- 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.

#### 6.5.16.9 Device Date Stamping

Date stamping is a mechanism that automatically marks devices with a time value when they are created.

The date stamp is stored using the system’s internal integer date format, that is, an unsigned integer representing the number of seconds since January 1, 1970. When a device is created or undergoes any major editing operation, the current time value is converted to a decimal string and assigned to the DateStamp.Dev attribute field. To create a more human-readable date value, the Export tool has date conversion functions available.

- ❖ See more information on date conversions in report scripts, see the section Date and Time References in [Export Script Language](#)

#### 6.5.16.10 Disabling Date Stamping

## Reference

The date stamping process incurs a small overhead in processing time and memory space. If you are not using this feature and wish to disable all internal processing related to it, you can place the following line in the initialization file EMTP.INI under the drawing section:

NODATESTAMP

This will take effect the next time you start the EMTPWorks program.

- ❖ See Device date stamping in INI file in the [\[Drawing\]](#) section.

## 6.6 Devices Recognized in EMTP

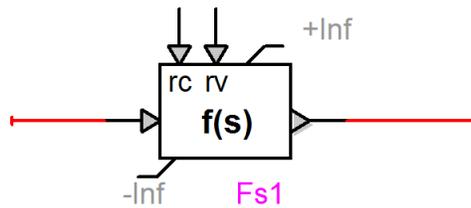
### 6.6.1 Devices Recognized in EMTP

A design circuit can hold various device types. EMTP can recognize option devices, and simulated devices. There are also model data function devices.

The primitive devices are recognized in EMTP using the Part device attribute. The primitive devices can be of power type or control type. A power type device is for modeling an actual physical electrical component. A primitive control type device is a block diagram device. It is a higher level device that can be used to simulate actual control system behavior or for creating electrical network models using function blocks.

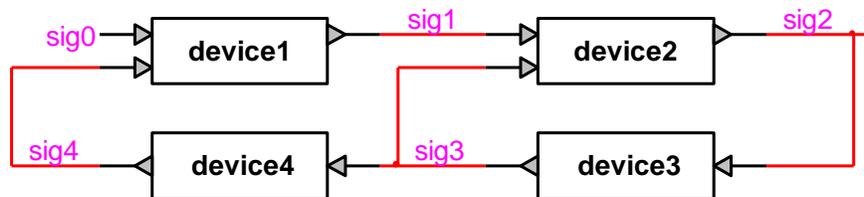
Power type devices are devices with power pins.

Control type devices are devices with control pins. The control type devices are interfaced with power type devices using sensors (meters) and actuators.

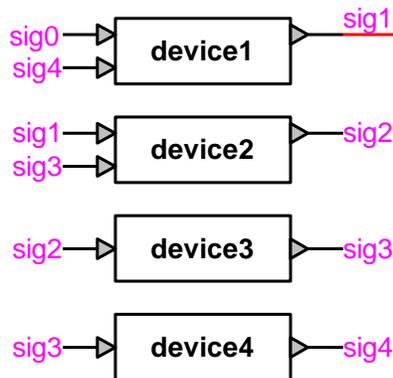


Control device

A control component is built in EMTPWorks as a directed graph containing elements joined by oriented signals, together forming a block-diagram description of the component to be represented. An example is given by:



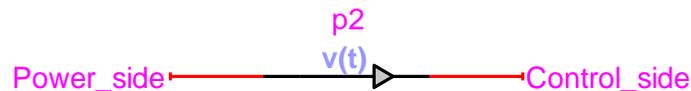
In a lot of designs it is preferable to present the control function in a procedural manner. This can be achieved using the virtual connection method described in [\[Connecting Signals by Name\]](#). This design is functionally identical to the one above.



### 6.6.2 Device pin types

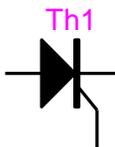
There are currently 3 pin types (also called Pin Functions): Power-pin, Input-pin and Output-pin. The Power-pin is for power devices. A control device may have input pins and/or output pins.

Signals connected to control device pins are control signals and do not connect to power pins directly, meter (sensor) functions are needed. An example of sensor taken from meters.clf library is shown in the figure below. It has a power pin that allows connecting to a power signal. It also has an output pin that allows connecting into a control device.



The control pins are most often shown with an arrow symbol for showing directionality. The current version of EMTPWorks does not automatically provide this functionality when subcircuits are used.

Some devices may have control and power pins. This is the case of the controlled thyristor found in the switches.clf library:



It has two power pins and one control pin which is for the firing signal.

Subcircuits may have mixed pin types. Signals connected to output pins can only connect to input pins.

Pins that can be interconnected are called compatible pins.

Device pin settings are made during the device creation using the Symbol Editor.

## Reference

EMTPWorks performs automatic checking for compatible pin types when connecting pins through signals. This checking is not available when virtual connections are used (see [Connecting Signals by Name](#)).

### 6.6.3 Built-in libraries

The available libraries (see Parts window) are based on primitive devices and masked devices. The primitive devices are those that are directly recognized in EMTP. The masked devices are based on subcircuits created from primitive devices used as building blocks. Users can build their own masked devices using primitive devices or available masked devices, or using DLL programming methods.

All primitive devices have a Help tab for detailed documentation on the model and model usage.

Most masked devices have documentation available in the mask; you can also access the contents of a subcircuit for more documentation on its design and functionalities.

You may also use the Attribute Probe “Home>Tools” to get quick functional information about a device.

Some subcircuits may be locked; you can unlock them using the method described in [Locking and Unlocking Subcircuits](#).

If you modify a device from this library after placing into the design, it will become different from its original library version and its functionality will be altered. See also [Subcircuit Uniqueness and Encapsulation](#).

The built-in libraries are locked (the subcircuit file is read-only) to avoid corruption. If you want to create your own libraries you should place them in your work folders and maintain them separately. More information on libraries is available in [Working with Symbol Libraries and Lib Maintenance](#).

## 6.7 Creating and Editing Signals

### 6.7.1 Signals

In EMTPWorks, a signal represents the electrical connection between any number of device pins. A signal can simply be represented on a schematic by a single line or a number of connected line segments, or more complex structures such as connection by name, busses, bundles, phases or cross-page connectors and hierarchy blocks can be used to simplify the representation of large designs. In this chapter, we will look at simple signals and connection by name within a single circuit page. Bundles and multipage signal interconnection schemes are covered in Bundles and Inter-page Connections respectively.

### 6.7.2 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. If both of the two signals being connected were named manually and not automatically, 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.

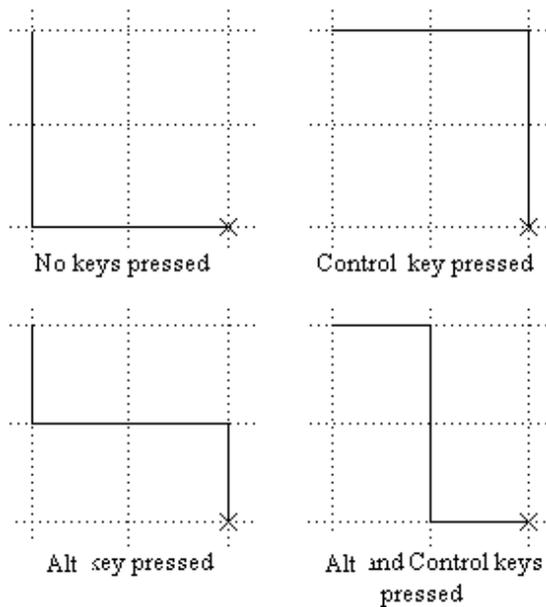
In addition to connecting by drawing a signal, a device can be connected to a signal by moving the device near the signal and touching the signal with its compatible pin. This is called connect-by-proximity. Under some circumstances when entire circuit sections or devices are moved near or over other circuit sections, the connect-by-proximity can become a nuisance, and can be turned-off by holding down the Ctrl key after starting the move.

#### 6.7.2.1 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 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 Ctrl and Alt keys after starting to draw, as follows:

## Reference



The Shift key constrains the movement to a single vertical or horizontal line. The Ctrl key inverts the order of line drawing, and the Alt key switches to three line segments with a center break.

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

### 6.7.2.2 Creating an Unconnected Signal Line

The Draw Signal cursor (“Home>Tools”) 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.

Note that the tip of a signal which is not physically connected is a T-shape.

### 6.7.2.3 Editing a Signal Line

The following features are available to edit signal lines:

- Zap mode (entered by selecting the Zap command in “Home>Tools”) 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 and up to the nearest intersection, device pin or segment join point.
- Selecting a signal line (by clicking anywhere along its length) and hitting Delete or 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 with the pointer cursor allows you to reposition the line. Vertical lines can be moved horizontally and vice versa.
- The pointer cursor can be used to start drawing from the ends or corners of an existing signal.
- The Draw Signal cursor can be used to start drawing from anywhere along an existing signal line. Double-clicking terminates drawing.

### 6.7.2.4 Checking connectivity

A single-click on a signal highlights the signal and shows its connectivity.

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

Holding the Ctrl button and double-clicking on a device selects the entire interconnected circuit, but not the virtually connected circuit (see [Connecting Signals by Name](#)). The selected circuit can be moved around using the mouse pointer (hold down any device and move) or the keyboard arrow keys.

You can also use the "Pin List" command in the signal pop-up menu (by right-clicking on the signal) to view the list of pins attached to the signal and go to any of them. 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.



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 bundle internal pin) the window will be centered on the parent bundle pin.

For convenience in navigating a schematic, connections to pseudo-devices such as bundle 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 using the Design Preferences command ("Options>Design").

- ❖ For more information on page references, see [Inter-page Connections](#).

### 6.7.3 Setting Signal Color

To select an individual color for a single signal line, right-click on the line and select the Color command in the popup menu. This setting will override the default signal color, which is determined by settings in the initialization file, described in [\[Drawing\]](#) section. To return a signal to the default color, select the Color command again and click the Default button.

### 6.7.4 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 bundle.

### 6.7.5 Selecting a Pin

A pin is selected by clicking on the pin line close to the device.

Since an **unconnected device pin is both a pin and a signal**, you determine whether you get the pin or signal pop-up menu as follows:

- right-clicking on the pin in the last 1/4 of the pin length away from the device will display the signal menu.
- right-clicking on the pin close to the device symbol will display the pin menu.

### 6.7.6 Pin right-click menu

The pin right-click menu has the following commands:

- Ground: This command grounds the pin. The attached signal is named GND if the pin is a power pin and it is name '0' (zero) if the pin is a control input pin. The signal name must be kept be visible for forcing and guaranteeing its name.
- Pin Info: Opens the pin information panel.
- Attributes: Shows pin attributes.
- Bundle Pin Info: For bundle pins (see [Changing Bundle Pin Connections: Bundle Pin Info](#)).

### 6.7.7 Signal right-click menu

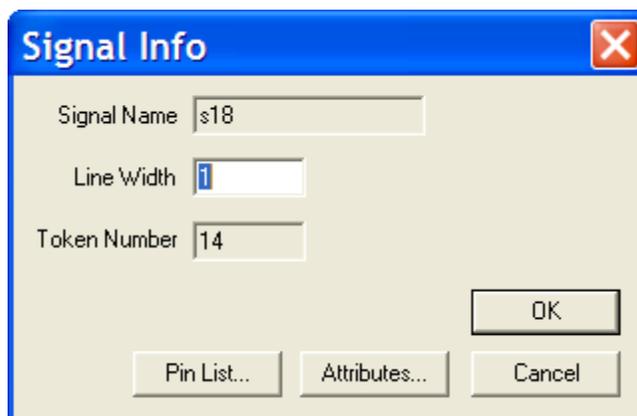
The signal right-click menu has the following commands:

- Signal Parameters: Opens a menu for setting extra signal parameters (see [Signal Parameters](#))
- Copy name to clipboard: Copies the signal name to the clipboard. It can be pasted at a later stage using Ctrl+V. This option can be used for naming a device connected to the signal.
- Properties: Opens the "Signal Info" box shown below.
- Attributes: Shows pin attributes.
- Name: Allows entering signal name.
- Color: allows changing signal color (see [Setting Signal Color](#))
- Pin List: Shows the list of pins connected to the selected signal (see [Interconnecting Signals](#))
- Line Type: Allows setting the signal lie type (see Signal Line Type)
- Cut: cuts the current signal
- Copy: copies the current signal

- Duplicate: duplicates the current signal
- Delete: deletes the current signal

### 6.7.8 Signal Info

Selecting the Properties command (right-click menu) with a signal selected causes the following box to be displayed:



Here are the options presented in this box:

- |            |   |
|------------|---|
| 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.  |
| 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 <a href="#">Entering and Editing Attribute Data - Basic Procedure</a> |

Every time a device or signal is created in a 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 EMTPWorks, 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 in this chapter.
- The token number can be written out in Netlists or bills of materials whenever a guaranteed-unique identifier is needed.

Note these 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.

### 6.7.9 Signal Parameters

This command from the right-click signal menu can be used to enter extra information for the selected signal. Help is available in the help tab of this command data web.

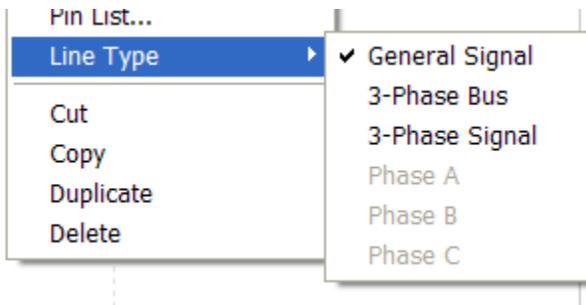
### 6.7.10 Signal and Pin Attribute probe

The "Attribute Probe" tool (also called sniffer, "Home>Tools") can be used to provide quick info on signals and pins. To activate this option, it is needed to click on the Attribute Probe tool and then single-click on any signal or pin. The spacebar deactivates.

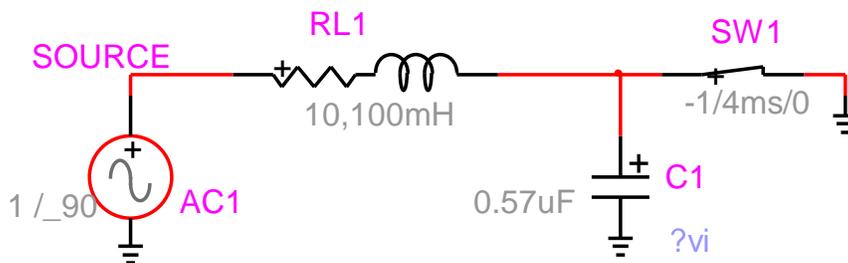
This is a script based tool; the scripts are named in the initialization file EMTP.INI (see Attribute Probe Scripts in [\[Drawing\]](#) section).

### 6.7.11 Signal Line Type

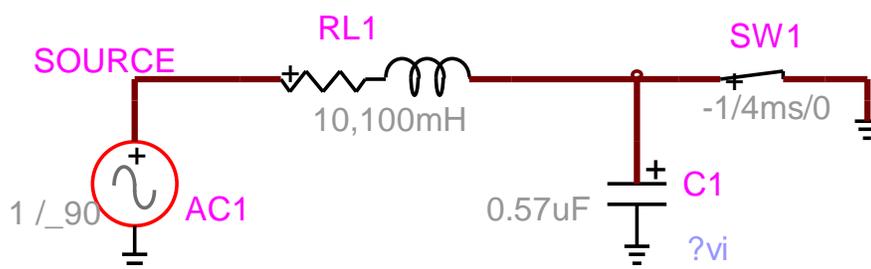
Each signal has right-click menu item "Line Type"



The default signal is called a general signal. It is a 1-phase signal. It can be drawn by starting from an existing signal or device pin or by using the "Draw signal tool" (see "Home>Tools"). This circuit is created using only the "General Signal" line type.



If any signal of the above circuit is selected and its "Line Type" is changed to "3-Phase Signal", then EMTPWorks will propagate the new setting through the entire interconnected circuit.



This propagation is only acceptable if all interconnected devices accept the new phase setting, an error message will result otherwise. The user can go back by selecting Undo (Ctrl+Z), the 1-phase devices will become otherwise disconnected in the Netlist and errors may occur.

Some power devices have pins that accept only 1-phase or 3-phase signals.

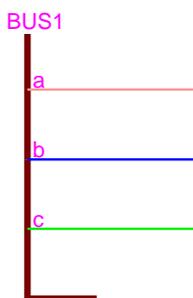
Devices that can become 3-phase will automatically adjust the data input functions to allow entering 3-phase data.

A 3-phase signal internally carries 3 names, one for each phase. For the signal SOURCE shown in the figure above, the internal names are SOURCEa, SOURCEb and SOURCEc. The same is done for device names. A 3-phase device can be decoupled or internally coupled.

It is allowed to move back to the 1-phase version of a signal, again using the signal "Line Type" option.

A control signal can only be a general signal. A power signal can be a: 3-Phase Signal, 3-Phase Bus, Phase A signal, Phase B signal or Phase C signal.

The signal drawn from a pin takes its phase property. The user can also create separate signals using the signal drawing tool "Draw 3-phase Bus" (see "Home>Tools>Draw Signal"). A 3-phase bus can only connect to 3-phase signals. It is also allowed connecting to individual phases:



Phase signals are drawn using the phase signal commands ("Home>Tools>Draw Signal"). Phase signals can be also created by converting a 3-phase signal line type to "Phase A" (or B or C) type.

The 3-phase bus acts as a signal concentrator; it has only one name, but appends the phase character to each phase signal, such as BUS1a, BUS1b and BUS1c. If a phase name is changed then the parent bus name is also changed.

Naming a phase signal or bus GND will ground all phase signals. The ground signal GND is not replicated using phase characters.

## Reference

If a phase signal is drawn separately in a design using the "Draw Phase" tool, then its name is internally concatenated with the phase character. In this example:



the actual signal name is BLUEb.

More details on 3-phase and 1-phase signals are given in the design three\_and\_one\_phase\_diagrams.ecf (see EMTPWorks directory Examples\ShowHow).

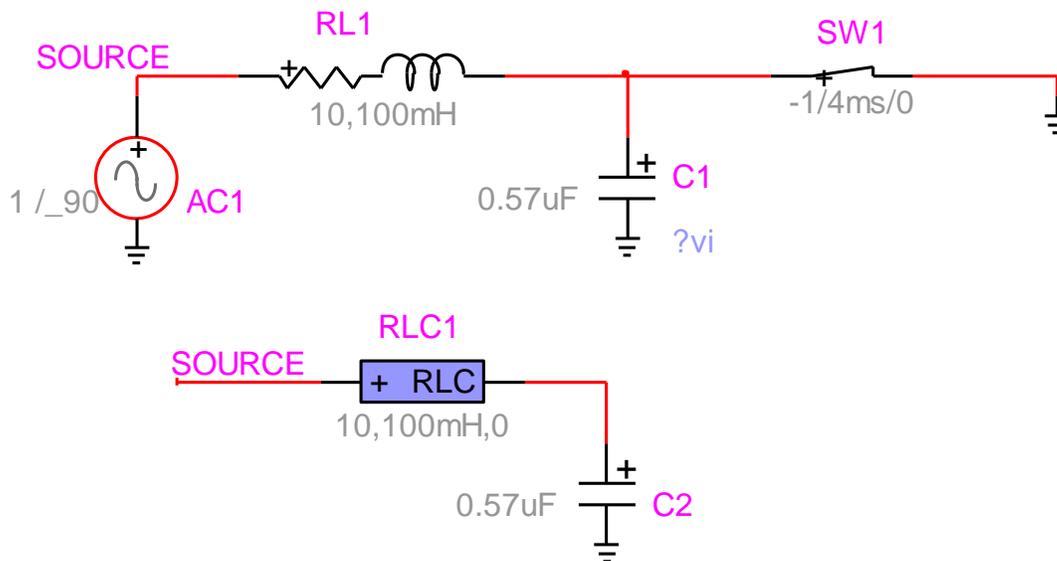
### 6.7.12 Connecting Signals by Name

Signal names can be used to make logical connections between wires that are not visually connected on the schematic. Like-named signal traces on a single page are 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.

Connection by name is also applicable across design pages.

Connection by name is also called virtual or logical connection.

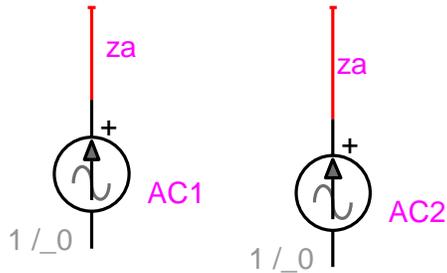
In this example the top pin of the source AC1 is connected to the positive pin of RLC1 using the signal name SOURCE. If you double-click on the SOURCE signal it will show its virtual connectivity.



A signal name must be visible to keep its name. The current version of EMTPWorks has the freedom to change user given signal names during rerouting if not visible. EMTPWorks maintains a default signal naming system.

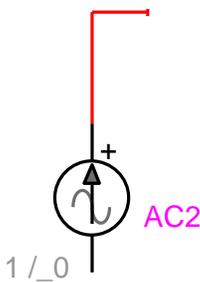
It is recommended to use distinctive names when showing a signal name. You must avoid using default signal names used in EMTPWorks. The Signal Auto-Naming Options are used to set the default naming methods. There are also Reserved signal names.

To undo connection by name it is not sufficient to hide a signal name. In the Example below the two current sources have their positive pins connected by name.



If you hide the name `za` on the signal of AC2, the connection will still remain in the subsequently generated Netlist. Two actions can be taken to ensure disconnection:

- Hide the name and reroute the signal (see figure below).
- Change the signal name to another name, a name that does not exist in your design. For example, `za` can be renamed to `za_`. Then hide the name.



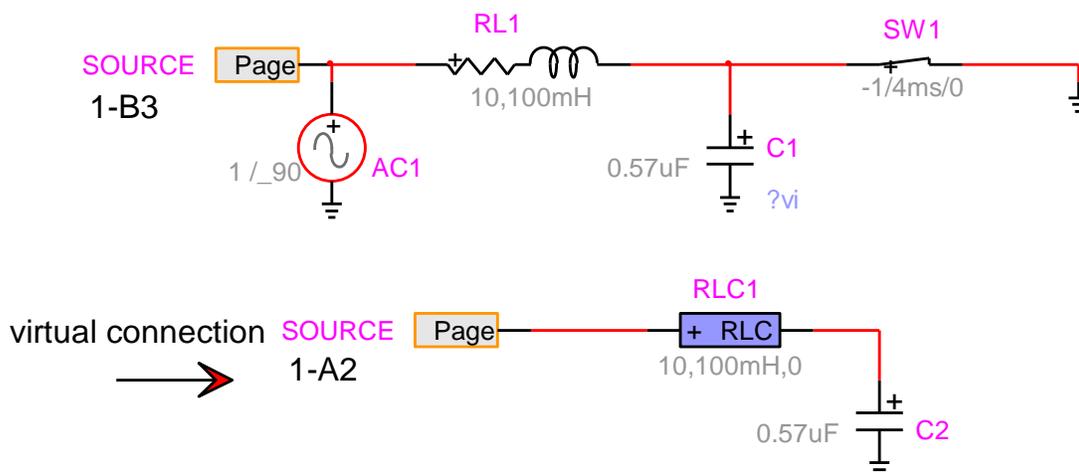
It is recommended to verify the signal name (Name edition panel) after hiding it, to make sure that the disconnection has worked.

Generally speaking it is not very useful to name individual signals manually and to make them visible, unless you need to create a special reference or apply connection by name. Most designs can be carried on without naming any signal. The power system bus name is on by default since in this case the name is a useful reference in the power network diagram.

Instead of using signal names for connecting by name, another strongly recommended approach is to use page connectors. This approach is more robust and less user error prone if the user decides to hide signal names and disconnect signals connected by name.

Page connectors can be used for connecting signals between pages or signals appearing on the same page. The example shown above is now shown using page connectors.

## Reference



This approach is more robust and provides visual location feedback (automatic display of page references) on the locations of connected signals. In this case it indicates that 1-B3 (coordinates using design sheet border numbers) the connected signal is found on page 1 at the geographical location B3. The same applies to all pages connectors. The connection is created using the same name for the two page connectors. Two or more page connectors can be used.

The page connector devices are found in the library “pseudo devices.clf”. The page connector is named using the right-click menu Name.

To turn on the visual location feedback it is needed to go to the menu “Options>Design>Design Preferences” and select the “Page Refs” tab. In this tab the checkbox “Enable Automatic Page References” must be checked.

A right-click menu item Properties on the page connector device allows using a Go To button to jump between connected page connectors.

**You must clearly understand the rules and consequences on maintaining signal names before heavy usage of such methods. The case of signal names in bundles is more complex and it should be avoided when possible.**

### 6.7.13 Naming Signals

The name associated with a signal can be placed anywhere on the diagram and will be removed if the object is removed. You can click on the name to highlight the associated signal or click on the signal to highlight the associated name. By default all signal names are invisible.

A signal name must be visible to keep its name. The current version of EMTPWorks has the freedom to change user given signal names during rerouting if not visible. EMTPWorks maintains a default signal naming system.

Generally speaking it is not very useful to name individual signals manually and to make them visible, unless you need to create a special reference or apply connection by name. Most designs can be carried on without naming any signal. The power system bus name is on by default since in this case the name is a useful reference in the power network diagram.

**You must clearly understand the rules and consequences on maintaining signal names before heavy usage of such methods. The case of signal names in bundles is more complex and it should be avoided when possible.**

Scopes for simulation waveform visualization are based on device names and not signal names.

Signals connected to control (input or output) pins are control signals. Signals connected to power pins are power signals. See Device pin types used for EMTP.

### 6.7.13.1 Reserved signal names

The currently reserved power pin signal name is GND. If a signal is named GND then it is automatically connected to ground (zero-volts). This is only true for power signals.

For control devices the currently reserved signal name is the number 0. If an input signal is given the name 0 then it will take the value 0 for the entire simulation.

If an input signal (connected to an input pin) is not connected to any output signal, EMTP assumes that its value is always zero.

### 6.7.13.2 Signal connector devices

EMTPWorks also provides signal connector devices. A built-in signal connector device is the "Ground" device:



The "Ground" device has one pin and its signal name is GND:



The GND name is an enforced signal name. It is the pin name of the "Ground" device which is given the primitive Part type "Signal Connector" in the Symbol Editor. It is illegal to change the name of such a signal manually.

The user can create other signal connector devices using the Symbol Editor. This is useful for making connections by fixing a given signal name through the pin name. In this example the fixed signal name is SOURCE:



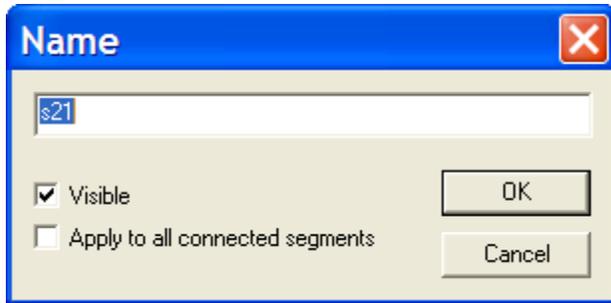
### 6.7.13.3 Entering and editing a Signal Name

Enter "Naming" mode either by right-clicking on the signal and selecting Name or by clicking on the Name tool in "Home>Tools>Name".

Once the Name tool command 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 anywhere.

When a signal name is visible, you can double-click on it to open the Name edition panel:



The following options are available:

- "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.
- "Apply to all connected segments": 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. You must select this option if you have signals connected by name (virtual connection).

**Important:** If you make a signal name invisible you lose control over its name management in EMTPWorks. EMTPWorks may change the name internally for reordering purposes. The signal connection by name (virtual connection) will not work if the name of a signal is not visible.

In order to minimize the chance of creating an accidental short between two signals with the same name, the EMTPWorks auto-naming feature will always assign a new, unique name to any signal that has an invisible name that is copied, duplicated or otherwise edited on the schematic.

A set of standard right-click menu options is also applicable to signal names.

#### 6.7.13.4 Moving a Signal Name

A signal name can be moved by selecting the pointer (normal) cursor, clicking and holding the mouse button on the name, and dragging it to the desired new position. Holding the Ctrl and Shift keys depressed while moving a name (or any attribute value) disables the grid snap, allowing you to micro-position the name for alignment with other graphic items on the schematic.

The "Name" is actually an attribute field, so all attribute editing techniques can be used on it.

#### 6.7.13.5 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:

- ❑ Select Name mode (using the Naming tool)
- ❑ 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 once 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 ("Home>Tools") or Hide in the right-click menu. If you remove the last visible name from a signal segment then the logical connectivity to other like-named signals is removed.

### 6.7.13.6 Signal Auto-Naming Options

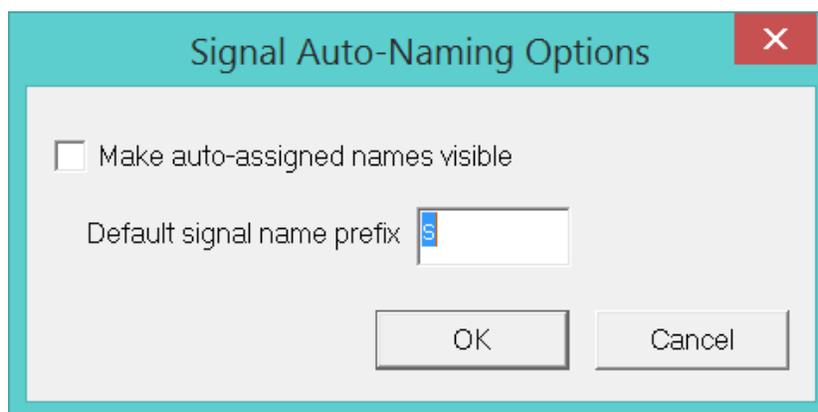
EMTPWorks has an automatic name assignment feature that ensures that every signal object that is created has a distinct name. Signal auto-naming is enabled by default when a new design is created, unless the selected template has specifically disabled it.

When Auto-Naming is enabled, all signals will be given a default name when they are created. In addition, whenever an editing operation causes a single signal to be broken into two, or part of a signal to be copied without a visible name label, a new, unique name will be auto-assigned. This ensures that every object in a design has a unique name.

The program generates new names using a specified prefix and the signal's token number, which will produce a unique name in most cases. However, the program does not check for uniqueness. If you have manually assigned a name to another signal that matches the auto-assigned format, there is the possibility that an auto-assigned name may already exist.

#### Auto-Naming options

Auto-Naming options are available with the “Options>Signal Naming Options” command:



If the “Make auto-assigned names visible” option is selected, the default name will be displayed on the signal.

The “Default signal name prefix” text box allows you to specify the characters to be used as a basis for auto-generated names. Changing this value does not affect existing names.

#### How Names are Generated

The auto-generated name consists of two parts, the fixed prefix and the numeric suffix. The prefix portion is derived from the signal name prefix.

The numeric portion of the name is generated from the 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.

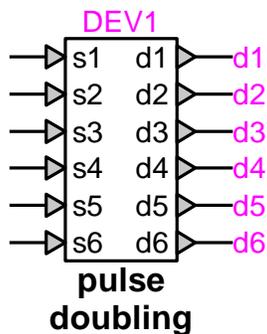
**The program does not check names for uniqueness. If you have manually assigned a name to another signal that matches the auto-assigned format, there is the possibility that an auto-assigned name may already exist.**

### 6.7.13.7 Using the Auto-Naming Features

Two features are available to simplify the naming of groups of related signals. These features are enabled after entering the signal name mode using the Naming tool.

- Auto-alignment - If the Alt 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.
- Auto name generation - If the Ctrl and Shift keys are held down while a signal 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 decremented.

The above method allows, for example, creating the sequential names for the device shown below.



### 6.7.14 Bundles

The bundle facility in EMTPWorks 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 bundle line.

A bundle can be also called a signal bus, but such a bus is different from a 3-phase power system bus. In some cases EMTPWorks messages and help section may mention the word "bus" for bundle.

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

- Bundle pins on devices can be connected directly to a bundle. Connections can be made by using a breakout to access the desired internal signals. A breakout is created using the "New Breakout" command in the "Home>Tools" menu. Connections can be also made directly by drawing a signal into a bundle. When a signal is drawn into a bundle, the "Bundle Connection" menu appears and allows to make connections to existing signals or establishing a new signal.
- You do not need to specify in advance what signals will be contained in a given bundle. Any signals that are present in a breakout or bundle pin attached to a bundle will become part of that bundle and can be brought out through another breakout anywhere along the bundle.
- Any two bundles can be joined together, regardless of their internal signals. When two different bundles are merged, any signal in either bundle becomes available anywhere

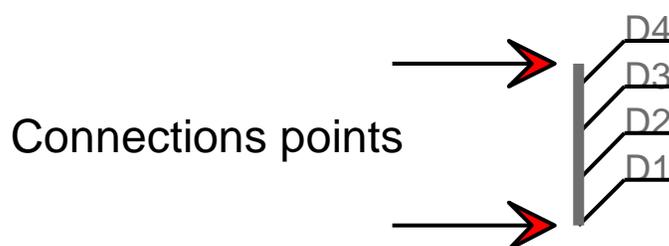
along the combined bundle. The "Bundle Connection" menu is used to connect signals when connecting bundles together.

- If you select a bundle line, then do a "Bundle Info" command (right-click menu), the displayed info box will show a list of signals currently contained in the bundle.
- A given signal can be present only in one bundle. If you attempt to connect together two signals in different bundles using breakout pins, a warning box will be displayed and the connection will be canceled.
- A bundle can be created by drawing the bundle lines first, then creating the breakouts to attach, or by creating a breakout and extending the bundle line starting at the bundle pin. Bundle lines are drawn or extended using exactly the same techniques as for signals, except that the "Draw Bundle" command ("Home>Tools>Draw Signal") or cursor is used instead of "Draw Signal".

#### 6.7.14.1 Creating a Bundle

A bundle can be created by any one of these methods:

- Select the "Draw Bundle" tool. Draw any desired contiguous set of lines on the diagram using the usual signal drawing techniques. This bundle will have no internal signals initially. Signals will be added implicitly when it is connected to any breakout or bundle pin.
- Create a breakout symbol using the "New Breakout" ("Home>Tools") or Ctrl+B command discussed in [Using Bundle Breakouts: New Breakout](#). The bundle pin (backbone) of the breakout can now be extended using the normal pointer or the Draw Bundle cursor. The bundle will contain all signals specified in the breakout.

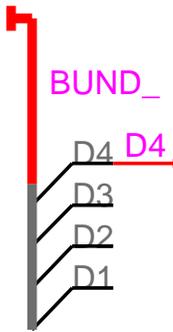


- Extend a line out from an existing bundle pin on a device using the normal pointer or the Draw Bundle cursor. The bundle will contain all signals specified in the bundle pin on the device. Connections between bundle internal pins and bundle internal signals can be changed using the "Bundle Pin Info" command on a bundle pin's pop-up menu. See [Using Bundle Pins](#) for more information on bundle pins.

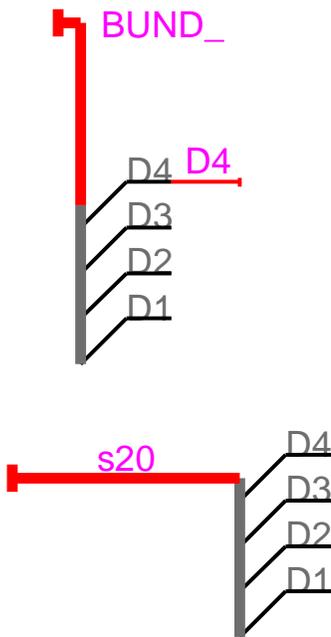
#### 6.7.14.2 Bundle signal names

It is important to remember that bundle signals names are preceded by the bundle name. In the example shown below, the actual bundle signal as seen in EMTP is BUND\_D4 and not D4.

## Reference



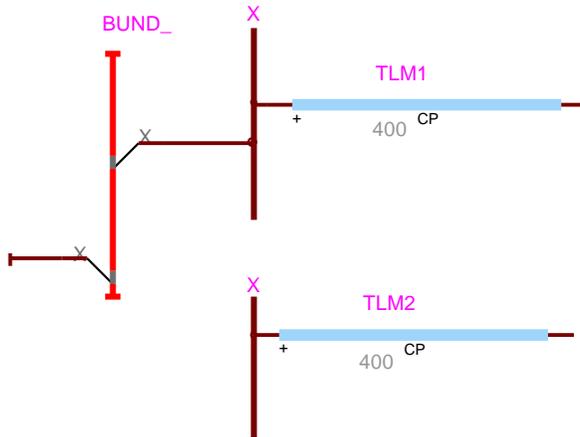
The breakouts must be interconnected through bundles to interconnect signals. In this example, there are two distinct sets of signals since the holding bundles do not have the same names.



**Attention:** only alphanumeric (numbers are letters) characters are acceptable for bundle signal names.

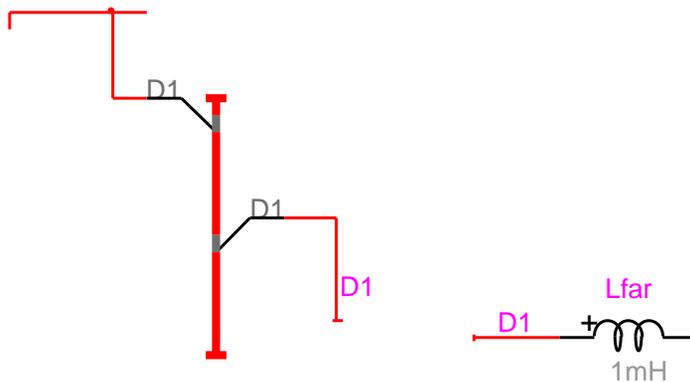
The signal name notation that appears on a breakout pin is actually a pin attribute. It can therefore be edited by the usual attribute editing mechanisms. You can double-click on the name and change it.

You must always remember that the signal connecting to a bundle pin automatically inherits the bundle pin name prefixed by the bundle name. This is also true for existing signals. In the example below, the bus named X existed before connecting to the bundle pin X. When it connects to the pin, its name automatically changes to BUND\_X. This is also true for the other bus named X which was connected by name. The two buses remain connected. If you delete the connection to the bundle pin, the bus name will revert back to X.



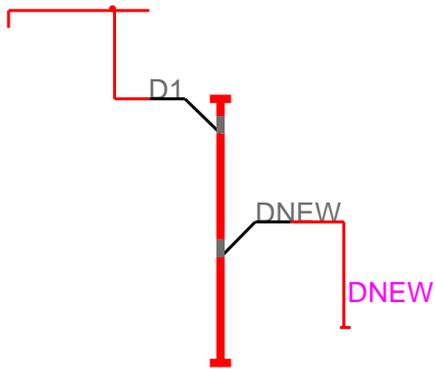
The above example also demonstrates that contrary to a general signal, the 3-phase signal automatically inherits the bundle prefix when connecting by name to a bus connected to a bundle pin. This is only true when a power system bus is initially used to connect to a bundle pin.

In the case of a general signal it is still possible to achieve connectivity by name, by manually showing the signal name connected to the bundle pin. In this case the name is D1 and Lfar is connected to the bundle signal D1 at its plus terminal.

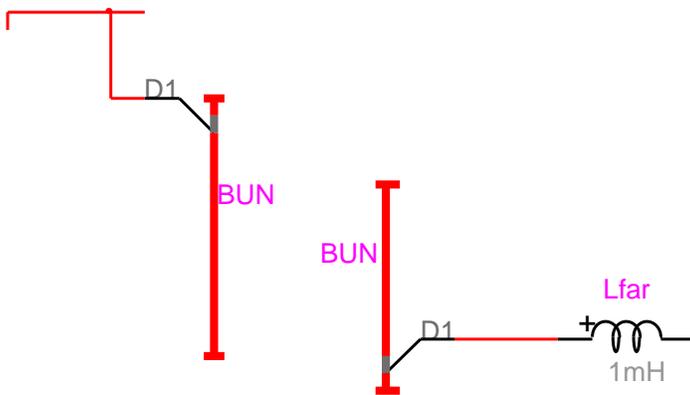


You must remember that the signal name coming out of a bundle pin must be the same as the bundle pin name. If you manually change the signal name, as in the drawing below, then you will create a new signal in the bundle. In this case a new pin name DNEW has appeared after manually changing the name of D1 to DNEW.

## Reference

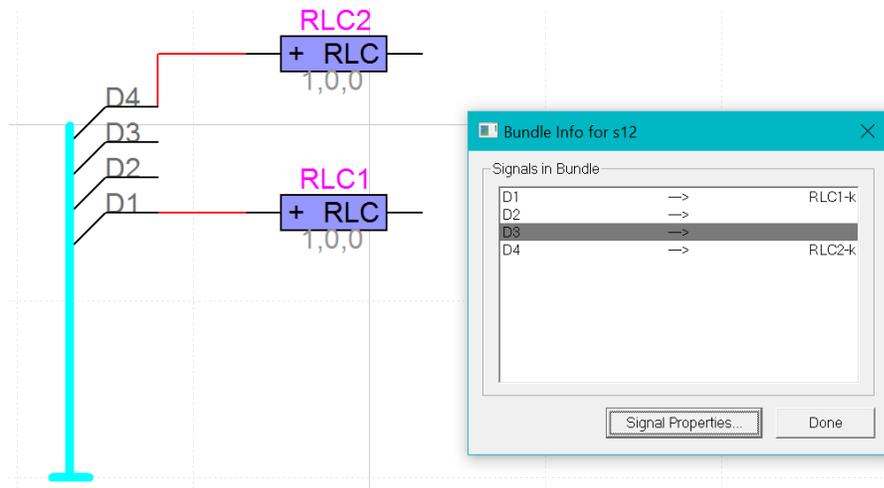


Even though the above options are available in EMTPWorks, connecting bundle signals by name is not recommended and can be avoided in most cases by simply reusing the bundle at the desired location. This is demonstrated below by simply naming the bundles and redrawing for each connection to any one of its pins. Naming a new bundle with the same name as an existing bundle automatically inherits the existing bundle pin list.



### 6.7.14.3 Getting Bundle Information: Bus (Bundle) Info

A list of internal signals can be seen by selecting a bundle and using the Ctrl+I command, or by right-clicking on the bundle and selecting the "Bundle Info" command. This action will display the following box for the shown case:



Here is a summary of the options presented in the bundle information box:

Signals in Bundle	This is a list of the signals contained in the bundle. This list is determined by the breakouts and bundle pins attached to the bundle. You cannot directly change this list. Doubling-clicking on any item in this list is equivalent to clicking once and then clicking the "Signal Info" button.
Signal Properties	This button displays the signal info box for the signal selected in the list.

### 6.7.14.4 Getting Information on Signals Inside a Bundle

Information on signals that are contained in a bundle can be accessed in several ways:

- Select the parent bundle and select the Properties command, or right-click on the bundle and select the "Bundle Info" command, then use the signal list and Signal Info button to display the signal info box.
- Right click on any bundle pin attached to the bundle to select "Bundle Pin Info", then choose the signal in the signal list and click the "Signal Info" button.
- Right click on any bundle pin and select "Bundle Pin Connect".

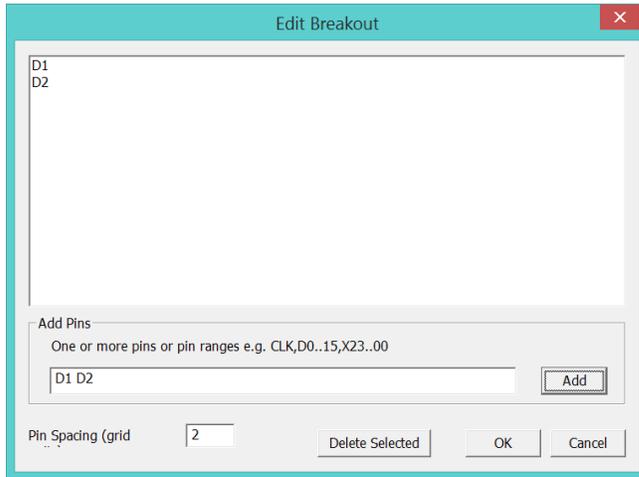
### 6.7.14.5 Using Bundle Breakouts: New Breakout

Signals are attached to a bundle via a special type of pseudo-device symbol called a "breakout". In EMTPWorks, 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.

Any breakout can always be attached to any bundle. When a breakout is attached that contains signals unknown in that bundle, the signals are implicitly added to the bundle.

## Creating a Breakout

To create a breakout, select "New Breakout" command in the "Home>Tools" toolbar, or right-click on the bundle that the breakout is to connect to and select the Breakout command. If the new breakout is to be similar to an existing one, first select the similar breakout or the bundle to which the new breakout is to be connected. This dialog box will appear:



If a bundle or breakout was selected on the circuit diagram then the breakout info box will display a list of the signals in that bundle 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 these 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 these 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 bundle (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 bundle, but we recommend dividing bundles up by function for ease of editing.

- any combination of randomly-named signals can be included in the list, as in these 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. An image of the breakout will now follow your mouse movements and can be placed and connected just like any other type of device.

**Attention:** only alphanumeric (numbers are letters) characters are acceptable for bundle signal names.

### Setting Breakout Pin Spacing

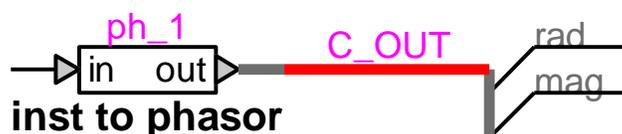
The number in the Pin Spacing box will be the spacing between signal pins on the breakout symbol, in grid units. The default value is 2 to match the standard EMTWorks libraries, but any number from 1 to 100 can be entered.

#### 6.7.14.6 Using Bundle Pins

EMTWorks supports user-created bundle pins on devices. A bundle pin can be defined to have any collection of named internal pins. Note these properties of bundle pins:

- The bundle 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 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 "Bundle Pin Info" command in the pin pop-up menu.
- When a device with a bundle pin is placed, it has a pre-created bundle attached to it by default. This bundle 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 Bundle Pin Info command in the pin pop-up menu. This box allows any internal pin to be connected to any signal in the attached bundle.

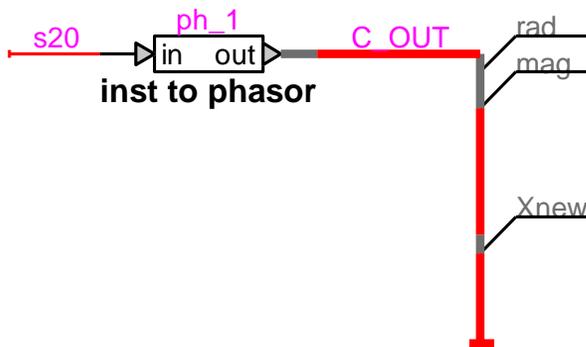
In this example, the device has a bundle pin and contains two signals rad and mag. The bundle name has been optionally turned on and changed to emphasize that the actual signal names are C\_OUTrad and C\_OUTmag.



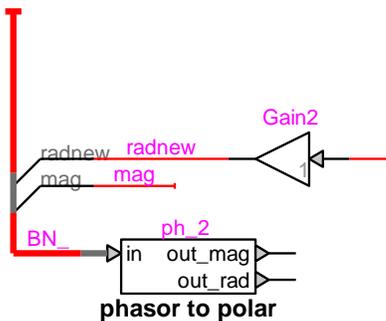
Such devices are providing a fixed set of pin names that can be connected to signals. Any number of such devices can exist in a given circuit, each using its own bundle name.

You can add other signals into the bundle, but only the existing initial signals will be recognized by the device. The extra signals can use the C\_OUT bundle for other channeling needs. You can add signals into an existing bundle by right-clicking on the bundle and selecting the Breakout command.

## Reference



You can also change a signal name connected to a bundle pin, but such an action (not recommended) will miss the connection into the device and simply create a new signal in the bundle. In this example the new signal radnew is simply channeling through the bundle and is not connected to the device pin rad that exists in the bundle pin. Note also that Gain2 is actually connected to BN\_radnew and not radnew.



It is not recommended to use manual signal names when connecting into bundle pins. This also makes things more complex when duplicating and copying. It can be avoided in the majority of cases.

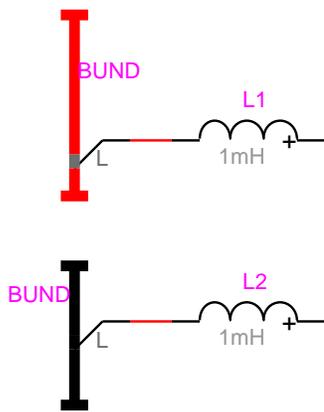
For more information on creating device symbols with bundle pins, see [Placing a Bundle Pin](#).

### 6.7.14.7 Bundles and connectivity by name

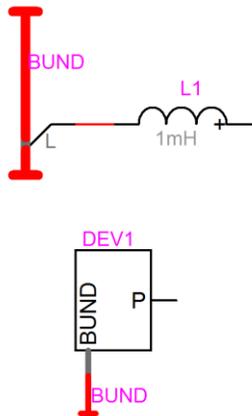
It is strongly recommended not to use connectivity by name (virtual connectivity) when working with bundle pin signals, unless absolutely necessary. In such exceptional cases you must always remember to precede the signal name with the bundle name. Preceding the signal name creates the connectivity in the Netlist, but does not show it on the drawing when you double-click on the signal. There is however an exception to this case when working with 3-phase buses (see [Bundle signal names](#)).

When using bundle pins, you must avoid using signals that do not exist in the bundle pin.

The principle is that if you need to connect two bundled signals located far from each other in a design circuit, then you can use the bundle name to connect by name. In this example, the black bundle is created after the red bundle BUND, but it is given the same name. If you right-click on the black bundle and select Breakout, it will list the pin L to which you can connect with L2. Place the breakout on the black bundle. The left pins of both L1 and L2 are now connected by name. Since you can select only the desired pins in a breakout list, you can always use this technique to create connectivity through graphically disconnected bundles.



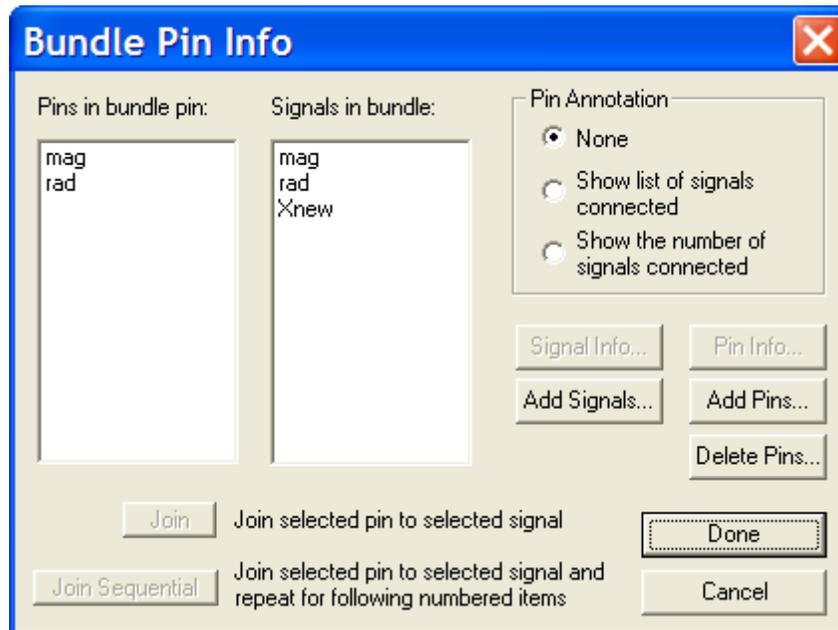
If L2 is made part of a subcircuit, then the signal BUNDL can be sent into the subcircuit by naming the subcircuit bundle BUND:



### 6.7.14.8 Changing Bundle Pin Connections: Bundle Pin Info

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

This menu item will be enabled only when a bundle pin on a device is selected. It allows the association between the bundle internal pins on the device and the signals in the bundle to be changed. This box will be displayed:



This is an advanced feature and should be used carefully.

The left-hand list shows the names of the pins contained in the selected bundle pin. The right-hand list shows all the signals in the attached bundle. 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 bundle pin. The following sections describe the operations available in this box.

#### Changing Signal Connections in a Bundle Pin

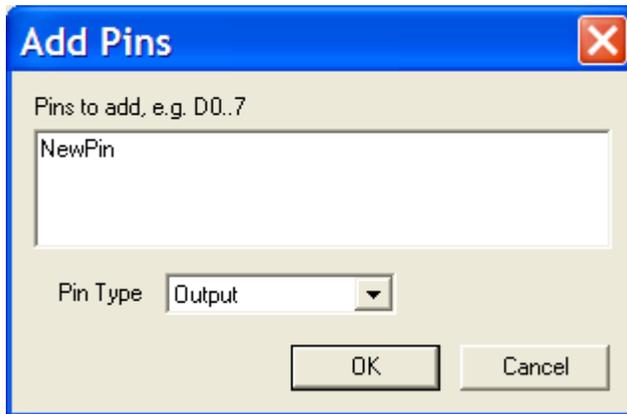
Two buttons are provided to change the association between pins and signals. The Join button causes the selected pin in the pin list to be joined to the selected signal in the signal list. If the selected signal is already attached to another pin in the list, then the signals will be swapped (i.e. a signal can only connect to one pin and vice versa). The signal list will be updated to show the new relationship. If, in the example above, the pin mag is joined to the signal Xnew, then the signals mag will not be connected to the device ph\_1 anymore and will become independent.

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, if pin D0 and signal D4 are selected, then "Join Sequential" will join D0-D4, D1-D5, D2-D6 and D3-D7. 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.

### Adding Pins to the Bundle Pin

The "Add Pins" button (see the "Bundle Pin Info" panel above) allows you to add internal pins to the selected bundle pin. When this option is clicked, the following box will be displayed:



In this case the NewPin is being added as Output (for control signal of output type) pin.

A list of pins can be typed into this box using the same format as the "New Breakout" command described in [Using Bundle Breakouts: New Breakout](#). A pin type can be selected and will apply to all pins created using this operation. This is related to the application.

### Deleting Pins from the Bundle Pin

Clicking the Delete Pins button will remove all internal pins currently selected in the list from the bundle pin. This operation does not become permanent until you close the Bundle Pin Info box.

This operation effectively modifies the definition of the symbol and any subcircuit it may contain, making it different from its original library definition. The pin changes are not permanent until the "Bundle Pin Info" box is closed. It is not allowed to use this option with built-in devices since it will result in data corruption.

#### 6.7.14.9 Bundle Connection menu

The "Bundle Connection" menu appears automatically when

- a signal is connected (drawn into a bundle) without using a breakout pin.
- two different bundles are connected together.

The "Bundle Connection" is also available when the user right-clicks on the menu item "Bundle Pin Connect" of a bundle pin.

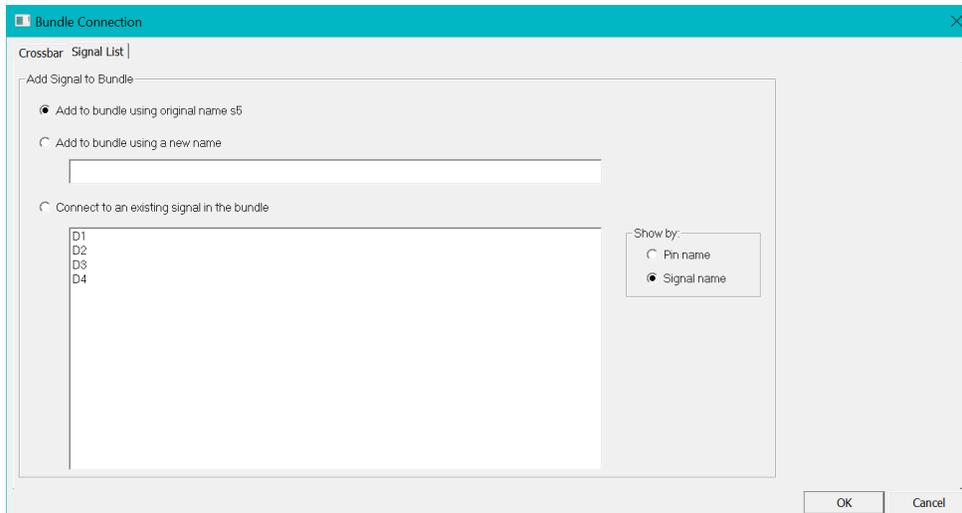
When a signal is connected to a bundle, the "Bundle Connection" menu shown below allows to specify how the signal will be added to the bundle. There are three options (three radio buttons) that allow to:

- connect and create/add the signal name in the bundle
- add to the bundle using a new signal name
- connect to an existing signal in the bundle by simply clicking/selecting on the desired signal name.

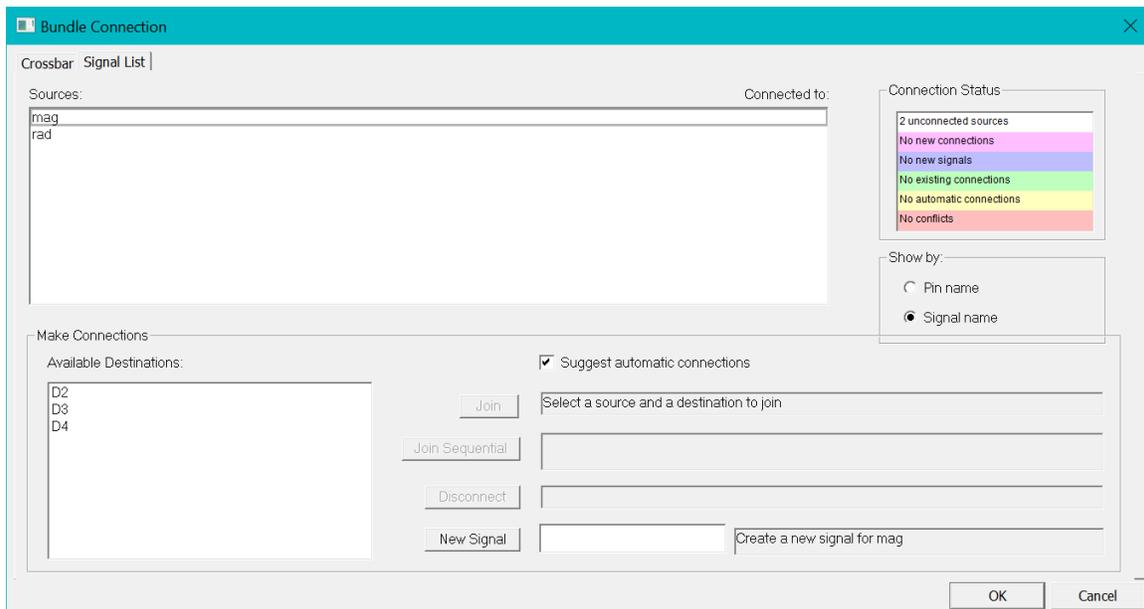
It is possible to show the existing signals with attached pin names using the "Show by" option. The default "Signal name" selection is the most convenient.

The Crossbar tab allows to make connections by clicking on the shown matrix of signals.

## Reference



When a bundle is connected to a bundle, then the “Bundle Connection” menu becomes



The “Available Destinations” box shows the available signal names in the existing (connecting to) bundle. This list will appear only after clicking on a signal in the Sources box. The remaining options are self-explanatory and provide tooltips according to selections. The Crossbar is also available to make manual connections using a matrix of signals.

### 6.7.15 Inter-page Connections

The page creation methods have been presented in [Adding, Deleting and Titling Circuit Pages](#). This section explains how to show signal interconnections for signals appearing on several pages. Due to the EMTWorks ability to connect signals by name (see [Connecting Signals by Name](#)) signals with the same name appearing on different pages are automatically connected. The inter-page connections are used to place a visual marker and provide a mechanism for jumping between pages through connected signals.

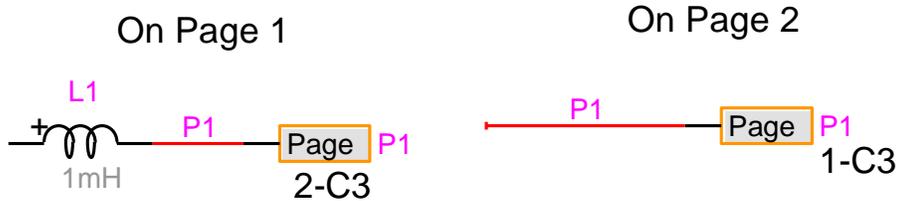
Inter-page connections are made using the "Page Conn" (Page Connector) devices from the "Pseudo Devices.clf" library. When attached to a signal line, the Page Connector makes the

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.

### 6.7.15.1 Automatic Display of Page References

EMTPWorks 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. There are separate page connectors for power signals and control signals.

For example, in this next case both pages have signals named P1 with power Page Connectors "Page Conn Power Signal" attached. Note that both Page Connectors are notated with the page number and grid reference:



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

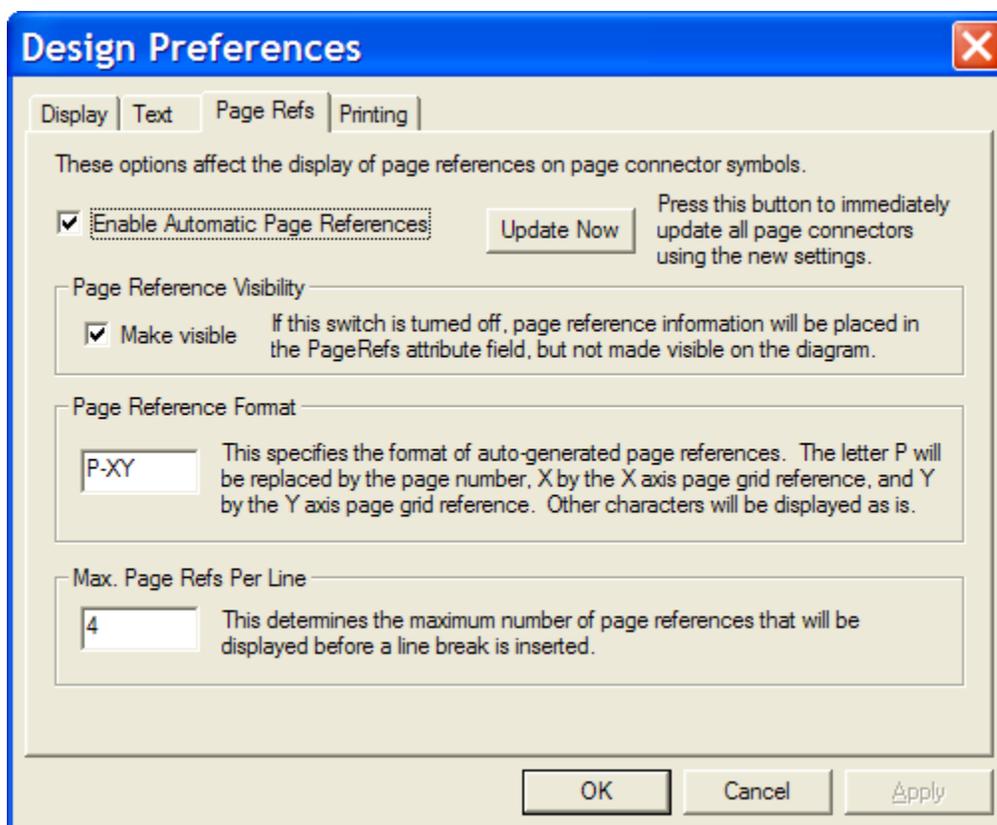
As long as automatic updating is enabled, these references will be updated automatically when any Page Connectors or attached signals are added, deleted, moved or renamed.

The "Page Conn Power Signal" is used for power signals and the "Page Conn output signal" is used for control signals.

You can place one or more Page Connectors on the same page. You can also place two page connectors on the same page for connecting signals with the same name. This provides an extra visual referencing method for virtually connected signals (see [Connecting Signals by Name](#)).

### Enabling and Disabling Automatic Page References

Automatic page reference updating is controlled by the "Design Preferences" command in the "Options>Design" toolbar. Automatic references are enabled in the "Page Refs" tab whenever the "Enable 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 they are disabled, the current page reference settings will be left untouched when any schematic editing is done.



### Manually Updating Page References

If automatic page references are disabled, page references will be updated only when the "Update Now" button in the Design Preferences box is clicked. This can be used for large designs where page reference updating may cause delays while editing.

### Setting Page Reference Format

To change the page reference format, select the "Design Preferences" command in the "Options>Design" toolbar. 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

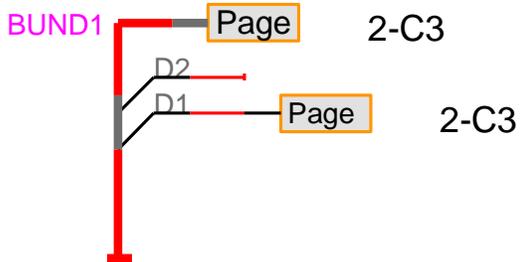
## Reference

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.

### 6.7.15.2 Connecting Bundles Across Pages

Bundles can also be connected (shown to be connected) between pages by the same method, except that the "Page Conn Bundle" Connector must be used. This is illustrated in this simple circuit:



The bundle page connector is used to show the connection between bundles. It is noticed that you can also use page connectors for individual signals in a bundle. The signals in a bundle are connected by name if they have the same name and the same bundle name.

### 6.7.15.3 Changing the Page Connector Symbol

Note that the symbol for the Page Connector device can be changed using the device symbol editor tool just as with any other device. In order that the resulting symbol be recognized by the program as a Page Connector, you must either start with an existing Page Connector symbol, or set the primitive type setting appropriately.

See [Creating a Page Connector](#) for specific information on creating Page Connector symbols.

### 6.7.15.4 Tracing Connections Through Page Connectors

If a "Page Connector" device is selected in the schematic, the right-click Properties command displays this box:



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 (jump 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.

## 6.8 Adding Text and Pictures to the Drawing

### 6.8.1 Using 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.

Text objects 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 the EMTP Netlist. Use the naming and attribute features to attach text to devices and signals.

### 6.8.2 Creating a Text Block

If you click the pencil (Name tool in "Home>Tools") 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 the Enter key. You can also enter text using the Text tool on the "Home>Tools>Draw Graphics" toolbar. This is the same as using the Name tool but exclusively reserved to text only and not for device attributes.

### 6.8.3 Editing a Text Block

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 Windows 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. The Paste menu command will cause the current text entry to be terminated and a new text box to be created. Similarly, the Ctrl key equivalents for Cut (X) and Copy (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.

#### 6.8.4 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 right-click menu Properties command.

#### 6.8.5 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 Pages 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 a "Drawing>Import Sheet Info" operation, this item will be copied to the destination.
- if a "Center in Page" command (design right-click) is used, this object will not be repositioned since it is assumed to be part of the border.

#### 6.8.6 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 Ctrl and Shift keys pressed while clicking on it. The Properties command (Ctrl+I) can then be used to disable the background status. You can also hold the Ctrl and Shift keys and double-click on the background object.

#### 6.8.7 Using Text Variables

The text facility allows special entries called variables to be placed in text boxes. When the text is displayed or printed, the appropriate system quantity 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.

#### 6.8.8 System Variables

System variables start with a "\$" and are enumerated here:

\$DATENOW	This will be replaced by the system date written mm/dd/yy. This is the current date as maintained by the system clock.
\$TIMENOW	This will be replaced by the system time written hh:mm. This is the current time as maintained by the system 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 Pages command.
\$PRINTPAGENUM	This will be replaced by the page number in the printed sequence within 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 \$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, not including its directory path.
\$FILEPATH	This will be replaced by the name of the design file, including its directory path. If the file hasn't been saved, this will be empty.
\$CIRCUITNAME	This will be replaced by the name of the circuit being printed. In the topmost circuit of a hierarchical design (or in any flat design), this is the same as \$FILENAME. In a subcircuit, this will be the hierarchical name of the circuit.

The format used to display date and time values is determined by the Regional Settings control panel in Windows.

### 6.8.9 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 design attribute field Revision was defined for the design with the following contents:

2.1A Mar 18, 2000

then the variable &Revision would appear as "2.1A Mar 18, 2000" on the diagram.

Data related to some EMTP simulation options is saved into the design attribute ParamsEMTP.

### 6.8.10 Editing Text with Variables

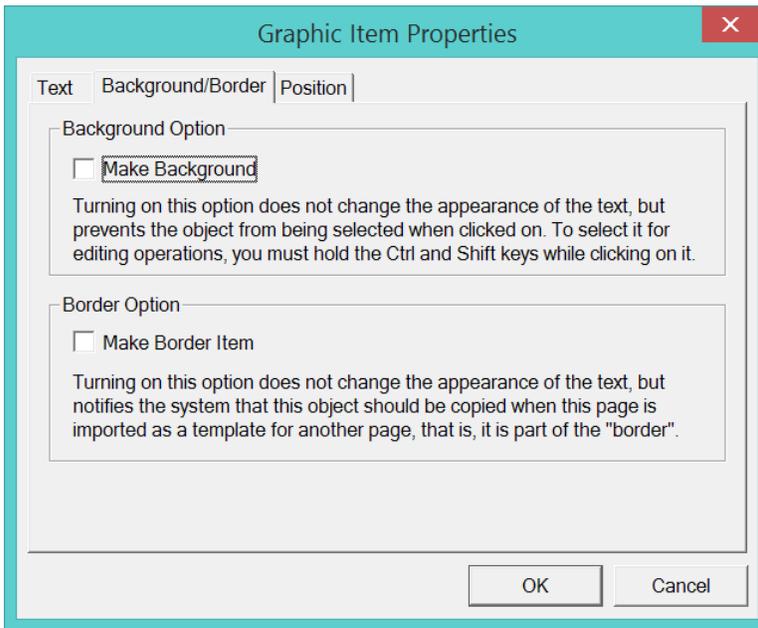
Text items on the schematic will normally be displayed with variables replaced by their values. When a text item is selected or clicked on with the Name cursor it is redisplayed in its raw format with the variable names shown without interpretation. This allows the items to be edited with the text in its actual stored position. In some cases you will need to push the End keyboard key or use "View>Zoom>Redraw" to correctly refresh the information.

### 6.8.11 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 (refreshed). The text box can be expanded as necessary by adding blanks at the end of any one line in the item.

### 6.8.12 Text Object Options

To set text display options and text style, select the text block by clicking on it with the Point cursor, then right-click to select "Properties" and then "Background/Border" tab. This will display the following box:



This table summarizes the options available.

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 Ctrl and Shift keys pressed while clicking on it.
Make Border Item	Turning this switch on marks this item as a border object. This means that it will be treated as part of the sheet border. That is, 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. See Creating Custom Sheet Border Graphics for more information.

### **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 Ctrl and Shift keys pressed while clicking on it.

### **6.8.13 Creating a Picture**

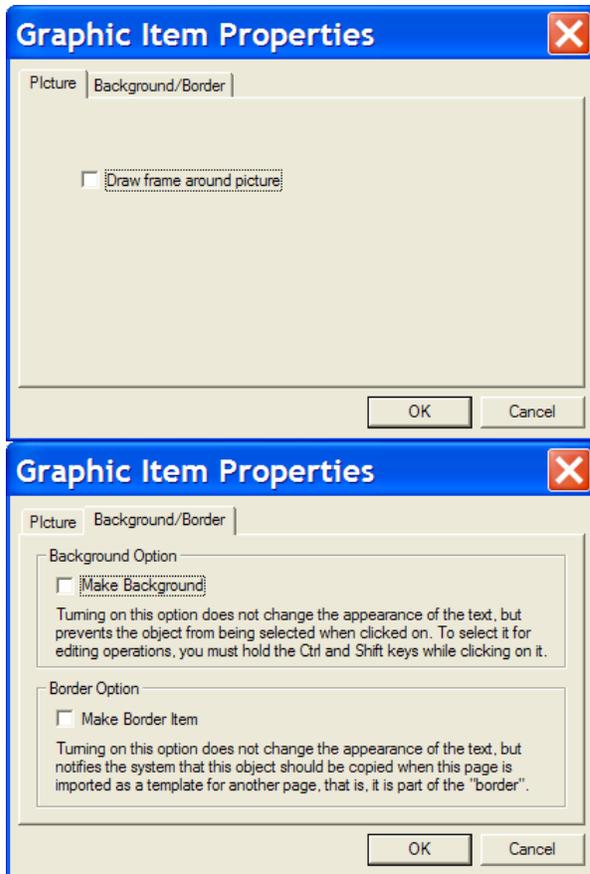
Pictures cannot be created from scratch right on the schematic diagram. They can be created either in another application or using the device symbol editor tool. To do this:

- Enter the device symbol editor, or other paint, drawing or drafting program to create the graphics object.
- Select the desired object.
- Select the Copy command in the Edit menu. This places a copy of the graphics item on the clipboard.
- Select your EMTPWorks 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.

#### **Setting Picture Characteristics (Picture Object Info)**

If a picture object is selected, the Properties command displays the following box:

## Reference



The following table summarizes the options available for pictures.

Draw frame around picture	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 Ctrl and Shift keys pressed while clicking on it.
Make Border Item	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.

## 6.9 Simulation options

Before simulating a design in EMTPWorks, the user must select simulation options. These options are available “Simulate>Advanced>Simulation Options” menu shown below.

Complete documentation is available in the document [Simulation Options](#).

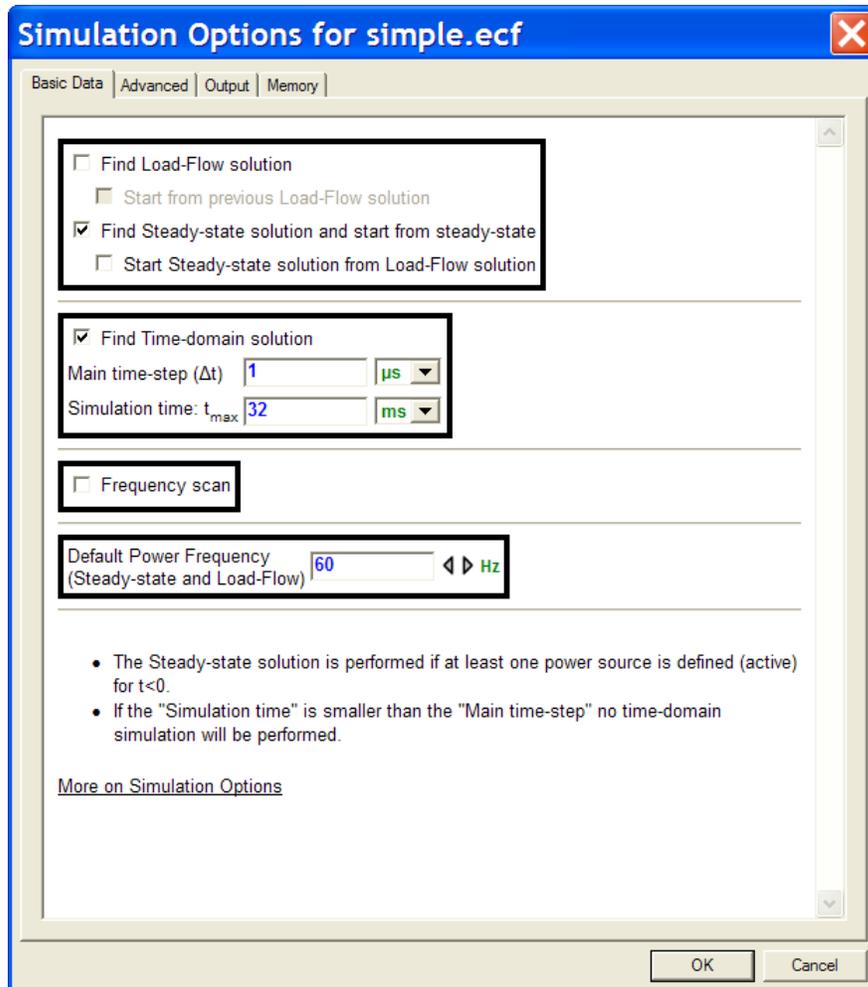
EMTP can simulate electrical networks (power networks or power circuits) and control systems. The control systems are simulated at the bloc-diagram level. Control blocks can be also used to model power devices and connect them to power networks.

EMTP has 4 distinct simulation options (see also tooltip for each option):

- Load-Flow solution
  - The electrical network equations are solved using complex phasors. The active (source) devices are only the Load-Flow devices (LF-devices). A load device is used to enter PQ load constraint equations.
  - Only single (fundamental) frequency solution is achievable in this version. The solution frequency is specified by ‘Default Power Frequency’ and used in passive network lumped model calculations.
  - Future EMTP versions will include a harmonic load-flow.
  - The same network used for transient simulations can be used in load-flow analysis. The EMTP Load-Flow solution can work with multiphase and unbalanced networks.
  - The control system devices are disconnected and not solved.
  - This simulation option stops and creates a solution file (Load-Flow solution data file). The solution file can be loaded for automatically initializing anyone of the following solution methods.
  - The ‘Start from previous Load-Flow solution’ is used to restart the Load-Flow from a previous solution. This option is not useful when the solution is found without convergence problems and the number of iterations is low. It may help to reduce the number of iterations when trying to improve convergence tolerance or provide better initial conditions for complex cases.
  - To learn more about this option, click on its help link ([Load-Flow options](#)).
- Steady-state solution
  - The electrical network equations are solved using complex phasors. All devices are given a lumped circuit model. This option can be used in the stand-alone mode or for initializing the time-domain solution.
  - A harmonic steady-state solution can be achieved.
  - The control system devices are disconnected and not solved.
  - Some nonlinear devices are linearized or disconnected. All devices have a specific steady-state model.
  - The steady-state solution is performed if at least one power source device has a start time (activation time) lower than 0.
  - The “Start Steady-state solution from Load-Flow solution” is used to initialize the steady-state solution from proper Load-Flow solution. This is a useful option for performing complete simulations from Load-Flow to time-domain.
- Time-domain solution
  - The electrical network and control system equations are solved using a numerical integration technique.
  - All nonlinear devices are solved simultaneously with network equations. A Newton method is used when nonlinear devices exist.
  - The solution can optionally start from the steady-state solution for initializing the network variables and achieving quick steady-state conditions in time-domain waveforms.
  - The steady-state conditions provide the solution for the time-point  $t=0$ . The user can also optionally manually initialize state-variables.

## Reference

- Frequency scan solution
  - This option is separate from the two previous options. All source frequencies are varied using the given frequency range and the network steady-state solution is found at each frequency.



## 7 Attributes

### 7.1 Attributes Overview

Attributes are arbitrary blocks of text that can be associated with any device, signal or pin in a design, or with the design itself. Attributes have a wide variety of uses, including:

- Displaying device name, part type, value, etc. on the schematic.
- Storing data for the device model. Attributes are used to store and send data into EMTP.
- Storing various EMTPWorks related data on the object (device, signal, pin or design).
- Storing any other user defined data for various tasks of design maintenance, annotations or some other encapsulation tasks.

Attributes can be used in conjunction with the scripting and report generation features of EMTPWorks to implement powerful interfacing, customizing and error checking features. You can define as many attribute fields as you need to store the text data required for your application.

In addition, many EMTPWorks 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 based changes and programming are only for advanced usage. Illegal manipulations will not be necessarily intercepted by EMTPWorks and you may corrupt device or design data and even make a design impossible to repair. A clear understanding of attributes is needed before making any changes.

### 7.2 Attribute Organization

#### 7.2.1 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 that field can be used in (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.

#### 7.2.2 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 EMTPWorks.

- ❖ See [Predefined Attribute Fields](#) for a complete listing of built-in fields.

### 7.2.3 User-defined Fields

You can add more fields to a design at any time using the "Define Attribute Fields" command in the "Options>Design" toolbar.

### 7.2.4 Primary vs. Secondary Fields

EMTPWorks has 50 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 primary/secondary setting is for viewing and editing convenience only and has no effect on the meaning or internal usage of the field. This setting for individual fields can be changed at any time using the "Define Attribute Fields" command in the "Options>Design" toolbar.

### 7.2.5 Attribute Limitations

Attribute fields have the following specific limitations:

- Maximum length of field name: 16 characters
- Maximum length of field data item: 2000,000,000 characters. Note: In principle, binary data could be pasted into an attribute field although this is not recommended since EMTPWorks 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.

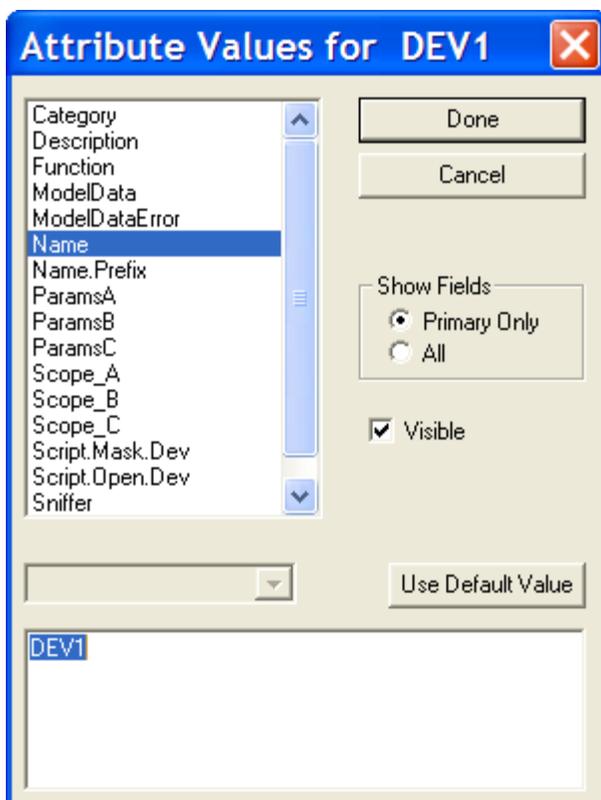
### 7.3 Entering and Editing Attribute Data - Basic Procedure

To edit the attributes associated with a device, pin or signal, you can either:

- Choose the object of interest on the schematic and right-click to select the command Attributes.
- The Attributes command is also available when you select the object and use Ctrl+I.

To edit design attributes, select the "Design Attributes" in "Options>Design" toolbar.

Any of these methods will display the general attribute editing box which is used to enter or edit attribute data on a single object:



By default the field list at the left side of this box shows only the primary fields available for the selected kind of object. You can click on the All fields button to view all available fields.

#### 7.3.1 Entering Data Values

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. When you select another field, data entered for the previous field is saved. 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.

### 7.3.2 Restoring the Default Value

Clicking the Use Default Value button sets the value for the selected field to the default value stored with the symbol. This is valid only for device and pin attributes for which a default value was specified. 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.

- ❖ See Setting Part Attributes for more information on entering default attribute values while editing a symbol.

### 7.3.3 Using the Value List Pop-up Menu

If the selected field has an associated value list field, then a pop-up value menu will appear near 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.

When an attribute field has more than one possible value, the list is displayed in one of these menus. For example, if the sub-menu "Part" appears, this means there are multiple part definitions available in this device type. Selecting one of the parts in the part sub-menu causes the part attribute field to be changed for that device.

See more information on value list fields in [Using Value List Fields](#).

### 7.3.4 Displaying an Attribute on the Schematic

To display device, signal or pin attribute text on the schematic:

- Edit the attribute value as desired.
- Turn on the Visible switch.

When you click Done, the attribute text will now be displayed in a default position near the device, pin or signal. It can be dragged to the desired location using the pointer cursor.

- Design attributes can be displayed on the schematic using text variables. See [Using Text Variables](#)
- For information on setting the default position for an attribute on a symbol, see [Using Default Position Fields](#).

## 7.4 Entering Design Attributes

To view or set attributes associated with the design itself, select the "Design Attributes" command in "Options>Design". 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 the Netlist output.

The Visible option is not available for design attributes, since there is no single object on the schematic to associate them with. Design attributes can be made visible by using text variables, described in [Using Text Variables](#).

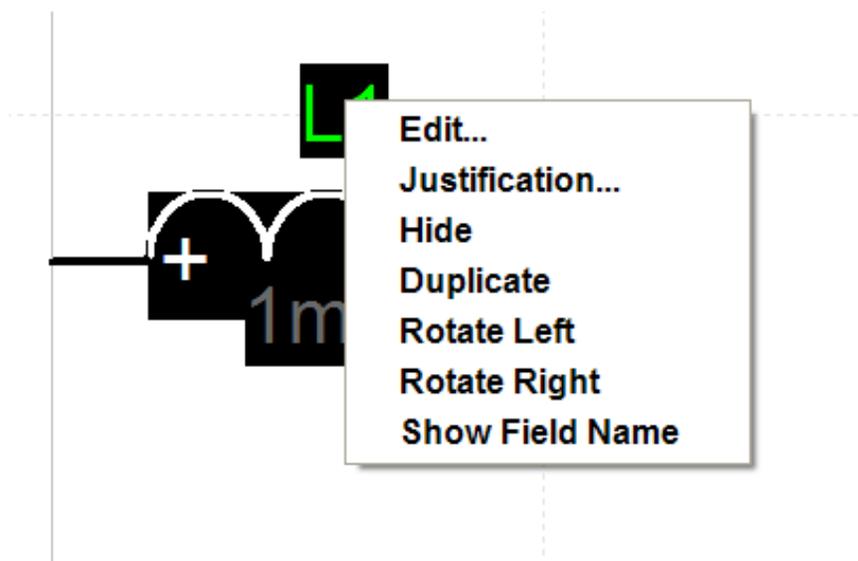
## 7.5 Entering Pin Attributes

You can view or set pin attributes by right-clicking on the desired pin and selecting the Attributes command.

The attribute editing box for pins is identical to the one for devices, signals and the design, except that Next and Previous buttons appear at the bottom. These can be used to view and edit other pins on the same device without having to return to the schematic and select them individually.

## 7.6 Attribute text right-click menu

When you right-click on a visible attribute text you have access to several editing commands. In this example the visible Name attribute is clicked:



The Edit command is equivalent to double-clicking on the attribute text.

Justification is explained in [Setting Visible Attribute Justification](#).

Hide is explained in [Hide a Visible Attribute Value](#).

Duplicate is explained in [Duplicate, Merge and Delete options](#).

The rotate commands allow rotating the text as explained in [Rotating Attribute Text](#).

The "Show Field Name" command is described in [Showing the Field Name with an Attribute Value](#).

## 7.7 Rotating Attribute Text

To rotate an attribute text item that is already displayed on the schematic, right-click on the attribute text and select the rotation related commands starting with Rotate.

## 7.8 Hide a Visible Attribute Value

If you wish to hide a device, pin or signal attribute value that is displayed on the schematic (that is, remove the displayed text from the schematic, but keep the value associated with the object), you can use any of these methods:

- Right-click on the text and select the Hide command in the pop-up menu.
- Right-click on the text and select the Edit command. In the text edit box that appears, turn off the Visible switch.
- Display the general attribute editing box for the object, using the methods described earlier in this chapter. Select the field in question and turn off the Visible switch.
- Click on the "Zap (delete)" tool in the tool palette, or select the Zap command in the Edit menu and then click on the attribute text you wish to hide.

Any of these methods removes the displayed text from the schematic without modifying the value that is stored with the object.

## Reference

If the value you are hiding is a signal name, this operation may have side effects on the connectivity represented by the signal. Hidden signal names are not controlled by the user and this operation will break the virtual connection method (see [Connecting Signals by Name](#)).

### 7.9 Setting Attribute Text Style

There are two ways to set attribute text style globally for all attributes and separately for individual attributes. When an attribute has a separate setting the global command will not affect it.

There is no way to set text style for an individual object attribute item.

Changing the attribute text style affects all visible attributes throughout the design for all attributes that have the default setting. This may alter text alignment and position to accommodate a different text size.

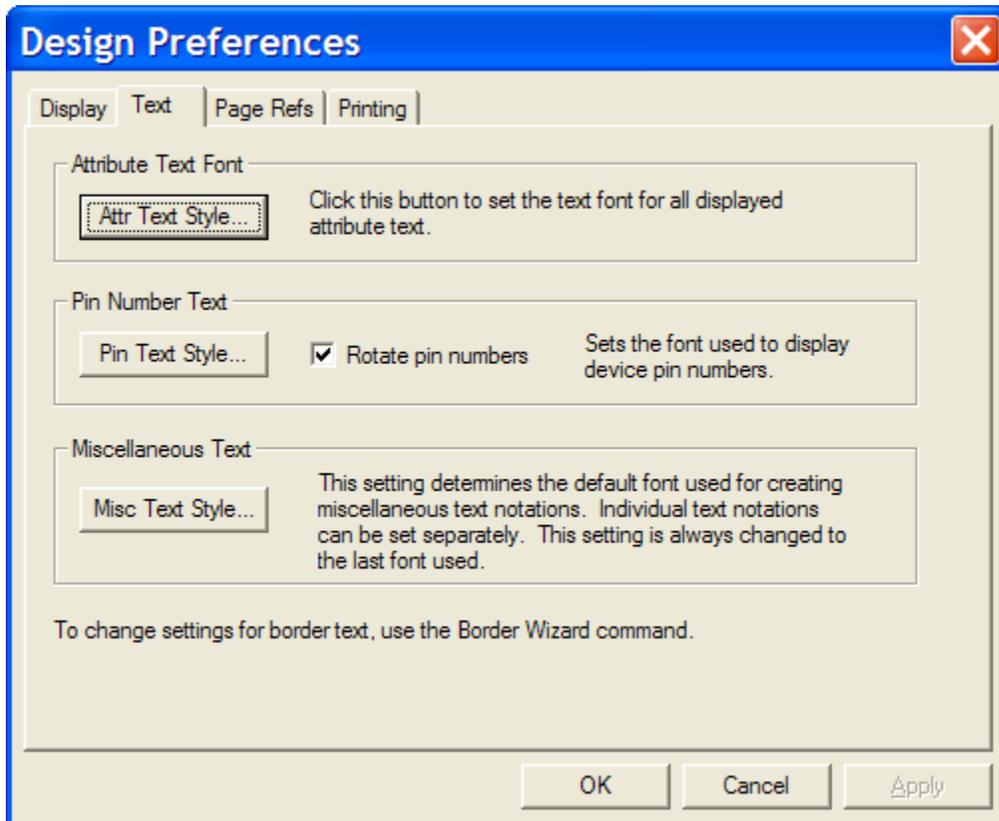
The attribute text style setting affects the following types of items:

- Device and signal names.
- Bundle breakout and bundle pin labels.
- All other attributes displayed on the diagram.

but does not affect:

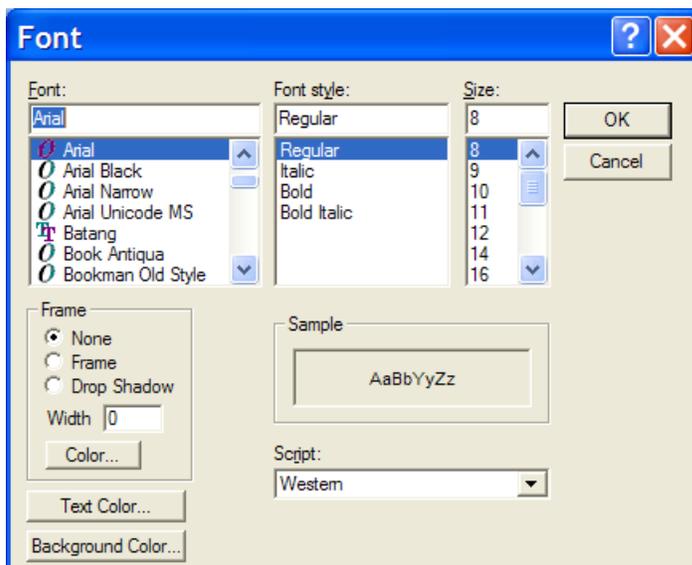
- Border text
- Pin numbers

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.



To set the global text style:

- ❑ Choose the "Design Preferences" command in the "Options>Design" toolbar.
- ❑ Select the Text tab and click on the "Attr Text Style" button. This will display the following box:



This box allows you to select the text font, size and style used for all attributes displayed in the design.

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 their position and framing is affected by the text change.

Another method for setting text style for individual attributes is available through the definition of attribute fields (see [Defining Attribute Fields](#)).

## 7.10 Setting Visible Attribute Justification

Attribute justification determines how text is aligned relative to the device it is associated with. Generally, the program chooses a justification that is appropriate for the initial placement of the text and there will be no need to change it. However, if text is being modified frequently and does not reposition itself as desired, the justification can be modified individually for text items.

To set the justification, right-click on the text item in question. In the attribute pop-up menu that appears, select the Justification command. 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.

The Justification command affects only the position of the text relative to the associated object. It does not affect the alignment of text in multi-line text items on the schematic. EMTPWorks does not provide any method of choosing center or right justification of multi-line text items.

### **7.11 Displaying an Attribute Value in Multiple Locations**

You can display a single attribute value in multiple locations on a schematic. This is intended for use particularly with signals, which can be spread over large areas of a page or multiple pages. If a value is displayed more than once, editing any one of the displayed values will affect all appearances of it.

To create multiple displayed attribute items, right-click on the text item in question. In the attribute pop-up menu that appears, select the Duplicate command. 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.

### **7.12 Showing the Field Name with an Attribute Value**

The "Show Field Name" command from the right-click on attribute text menu 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.

### **7.13 Using Value List Fields**

A value list field allows you to specify a list of possible values that can be used in a selected field. This list of values is then displayed to the user at various locations, allowing easy selection of one of the available items.

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 takes these steps:

- When the attribute edit box is displayed for a given object and the field "xxx" is selected, a pop-up menu is displayed containing all the possible values for the field. The user may then select the appropriate value from the menu.
- When a pop-up menu is displayed by right-clicking on an object, 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".

#### **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 device symbol editor 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.

## 7.14 Using Script fields

EMTPWorks allows you to call a script (attribute change script) when a given attribute value changes. A script field has the same name as the associated field but with ".Script" appended. You must be careful not to create loops through scripts, since that may kill the application. When an attribute changes, EMTPWorks first searches for the attribute change script in the object where the change occurred and if the corresponding "xxx.Script" string is empty, it looks into the design.

## 7.15 Using Default Position Fields

For device or pin attribute fields that are to be displayed on the schematic, an associated field called a "default position field" can be created. For example, this mechanism is used to provide default positions for the "Name" and "Part" fields used in all standard EMTPWorks 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 and 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 measured in 1/1000". Negative values are allowed (negative is up and left).
- The X coordinate can be optionally followed by an upper case letter indicating horizontal justification, as follows: "L" for left, "M" for middle, "R" for right. 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 X coordinate can be optionally followed by a lower case letter indicating which point on the symbol's bounding box to use as the reference point, as follows: "l" for the left edge, "m" for the midpoint, "r" for the right edge. The given number specifies the offset from the left, middle or right position of the symbol. If no letter appears, left is assumed.
- The Y coordinate can be optionally followed by an upper case letter indicating vertical justification, as follows: "T" for top, "M" for middle, "B" for bottom. 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.
- The Y coordinate can be optionally followed by a lower case letter indicating where on the symbol the position is to be measured from, as follows: "t" for top, "m" for middle, "b" for bottom. The given number specifies the offset from the top, middle or bottom of the symbol's bounding box. If no letter appears, top is assumed.

Some examples of value points are:

OR,OT	The value will extend right and down from the top-left corner of the device.
OMm,OBt	The value will be centered above the symbol with the bottom of the text at the top of the

## Reference

symbol.

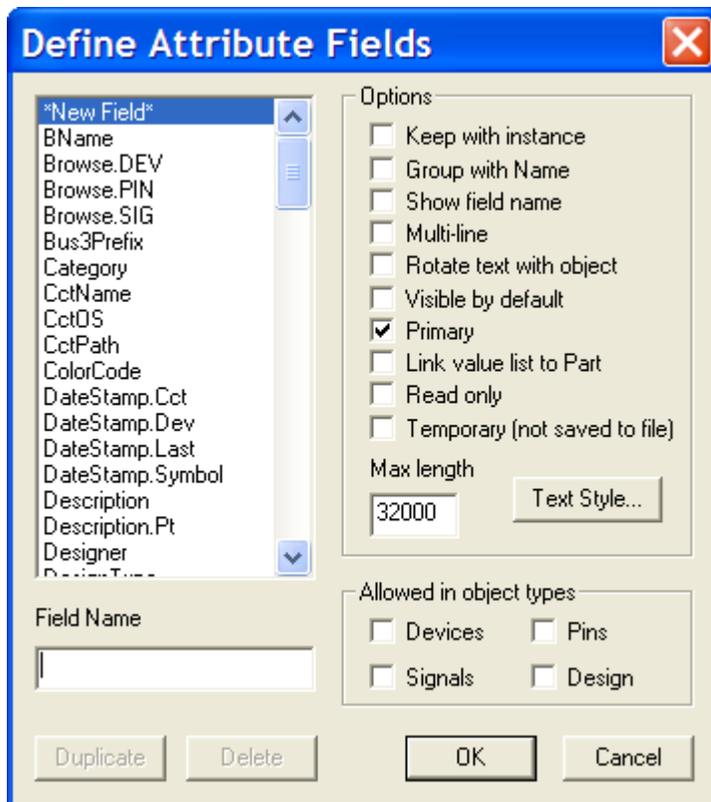
- 150,-75            The value will be centered at a position above and to the right of the top-left corner.
- (400L,100)        The value will extend left from a point to the right of the top-left corner of the device. (Leading and trailing characters are ignored.)

Normally, a default value for the "xxx.Pt" field should be defined with the symbol in the library using the device symbol editor. 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.

You can place a symbol on a schematic, position all the displayed attributes in the locations you want them, and then use the "Options>Part Type>Save to Lib" command to store the symbol back into a library. The "Save to Lib" command automatically creates .Pt fields for any displayed attributes and saves them with the new definition. See [Saving Part Type to Library](#) for more information.

## 7.16 Defining Attribute Fields

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



To create a new field:

- ❑ Select the "\*New Field\*" item in the list, if it is not already.
- ❑ Type the name of the field in the "Field Name" box.

- ❑ Select one or more of the boxes under "Allowed in object types".
- ❑ Optionally enter a maximum length.
- ❑ Optionally enable any of the other field options.
- ❑ Click OK. The new field will now appear in the attribute data entry box for the selected object types.

You should not modify already defined fields for the EMTP template as this may have dangerous consequences and corrupt design data.

### 7.16.1 Setting 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.

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.

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.

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 [Temporarily Displaying 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.

#### Field Name

The Field Name box allows you to enter the name of a new field or rename an existing user-defined one. If you rename a field so that it has the same name as another existing field, this becomes a Merge operation described in Merging Two Existing Attribute Fields. You will be prompted to confirm a merge before proceeding.

#### Keep with instance

This option is not used in the current version of EMTPWorks.

#### 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.

This option can be used for fields that are typically shown with the name on a schematic.

#### Show Field Name

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

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

#### Allow Carriage Returns: "Multi-line"

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.

Changing this setting has no effect on values already in the circuit.

### **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.

### **Link value list to Part**

This is for cases where you have one symbol that might have multiple part names. In the Part.List field you can have a comma-separated list of values that can be put in for Part. Any other field X can also have a field X.List. If "link value to Part" is turned on, then when the user selects a Part value from the list, the corresponding value for X is automatically set. Learn more on the List field in [Using Value List Fields](#).

### **Read only**

This option makes the attribute read only. It cannot be modified by the user by selecting the attribute in the design. This is typically used for setting data from scripts or saving data that should not be touched by users.

### **Temporary**

Creates a temporary attribute.

### **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, although in general we suggest using different names for fields used with different object types.

At least one of the "Allowed in Object Types:" switches must be selected for a new field.

### **Maximum Length**

You can specify a maximum length (in characters) for the field data. This value does not determine how much storage is allocated internally for the data, it only affects checks that are done when the user enters data. All attribute data stored with the design occupies only the space required for its current value.

### **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. More information on this operation is given in the following section.

### **Delete**

This button allows you to permanently remove a field definition and all associated data from the design.

THIS CANNOT BE UNDONE!!!

More information on the delete operation is given in the following section.

## 7.17 Duplicate, Merge and Delete options

The Duplicate, Merge and Delete Attribute functions do more than just add or delete a name in the attribute list. They will actually scan the entire design and update the data stored with each object. This makes them very powerful tools for updating your design. First we will review how the functions are invoked, then mention some possible uses for them.

These three functions are all invoked from the "Define Attribute Fields" command. No design data is modified until the OK button is clicked in the Define Attribute Fields box.

When the design is updated, all Duplicates are done first, then all Merges and Deletes. This means that a field can be duplicated, then the original source deleted, all in one invocation of the "Define Attribute Fields" box.

There is no guarantee of the order of execution within the Duplicates or within the Deletes and Merges. Therefore, chains of operations (e.g. making a copy of a copy) are not guaranteed to work as expected. If there is any doubt, perform one operation, then exit the "Define Attributes Fields" box using the OK button, then perform the next operation.

Each of these operations can result in data being updated throughout a design, and they cannot be undone! For large designs there may be a considerable delay after the OK button is clicked in the "Define Attribute Fields" box.

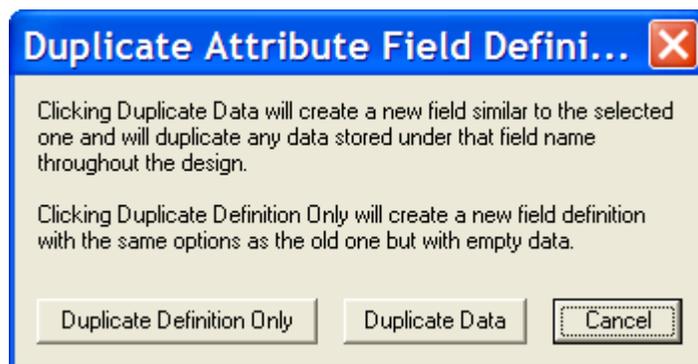
## 7.18 Globally Duplicating Attribute Data

The Duplicate operation adds a new field name to the list with characteristics the same as an existing one. It will also optionally duplicate the data stored under that name throughout the design.

The pre-defined fields in EMTWorks cannot be renamed or removed. However, you can duplicate them, and the copy will be marked as "user-defined", meaning that it can be deleted again or modified in any way.

A field is duplicated by selecting its name in the field list and then clicking on the Duplicate button in the "Define Attribute Fields" dialog:

At this point a box will be displayed allowing you to choose one of the following options:



Duplicate Definition Only

A new field will be defined with characteristics just like the one being duplicated, but it will have an

## Reference

empty value in each of the objects in the design.

### Duplicate Data

A new field will be defined and the associated value in each object in the design will also be duplicated and associated with the new field name. Changing the value of the new field in a given object will have no effect on the original field.

One important feature of this Duplicate operation is that any attribute display options, such as "Visible by Default" or "Show with Field Name", that are turned on before leaving the "Define Attribute Fields" box will be applied to the duplicated data. This means that a field that was formerly hidden can be duplicated and displayed throughout the entire design with just this one operation.

## 7.19 Merging Two Existing Attribute Fields

The Merge operation takes two existing attribute field definitions and merges them into one. If a given object (device, signal, pin or design) has values in both of the original fields, the one being renamed takes precedence and the other is lost.

Pre-defined fields cannot be renamed, so they cannot be merged into other fields. You can however merge a user-defined field into a pre-defined one.

There is no explicit button for the Merge operation. Merge is invoked whenever a field is renamed so that its name matches another field's. The procedure is as follows:

- Click on the field to be renamed in the field list. The data in this field will take precedence when the design is updated.
- Type the new name (i.e. the name of the field you wish to merge to) into the Field Name text edit box.
- Type the Tab key or click on any of the option switches to indicate that you have finished typing.
- A box will be displayed asking you to confirm the merge. Click the OK button:

The actual merging of design data does not take place until you click on the OK button in the "Define Attribute Fields" box.

You can only merge fields that have similar "Allowed in Object Types" settings. It means that you cannot merge a signal field into a device field.

## 7.20 Deleting Attribute Field Definitions

The Delete operation removes the selected name from the list of available fields and removes all data values associated with the field throughout the design. Pre-defined fields cannot be deleted. Before deleting any field, verify its documentation or its usage, harmful conditions will result otherwise.

This operation cannot be undone and should be used with care.

To delete a field, simply select its name in the field list and click on the Delete button.

In any case, no design data is updated until the OK button is pressed on the "Define Attribute Fields" box.

## 7.21 Temporarily Displaying Attributes

The Duplicate function is a convenient method of displaying data temporarily on the schematic. For example, you may wish to display some simulation parameters on the schematic while tracking a particular design problem, even though you don't want them there for the final printout.

To do this:

- Select the Define Attribute Fields command ("Options>Design").
- Choose the field you wish to display.
- Click on the Duplicate button.
- Click on the Duplicate Data option in the confirmation box.
- Enable the "Visible by Default" and (if desired) the "Show Field Name" and "Primary" options.
- Click the OK button in the Define Attributes box to complete the operation.

The duplicate field will now be displayed on each object in the design. When the duplicate field is no longer needed, it can be removed with the Delete function described above.

## 7.22 Permanently Showing Data Throughout a Design

The "Visible by default" option in the "Define Attribute Fields" box normally has no effect on values that already exist in the design. Enabling the option for a field that is already in use affects only future entries. However, the Duplicate and Merge operations can be used to update the display of existing data throughout the design. This is done as follows:

- Select the "Define Attribute Fields" ("Options>Design").
- Select the field to be displayed in the field list.
- Click on the Duplicate button.
- In the confirmation box, click on the "Duplicate Data" button.
- You will now have a copy of the original field: Make sure the newly-created field is selected in the list, then rename it back to the name of the original field. If the original field was "Name", the duplicate will be "Name1" and we now select "Name1" and rename it back to "Name". Press the Tab key or select another field name when you are finished typing.
- The Merge confirmation box will appear. Click the OK button.
- Click on the "Visible By Default" option and any other display options desired.
- Click the OK button to complete the operation

This operation in effect re-enters all the data into the selected field with the new display options.

## 7.23 Merging Dissimilar Designs

The Merge feature is a useful tool when bringing together designs created with different attribute definitions, or when importing designs from other systems or older versions of EMTPWorks.

We strongly recommend reviewing attribute field usage under these circumstances and using the Merge feature to convert all non-standard fields to pre-defined ones where possible. This will make the design more compatible with the standard report formats provided with EMTPWorks and will reduce confusion in future design revisions.

## 7.24 Importing Attribute Definitions

Whenever circuit data is brought into a design, either from a design file, a device library or the clipboard, it is possible that the incoming data contains attribute fields that are not defined in the destination design. This section discusses how EMTPWorks deals with this situation.

## Reference

In general, EMTPWorks attempts to ensure that an entry exists in the attribute definition table for every field used in the design. When new data is imported, field usage in the imported data is compared to the existing table. Normally, no discrepancies are found and this process is invisible. If any mismatch is found, you will be warned and given a chance to add the new definitions to the table automatically.

### 7.25 Pasting from the Clipboard or Placing a Library Device

In these two cases, EMTPWorks does the following:

- All fields used in the incoming device or circuit scrap are looked up in the design's attribute table.
- Any field that doesn't exist in the design is put on a list to be added.
- Any field that does exist in the design but is incompatible (i.e. different "object type" or "keep with instance" settings) is renamed by adding an underscore "\_" to the name.
- A list of the new fields that will be added is displayed in a warning box. If any of these fields are unexpected, or can be converted to standard fields, the names should be noted at this point. Changes in field usage can be made using the Merge and Delete operations mentioned elsewhere in this chapter.

### 7.26 Predefined Attribute Fields

The following tables list fields that are predefined for each new design that is created from the standard EMTP template (EMTP Default Design.ecf). These fields cannot be removed because many of them are used internally by specific EMTPWorks or EMTP features.

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.
- Description: How the field is used.

More on the usage of these fields can be found in other documentation sections. The documentation of individual devices also contains information on device data storage into attributes.

Field Name	Used In	Description
ABELSrcName	Devices	Only in the symbol editor. Internal usage.
AltPicName.Dev	Devices	Used for the addAltPic method for DWType objects (see <a href="#">EMTPWorks JavaScript extensions</a> )
AutoSym.Bottom	Devices	The string used in the "bottom pins" box in the "Auto Create Symbol" command in the device symbol editor. See <a href="#">Auto Create Symbol</a> .
AutoSym.Left	Devices	The string used in the "left pins" box in the "Auto Create Symbol" command in the device symbol editor. See <a href="#">Auto Create Symbol</a> .
AutoSym.Right	Devices	The string used in the "right pins" box in the "Auto Create Symbol" command in the device symbol editor. See <a href="#">Auto</a>

		<a href="#">Create Symbol</a> .
AutoSym.Top	Devices	The string used in the "top pins" box in the "Auto Create Symbol" command in the device symbol editor. See <a href="#">Auto Create Symbol</a> .
Bname	Signals	Not used, kept for future needs.
Browse.DEV	Design	Internal usage for the configuration of the Browser tool.
Brows.PIN	Design	Internal usage for the configuration of the Browser tool.
Brows.SIG	Design	Internal usage for the configuration of the Browser tool.
Bus3Prefix	Design	Internal usage for setting the 3-phase bus default prefix.
BusJoinLabel	Signals	Internal usage.
Category	Devices	Allows categorizing devices. Available to the user. Can be used as an alternate device prefix by entering the name of this field in PrefixField. Not used internally.
CalculateModelData	Devices	Used by some devices for providing a function for calculating model data.
CctName	Design	Design file name. Sets the window title and name of next saved file. Normally only set by the Open and Save commands.
CctOS	Design	The system (Windows) that was used to create the design file. Normally only set by the Open and Save commands.
CctPath	Design	The directory path name for the design file. Normally only set by the Open and Save commands.
ColorCode	Devices	Used for color coding devices based on some data criteria. Holds the color code number.
DataDevice.State	Devices	Used by data devices.
DataDevice.State.Script	Devices	Used by data devices. Triggered by DataDevice.State change.
DataDevice.Tag	Devices	Used by data devices: unique identifier.
DateStamp.Cct	Design	This field is set by the program to the current date and time each time the circuit is modified. This is intended to allow scripts to determine if a design has been modified since the last time the script was run.
DateStamp.Dev	Devices	This field is set by the program to the current date and time when a device is placed in the circuit and not modified thereafter. See <a href="#">Device Date Stamping</a> .

## Reference

DateStamp.Last	Design	This field is used by the program as part of the date stamping process. See <a href="#">Device Date Stamping</a> .
DateStamp.Symbol	Devices	This field is set by the program when a symbol (part definition) in a library is modified. See <a href="#">Device Date Stamping</a> .
Delay	Devices	Internal usage, only in the Symbol Editor.
Depth	Devices	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.
Description	Devices	Used for saving and making visible device scope requests or initialization options. Used for EMTP.
Description.Cct	Design	
Description.Pt	Devices	Provides attribute position, scriptable.
Designer	Design	Intended to be used to display the designer's name on the sheet. See <a href="#">Creating a Title Block</a> .
DesignGlobalData	Design	Internal usage for global data transmission between devices
DesignGlobalDataOptions	Design	Internal usage for global data transmission between devices.
DesignType	Design	A value used to distinguish which design template was used to create the design.
DeviceDoc	Design	Used for setting device documentation location in some user-defined device masks.
DeviceVersion	Device	Provides a model version number.
DeviceVersionCompatibility	Devices	Allows to set device compatibility with software version. 1 means compatible, 0 means incompatible.
DevPrefix	Design	Default prefix for auto-assigned device names. Normally set using the "Device Naming" Options command. See <a href="#">Device Naming Options</a> .
Device.FullName	Devices	Device hierarchical name.
Device.Scope	Devices	Lists device scope variable group.
DevScript.Dev	Devices	Starts the specified script when a double-click occurs on the device. It is also possible to specify a ondblclick method in that script:

		<pre>function ondblclick(dblClickDevice, x, y) { // dblClickDevice is the device // x and y are device coordinates }</pre>
DrawingData	Devices	Used by some devices to keep specific symbol drawing data.
DWLSrcName	Devices	Internal usage, only in the Symbol Editor.
DWLSrcOS	Devices	Internal usage, only in the Symbol Editor.
DWLSrcPath	Devices	Internal usage, only in the Symbol Editor.
EMTP_lastRun_status	Devices	Used for parametric studies.
EMTPPhaseLabelA	Signals	Holds the phase label when a signal is 3-phase. Internal usage. The phase label is usually shown near the bus.
EMTPPhaseLabelB	Signals	Holds the phase label when a signal is 3-phase. Internal usage. The phase label is usually shown near the bus.
EMTPPhaseLabelC	Signals	Holds the phase label when a signal is 3-phase. Internal usage. The phase label is usually shown near the bus.
Exclude	Devices	<p>Indicates if the device is excluded from the Netlist (from the simulation data). Available values are "Ex" and "2".</p> <p>"Ex" means that the device is temporarily excluded.</p> <p>"2" means that the device is permanently excluded. This value is used, for example, for devices that are setting simulation options and should not appear as "parts" in the Netlist.</p>
Exclude.Script	Design	Names the script called when the Exclude value changes. It is held by the design for all devices.
ExtCctDate	Devices	Modified date of external design file containing device internal circuit. Normally set by the device Symbol Editor.
ExtCctLib	Devices	Name of library to search for an internal circuit. Used only by the Symbol Editor.
ExtCctName	Devices	Name of external design file containing device internal circuit. Normally set by the device Symbol Editor.
ExtCctOS	Devices	Source system type for design file containing device internal circuit. Normally set by the device Symbol Editor.
ExtCctPath	Devices	Directory path for the design file containing the device internal circuit. Normally set by the device Symbol Editor.

## Reference

ExportedMask	Devices	Used to export mask to upper level when set to 'Exported Mask'.
FormData	Devices Signals Design	Used for saving device data that is not necessarily sent to the Netlist.
FormData1	Devices	Used for saving device data that is not necessarily sent to the Netlist.
Function	Devices	<p>This device attribute can be set to OPTION or DataFile.</p> <p>When it is set to OPTION: the device data in ParamsA and/or ModelData attributes will appear at beginning of the Netlist. It is used for devices that are setting simulation options. Such devices can appear only at the top level of a design and not in subcircuits.</p> <p>When it is set to DataFile, then the ModelData attribute is used to provide the name of an external file. The contents of the external file are inserted directly into the Netlist file after the simulation options. Such a file can contain generic subcircuit definitions and include calls to other subcircuits. It can also contain the definition of simulation parameters. The simulation parameters must be always defined at the top of the Netlist file.</p> <p>It is allowed to use one or more devices with the Function attribute set to DataFile in a given design. Such devices can appear only at the top level of a design and not in subcircuits.</p>
Ground	Devices	List of pins to be hooked to ground in the Netlist. Not used in EMTP. Available in the Symbol Editor.
GlobalData	Devices	When this value is 1, it means that this device uses Global Data and needs automatic updating.
GlobalDataTag	Design	Internal usage for global data transmission between devices.
HideSBBRules	Devices	<p>Used in built-in device masks for blocking access to some mask data input sections. Also used to provide a user-defined title for the masked device. It is a comma separated string.</p> <p>The first part is for blocking mask sections. The following values are available:</p> <ul style="list-style-type: none"> <li>▪ 0: blocks the initialization section</li> <li>▪ 1: blocks the rules section</li> <li>▪ 12: blocks the rules and transmitted data</li> <li>▪ 012: blocks initialization, rules and transmitted data sections.</li> </ul> <p>The second (optional) string part is for providing the title.</p>

HierNameSep	Design	Contains the separator string used in generating hierarchical device and signal names for Report output. Normally set only by commands in a report script file. See the entry for the \$HIERNAMESeparator command in Export Script Language
InstName	Devices Signals	Reserved for future use.
LibDate	Devices	"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. See <a href="#">Update from Lib</a> .
LibName	Devices	Name of the library the device was read or updated from. This is used by the "Update from Lib" command. See <a href="#">Update from Lib</a> .
LibPath	Devices	Directory path of the library the device was read or updated from. This is used by the <a href="#">Update from Lib</a> command.
LibOS	Devices	Source system of the library the device was read or updated from. This is used by the <a href="#">Update from Lib</a> command.
LibType	Devices	The original name of the part when it was read from a library. This is used by the <a href="#">Update from Lib</a> command.
LineStyle.Line	Devices	Internal usage.
LineStyle.Sig	Signals	Internal usage.
Mask.Dev	Devices	Becomes 1 when the device (subnetwork) is masked with the default properties script.  Becomes 2 when the device (subnetwork) is masked with a user specified script.
Mask.Dev.Script	Design Devices	Names the script called when Mask.Dev changes. This is used to optionally clear mask data when unmasking.
ModelData	Devices	Used for saving EMTP device simulation data into the design and sending into the Netlist file.
ModelData.Pt	Devices	Position of attribute
ModelData1	Devices	Extra data for some devices.
ModelDataError	Devices	Set to 1 when the device has a data error. Can be used in search functions.
MPLLevel	Devices	Device patch version number.

## Reference

Name	Devices, signals	Device or signal name, (sometimes called "reference designator"). See Device Naming Options.
Name.Pt	Devices	Name attribute position, scriptable.
Name.Prefix	Devices	Prefix used to create default Name value. Used by Auto-name options. See <a href="#">Setting the Auto-Generated Name Format</a> .
Name.Script	Devices	Script applicable to name change
Name.Spice	Device	Reserved for future usage.
NominalVoltage	Signals	Signal nominal voltage.
NominalVoltageUnits	Signals	Signal nominal voltage units.
Notes	Design, Devices, Signals, Pins	User notes.
ObjectProps.Cct	Design	Internal usage.
ObjectProps.Dev	Devices	Internal usage.
ObjectProps.Line	Signals	Internal usage.
ObjectProps.Pin	Pins	Internal usage.
ObjectProps.Sig	Signals	Internal usage.
OKErrors	Devices, Signals, Pins	Used by the Find tool to mark errors set as OK by user. Should not be set manually. See <a href="#">Creating Scripts for the Find Tool</a> .
Package	Devices	Reserved for future usage. Available in the Symbol Editor.
Package.List	Devices	Reserved for future usage. Available in the Symbol Editor.
PageRef	Devices	Used in Page Connector pseudo-devices only. Contains the auto-generated page references. See <a href="#">Automatic Display of Page References</a> .
PageRefFormat	Design	Format string for page references. Normally modified only using the "Design Preferences" command. See <a href="#">Automatic Display of Page References</a> .
PageRefWidth	Design	Maximum number of items per line in page references. Normally modified only using the "Design Preferences" command. See <a href="#">Automatic Display of Page References</a> .
ParamsA	Devices	Used for saving EMTP device simulation data into the

		design and sending into the Netlist file.
ParamsB	Devices	Used for saving EMTP device simulation data into the design and sending into the Netlist file.
ParamsC	Devices	Used for saving EMTP device simulation data into the design and sending into the Netlist file.
ParamsEMTP	Design	Holds main simulation parameters, such as time-step and maximum simulation time.
Part	Devices	Part type code. Used in the Netlist to identify the device. EMTP uses this name to identify the device in its list of built-in models.
Part.Pt	Devices	Part attribute position.
Part.List	Devices	List of part types possible for this symbol.
PartTemp	Devices	Holds a unique part name for subcircuits. Avoids user errors when the user does not maintain unique part types correctly.
PatchLevel	Design	Design patch level.
Permutable	Devices	Reserved for future usage. Available in the Symbol Editor.
Phase	Devices Pins	Gives the number of device phases. This number is 3 when the device accepts 3-phase signals.  For pins, '1' means that the number of phases is blocked to 1 '3' means that the number of phases is blocked to 3 '1' means can be changed to 3 '3' means can be changed to '1' blank means flexible, can be changed
PinSequence	Devices	This field can be used to determine pin order in Netlist output formats that require specific device pin ordering. This is not defined in the standard EMTPWorks libraries. Not used internally. See the entry for the \$DEVPINSEQUENCE command in Export Script Language
PkgLevel	Devices	Reserved for future usage. Available in the Symbol Editor.
PkgPrefix	Design	Reserved for future usage.
Power	Devices	Reserved for future usage. Available in the Symbol Editor.
PrefixField	Design	Contains the name of the field to use as a name prefix for device auto-naming. See <a href="#">Setting the Auto-Generated Name Format</a> .

Reference

ProbeHotPt.Dev	Devices	Reserved for future usage.
ResizeInfo.Dev	Devices	Reserved for future usage.
Restrict	Devices	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 Properties command.
Revision	Design	Intended to be used to display the designer's name on the sheet. Not used internally. See <a href="#">Creating a Title Block</a> .
Scope_A	Devices	Used for holding programmable scope name on phase A or general signal device.
Scope_B	Devices	Used for holding programmable scope name on phase B or general signal device.
Scope_C	Devices	Used for holding programmable scope name on phase C or general signal device.
Script.Copy.Dev	Design	Names the script for copying device data. Called from device right-click "Copy/Paste" command.
Script.Edit.Des	Design	Reserved for future usage.
Script.ExInc.Dev	Design Devices	Names the script for excluding and including a device. Default value is found in Design attribute.
Script.Dev	Design	The name of a script to run each time a device is placed. Disabled in the current release.
Script.DevObj	Devices	Names a script that can be used to extract data and methods from the device. The name format is: xxx_d.dwj.
Script.Info.Dev	Devices	Names a script called when the user right-clicks on the device and selects "Extras/Summary".
Script.Info.Sig	Design Signals	Names a script called when the user right-clicks on a signal and selects "Signal Parameters".
Script.Mask.Dev	Devices	Names the mask script called when the subcircuit is double-clicked.
Script.Netlist	Design	Internal usage.
Script.Open	Design	The name of a script to run each time the design file is

		opened. Disabled in the current release.
Script.Open.Dev	Devices	Names the script called when the device is double-clicked.
Script.Paste.Dev	Design	Names the script for pasting device data. Called from device right-click "Copy/Paste" command.
Script.PhaseChange	Devices	Triggered when a phase change occurs.
Script.Pin	Design	Reserved
Script.Sig	Design	Reserved
Script.ss	Devices	When equal to '1', means that this device uses its own steady-state or load-flow visualization script.
ShowPhasorDev	Devices	Device steady-state or load-flow visualization options.
ShownPhaseVoltageAngleDigits	Signals	The number of digits in Phasor angle for phase voltage.
ShownPhaseVoltageDigits	Signals	The number of digits in Phasor for phase voltage.
ShownV0AngleDigits	Signals	The number of digits in Phasor angle for zero-sequence voltage.
ShownV0Digits	Signals	The number of digits in Phasor for zero-sequence voltage.
ShownV1AngleDigits	Signals	The number of digits in Phasor angle for positive-sequence voltage.
ShownV1Digits	Signals	The number of digits in Phasor for positive-sequence voltage.
ShownV2AngleDigits	Signals	The number of digits in Phasor angle for negative-sequence voltage.
ShownV2Digits	Signals	The number of digits in Phasor for negative-sequence voltage.
ShowPhasor	Signals	'1' means show positive sequence voltage phasor
ShowPhasorV0	Signals	'1' means show zero-sequence voltage phasor.
ShowPhasorV2	Signals	'1' means show negative-sequence voltage phasor.
ShowPhasorVabc	Signals	'1' means show a, b and c voltage phasors.
SimModelInfo.Dev	Devices	Reserved for future usage. Available in the Symbol Editor.
SignalType.Pin	Signals	Internal usage.
SignalType.Sig	Signals	Internal usage.

## Reference

SignalVoltage	Signals	Holds the signal voltage level selected from the "Signal Parameters" command.
SignalVoltage.List	Design Signals	List currently available voltage values for color coding signals. These values are available in the design attribute, but can be separately saved into the signal attribute. This is a comma separated list of values used in the script named by Script.Info.Sig.
SignalVoltage.Script	Design Signals	This script is called when SignalVoltage changes. It is named in the design.
SigPrefix	Design	Prefix used to generate default signal names. Normally set using the Signal Naming Options command. See <a href="#">Signal Auto-Naming Options</a> .
SimLoad	Design	Internal usage.
simulation_type	Design	Holds the EMTP simulation type.
Sniffer	Devices	Names the device script called when the device is clicked on.
Spice	Devices	Reserved for future usage. Available in the Symbol Editor.
SSPhasor	Signals	Attribute for visualization of steady-state or load-flow solution.
SSPhasor.Pt	Signals	Attribute position.
SSPhasorDev	Devices	Attribute for visualization of steady-state or load-flow solution.
SSPhasorDev.Pt	Devices	Attribute position.
SSPhasorPin	Pins	Attribute for visualization of steady-state or load-flow solution. Data is shown on pins for some devices.
SSPhasorPin.Pt	Pins	Attribute position.
Status	Devices	Used for updating global data in devices during global data transmission.
Status.Script	Devices	Updating method when Status changes.
statistical_option	Design	Specifies the statistical study options: 'Statistical case', 'Base case + Statistical case', 'Base case + Statistical case' or 'No random data'.
SteppedOut	Devices	Becomes 1 when the device is a control device and has the Step option turned on in its solution mode.

SteppedOut.Script	Design Devices	Script called when the SteppedOut attribute changes. Used for control devices and named in the design.
SubCctCode.Dev	Devices	Can contain the name of an external Netlist file. This attribute is use to insert external Netlist files through subcircuits. Pin names are used to interface with external Netlist signals.  This option is only for defining subcircuit Netlists. The subcircuit header for interfacing signals is automatically defined by EMTPWorks.
TestVectors.Dev	Devices	Reserved for future usage. Available in the Symbol Editor.
Timing.Save	Design	Reserved for future usage.
UserNamedValues	Design	Global named values
Unit	Devices	Reserved for future usage. Available in the Symbol Editor.
UnusedPins	Devices	List of unused pins in this part type. Not used internally.
user_fill_color	Devices	Saves the previous color when color coding is applied.
user_fill_color1	Devices	Saves the previous color when color coding is applied.
user_line_color	Devices	Saves the previous color when color coding is applied.
user_line_color1	Devices	Saves the previous color when color coding is applied.
Value	Devices	Used to keep and show some device data on the design drawing.
Value.Pt	Devices	Attribute position.
Value1	Devices	Used to keep and show some device data on the design drawing.
Value1.Pt	Devices	Attribute position.
Value2	Devices	Used to keep and show some device data on the design drawing.
Value2.Pt	Devices	Attribute position.
ValueDisplayOpt	Devices	Reserved for future usage. Available in the Symbol Editor.
View.PowerFlow	Design	Options for View Steady-State
View.SteadyState.on	Design	On or off for View Steady-State, Enable or Disable.
Visual.Dev	Devices	Becomes 1 when the subcircuit is set to be visual.

XMLDataGrids	Devices	Device data saved with XML content. For saving grid data.
--------------	---------	---

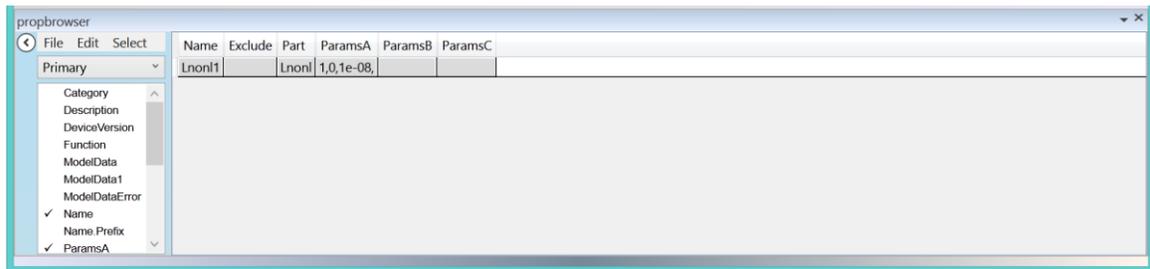
## 7.27 Property Browser

### 7.27.1 Browser for attributes

The Property Browser (also called 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. The nets show all the pin connections on a signal.

### Opening the Browser

To open the Browser, select the Property Browser item in "Home>Find". This opens a spreadsheet window.



Close Box	Click this control (x) to close the Browser window. Another method of closing is to select the Browser item in the View menu.
File Menu	This menu contains the Save As command, which allows you to save the contents of the spreadsheet as a text file.
Edit Menu	This menu contains the standard clipboard editing commands which allow you to operate on selected cells in the spreadsheet.
Select Menu	This menu contains options to control the type of objects displayed and how they are selected.
Attribute Type Selector	<p>This list allows you to select what type of fields are shown in the Attribute List. By default, only Primary fields are shown, i.e. attribute fields marked as "Primary" in the <u>Defining Attribute Fields</u> command. You can also select to show:</p> <p>Secondary fields: fields not marked as "Primary"</p> <p>Pseudo attributes: Text values that are associated with an item by the system, but are not normal attribute fields, including the internal part name and graphical information.</p> <p>Definition fields: All attribute fields not marked</p>

	<p>"Keep with Instance" in the "Define Attribute Fields" table.</p> <p>Instance fields: All attribute fields marked "Keep with Instance" in the "Define Attribute Fields" table.</p>
--	--

By default the Browser 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. You can select what must be shown using "Select>Object Type".

### **Closing the Browser**

To close the Browser window, either click in the Close box (the small X button at the corner of the window), or select the Browser item again in the View menu.

### **Updating the Browser Window**

The Browser will update any time a change occurs in the schematic that affects the data displayed in the spreadsheet, or when its scope is set to Circuit and the current circuit changes.

### **Selecting the Type of Objects Displayed**

The Browser is controlled by the separate File, Edit and Select menus on the Browser panel itself. The menus in the main application menu bar do not affect the Browser. Specifically, the Cut, Copy and Paste edit menu items in the main menu bar will still operate on the selected circuit items and not on the spreadsheet.

The "Object Type" submenu under Select allows you to change the type of object being displayed. There are eight different types of objects that may be displayed Devices, Signals, Pins, Nets, and selected Devices, Signals, Pins, and Nets.

If the "Selected" form of an object is chosen to be displayed, e.g. "Selected Devices", then only items which are selected in the schematic will be displayed. Similarly, if "Selected Signals" is chosen in the "Object Type" menu, then only signals that are selected in the schematic will be listed in the spreadsheet.

### **Determining Where to Search for Objects**

When looking for objects, the Browser will limit its search scope by using the selection made in the scope menu pop-up. There are four different scopes:

Design	In this scope the entire design hierarchy is searched for objects of the correct type. This scope has a side-effect. Some of the objects attribute values displayed may not be editable because of definition/instance conflict resolution. The uneditable objects and/or fields will be displayed with a gray background.
Circuit	In this scope the search is limited to the pages of the circuit with the most active window.
Here Down	In this scope the search for objects is in the circuit with the active most window and in all circuits below it in the hierarchy. Some of the objects and/or fields may not be editable for the same reasons as explained in Design scope.
Page	This scope only looks for objects in the circuit with the most active window and in addition only looks at the

page associated with that window.

### Displaying Attributes

To add an attribute field to the spreadsheet, simply double-click its name in the attribute list.

The Nets and Selected Nets object types display a different format in the spreadsheet and do not allow attributes to be added.

The attribute field names appearing in the list will change as the type of object selected for display is changed. Only attributes that apply to a given object type are displayed. For example, the attribute field "Part" is associated with Devices but will not be associated with Signals.

If an attribute name is checked in the attribute list then it will be added as a new column at the end of the spreadsheet. Double-clicking a checked attribute name removes the associated attribute from the spreadsheet. The sorting is based on the first column. You can change the ordering by moving the columns.

### 7.27.2 Changing Attribute Values

#### Selecting and Editing Cells

To edit a cell, just click on it and start typing. Cells that are shaded gray are displaying values that cannot be edited in the spreadsheet. This could be the case if a given circuit or subcircuit is locked (read-only).

Modifying the value of a cell will cause the corresponding attribute value of the object in the schematic to change!

The schematic data is **updated immediately** and there is no **undo**.

#### Using the Edit Menu Commands

The Edit menu in the Browser panel contains the standard clipboard editing commands Cut, Copy, Paste, and Clear, plus four additional menu items:

- |                |  |
|----------------|--|
| Fill Down      | The Fill Down command takes the selection rectangle and copies the values in the cells at the top of the selection to all the selected cells below them.   |
| Fill Right     | The Fill Right command takes the selection rectangle and copies the values in the cells at the left of the selection to all the selected cells to the right of them.   |
| Make Visible   | The Make Visible command takes the object attributes affected by the selection rectangle and makes them visible in the schematic. This command applies only to attribute values (not pseudo-attributes like the part name) and only to visible objects. For example, it is not possible to display attributes associated with bundle internal signals. |
| Make Invisible | The Make Invisible command takes the object attributes affected by the selection rectangle and makes them invisible in the schematic.  |

To select several cells you can click hold and drag. You can also select the first cell and use the shift button to select a range of cells.

### Disallowed Editing Changes

Any editing change that may cause a structural change to the connectivity of the schematic (although legal under some conditions) 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.

### 7.27.3 Saving and Printing Data in a Browser Window

The "Save As" command in the Browser's File menu provides a method of directly saving data that appears in the spreadsheet display. There is no direct printing command to print the contents of the spreadsheet, but you can open the resulting test file in any text editor or word processor to format it for printing. If it is required to automate such a task, then a better approach is to use an Export script.

### 7.27.4 Browser Clipboard Data Format

The Browser exports data in a tab-delimited text format commonly used by spreadsheet and database programs. The following points summarize this format:

- Items appearing on one line in the spreadsheet display appear on one text line.
- Items on one line are separated by a tab character.
- Lines are terminated by a single carriage return character.
- If an item contains anything other than alphanumeric characters and spaces it is enclosed in double quotes. If it contains a double-quote character, that character is doubled, i.e. it is preceded by a second double- quote.

Note that attribute items can themselves contain carriage returns and other control characters. Such items will be enclosed in double-quotes in the clipboard text.

### 7.27.5 Showing Objects in the Schematic

When the "Show in Schematic" option in the Select menu is checked, the Browser will always attempt to display the associated schematic object whenever a cell is selected in the spreadsheet.

The Browser may sometimes not be able to show the object. The reason for this is generally that the object was not in an open window, or does not have a visible representation, e.g. bus internal signals.

### 7.27.6 Sorting Displayed Objects

Items are always displayed in the spreadsheet sorted in order by the first column. If the values in the first column are the same, the second column is used, etc. The sort order can be changed by re-ordering the columns so that the primary sort value is in the first column. You can use attributes to create a grouping function. If the Category (available to the user, see [Predefined Fields](#)) attribute is selected for categorizing devices, then it can be made as the first column in the spreadsheet to show devices sorted within this grouping.

### **7.27.7 Adjusting the Spreadsheet**

#### **Resizing Columns**

Columns may be resized by moving the cursor into the column heading area and positioning it over the vertical dividing line. The cursor when correctly positioned it will change shape. Press the mouse button and drag the line to size the column to its desired size. Rows cannot be resized.

#### **Moving Columns**

Columns are reordered by moving the cursor into the column heading area and positioning it over the column heading to move. Click once on the column label to make the column become selected, then hold and drag the label to its new position.

It can be useful to move columns so that the sorting mechanism will base the sort on the new ordering of the columns

## 8 Subcircuits and Hierarchy

### 8.1 Subcircuits and Hierarchy

#### 8.1.1 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 a "hierarchical block".

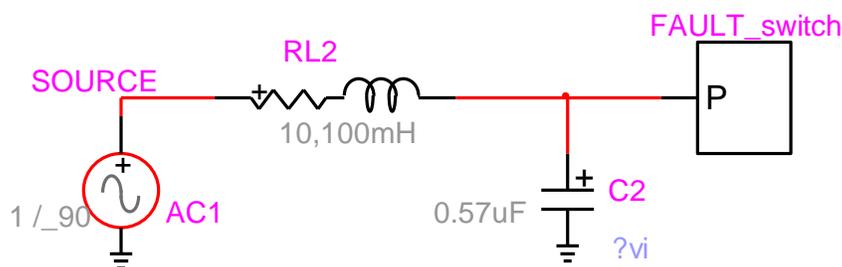
Subcircuits (also called subnetworks) are the building blocks of hierarchy.

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 details.

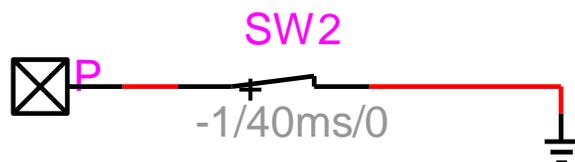
Hierarchical design in effect adds a "third dimension" to a schematic diagram. It also raises some complex issues that should be understood before embarking on a major design.

**Please review this chapter carefully before making extensive use of the hierarchical features of EMTPWorks and manipulating subcircuits.**

A simple example is given by this circuit.



The **FAULT\_switch** is a subcircuit and contains the following circuit:



The subcircuit is located in a parent circuit. It has a parent device (its symbol with optional pins). The subcircuit contents are actually the contents of the parent device, but to simplify the terminology, the subcircuit parent device is called "subcircuit" and its contents are also called "subcircuit".

## Reference

EMTPWorks offers several methods for creating subcircuits. Subcircuits can be created from any location in a circuit. It is also possible to make a subcircuit from the entire circuit.

The subcircuit above is created from the switch device connected to ground. The "Options>Subcircuit>Create Subcircuit Block" (Ctrl+Shift+Q) menu is used to create a subcircuit from parts of a circuit. If port connectors (the interface between the subcircuit contents and the holding circuit) are not placed before using Ctrl+Shift+Q, then EMTPWorks suggests automatic port positions, such as "Add port connectors to all signals with visible names" or "Add port connectors to all signals with loose ends".

A subcircuit may have zero, one or more pins. Any pin types can be used.

The top circuit can contain several subcircuits and each subcircuit can also contain subcircuits with unlimited number of levels.

The default subcircuit symbol is a square box. The Symbol Editor can be used to modify the subcircuit symbol.

After creating the subcircuit it can be opened in a separate circuit drawing by double-clicking on its symbol. The available keyboard shortcuts are: "Ctrl+Shift+I" (in) for getting into the subcircuit and "Ctrl+Shift+U" (up) for going up.

The first time a subcircuit is entered it is locked. It is needed to unlock the subcircuit for making changes. There are two methods for unlocking: through the "Circuit Info" menu (Ctrl+I) or by moving a device and accepting to unlock from the subsequent panel popup.

Subcircuit creation and modification methods are illustrated in Tutorials 3 to 5 (See [Tutorial 3: Creating a Subcircuit Bottom Up](#) as a starting point).

After the subcircuit is created, it may be required to modify its port interface by adding or deleting pins. These procedures are explained in [Tutorial 5: Advanced Subcircuit Topics](#).

## 8.2 Subcircuit Uniqueness and Encapsulation

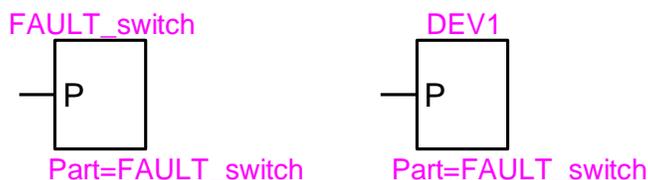
You must read this section before using subcircuits.

The contents of a subcircuit are hidden (encapsulated) from the holding circuit or subcircuit. The holding circuit is also called the parent circuit. The only connection between the parent and its subcircuit is through the interfacing pins. This means that two signals may have the same name in the parent circuit and in the subcircuit, but they are not connected. The global naming system used in EMTP is based on concatenating a subcircuit path name to all its internal names. If the device named SW2 is located in a subcircuit named FAULT\_switch then its global name becomes FAULT\_switch/SW2. If there are more levels, then the path name changes to account for all levels. In FAULT\_switch/DEV1/XX1/power\_line, the power\_line signal is located in the subcircuit XX1 which is located in the subcircuit DEV1 and which is located in the subcircuit FAULT\_switch.

The following options are available:

- A subcircuit may have one or more pins. Standard pins and bundle pins may be used.
- A subcircuit may have no pins. In which case it is an electrically stand-alone circuit.
- A subcircuit can be also made of non-electrical objects.
- A subcircuit may hold (be parent to) one or more subcircuits.
- A subcircuit may be multipage (see [Adding, Deleting and Titling Circuit Pages](#)).
- Subcircuits can be created from any point in a design and from any part of a circuit. The subcircuit interface pins are fully controllable. It is also allowed to create subcircuits from entire designs.

When a subcircuit is created and duplicated, EMTWorks continues keeping only one copy of its internal circuit. **This means that any changes inside one subcircuit are automatically reflected into the other subcircuit instances (copy).** EMTWorks also sees only one copy of the circuit, but two separate calls. This is similar to programming using a given function from different locations in the program. The two subcircuits of below are identical. They have the same Part attribute and same Type name. This is a powerful feature for large designs. If a large number of subcircuit devices of the same type are used in a design, then by changing only one device allows changing all other devices automatically.

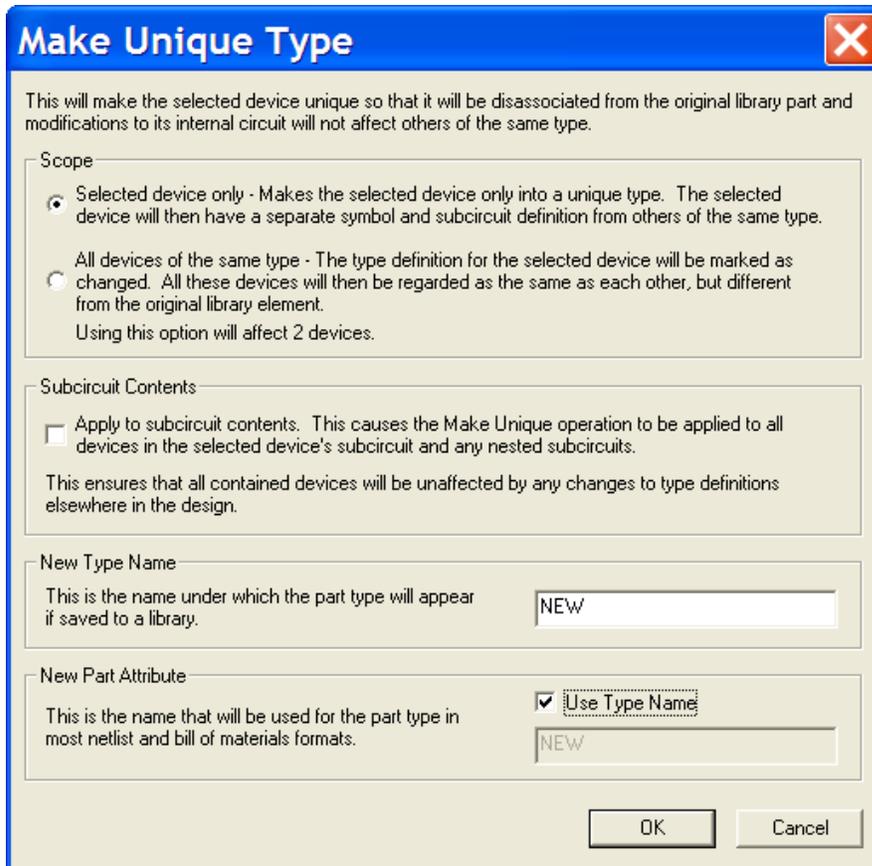


The Type name of a device can be found by selecting the device and using the right-click Properties command.

It is very important to understand the implications of the "Make Unique Type" command and the notion of subcircuit uniqueness before making any changes to subcircuits. **The user should never modify a built-in EMTWorks subcircuit based device before applying the "Make Unique Type" command to the device.** A built-in device may be used by other devices and thus modify their contents if not made unique before changing.

### 8.2.1 Make Unique Type

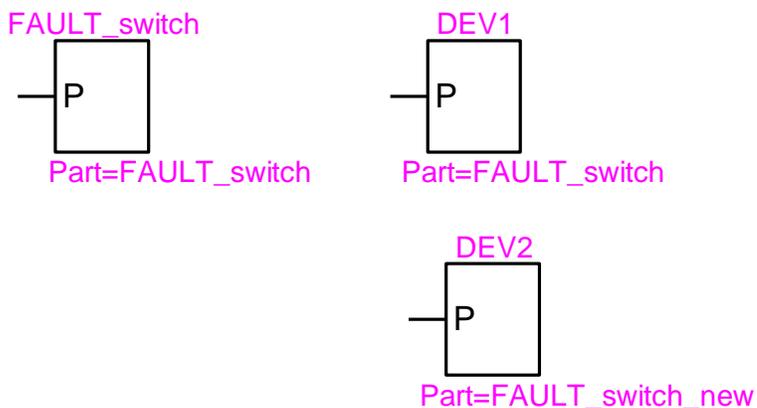
If it desired to detach one subcircuit definition from the other, then the detached subcircuit must be made unique. This is completed through the "Options>Part Type>Make Unique Type" menu.



The options available on the above panel are self-explanatory.

In the figure below, the subcircuit DEV2 is first created by duplicating DEV1 and then made unique. To ensure circuit data integrity and to manage subcircuits, it is very important to enable visual feedback for distinguishing dissimilar subcircuits. A suggested approach is to show the Part name. An even better approach is to change the subcircuit symbol using the Symbol Editor.

EMTPWorks also keeps a separate attribute PartTemp for verifying uniqueness when the user makes mistakes by using existing part names or makes attribute changes without making a unique type.



At this stage DEV2 is detached from DEV1 and FAULT\_switch: it has its own copy of the original subcircuit. This is similar to making a copy of a function and renaming it to allow new contents, calling methods and usage.

If a subcircuit is available in a library and if it is dragged into the design and modified, then it becomes automatically unique: detached from the library copy. The user should change the symbol and make the new Part name visible to avoid confusion. Any new copy dragged in from the original library remains detached from the one modified in the design.

Locking the contents of a subcircuit after its creation is used for reminding the user about the potential repercussions of changes made in the subcircuit. See [Locking and Unlocking Subcircuits](#) for more information on this option.

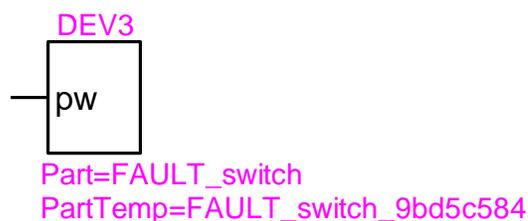
It is allowed to add and delete subnetwork pins after subnetwork creation. When a signal line type is changed in a subnetwork it will not propagate to its parent circuit. Such changes must be performed manually.

In the current version of EMTPWorks, subcircuit control input and output pins are not automatically given an arrow tip. This is only achievable by modifying the subcircuit symbol using the Symbol Editor.

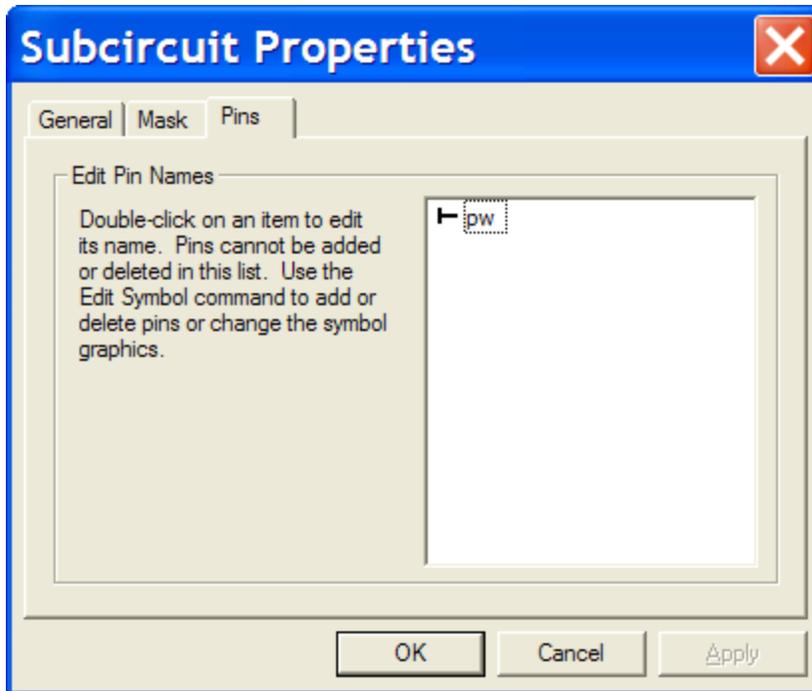
If after exiting a subcircuit, the user receives an error message, it means that there is pin interface corruption in the subcircuit and the resulting Netlist will also have problems. Error and warning messages should not be ignored before continuing.

### 8.2.2 PartTemp attribute

As explained earlier, in addition to the Part attribute which is used to distinguish subcircuits, EMTPWorks uses one more attribute: PartTemp. This attribute is used to verify uniqueness when the user makes changes that may affect the uniqueness of a subcircuit. An example of change that automatically creates a new copy of a subnetwork is shown below.



The original device FAULT\_switch is copied and then its pin name is changed. It automatically changes the internal port and makes DEV3 different from FAULT\_switch. EMTPWorks modifies the PartTemp to account for this condition. Notice that the PartTemp attribute is corrected only after the first subsequent Netlist generation. The pin name can be changed by selecting the subcircuit, opening the right-click menu "Subcircuit Info" and moving into the Pins tab.



The listed pin names can be clicked on and modified.

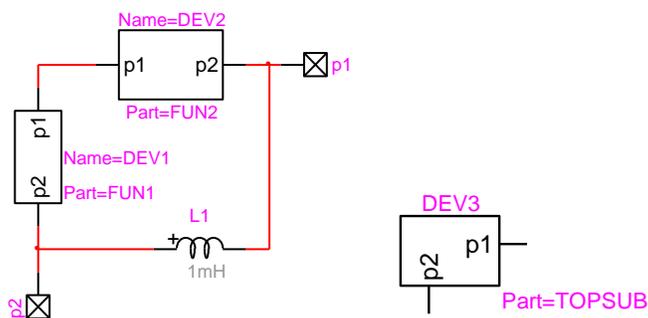
### 8.2.3 Subcircuits within subcircuits

As understood in the introduction to this section, a subcircuit may contain other subcircuits and the contained subcircuits may contain subcircuits, which results into a tree of subcircuits.

The best way to understand multilevel designs and Unique property of a subcircuit, is to view a subcircuit as a program function. A function is a program building block. A given function has only one version and can be called from other functions. If it is desired to change a function, then its name must be changed (made unique) in order to change its contents. A subcircuit level can be viewed as the body of the function.

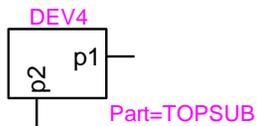
The best way to illustrate this concept is to create a simple example.

Two subcircuits FUN1 and FUN2 are contained in a subcircuit TOPSUB.

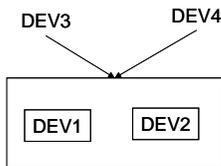


Initially there is only one copy of TOPSUB in the memory. There is also only one copy of FUN1 and one copy of FUN2. Making any changes in the TOPSUB level is automatically reflected into all copies of TOPSUB. The same is true for FUN1 and FUN2.

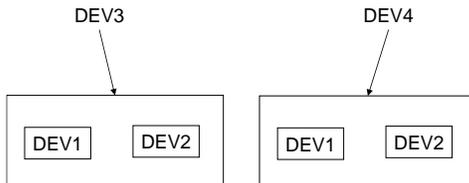
If we make a copy of TOPSUB to create DEV4 it is like calling the same function tree from another location in the main program (the top circuit).



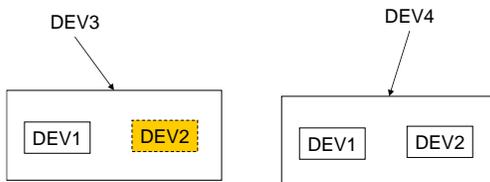
Making any changes in the DEV4 level is automatically reflected into DEV3 level. If it is desired to modify FUN2 in DEV3 without modifying FUN2 in DEV4, DEV3 must be first converted to become unique using the **Make Unique Type** procedure. Its level will then become separated in memory from DEV4. Then DEV2 can be also set to a new type using again **Make Unique Type**. This last action will detach DEV2 and allow modifying its level contents. Here is a visual representation of the steps. First DEV3 and DEV4 are sharing the same body:



Then DEV3 becomes unique, holds its own body, but continues sharing DEV1 and DEV2:



The next step detaches DEV2 by making it unique.



This means that if you want to modify a multilevel subcircuit to make it completely detached from others in all its levels and used subcircuits, you must modify all hierarchical levels.

It is also feasible to propagate uniqueness from top down for a subcircuit device containing other subcircuits by checking the option "Apply to subcircuit contents" on the Make Unique Type command panel.

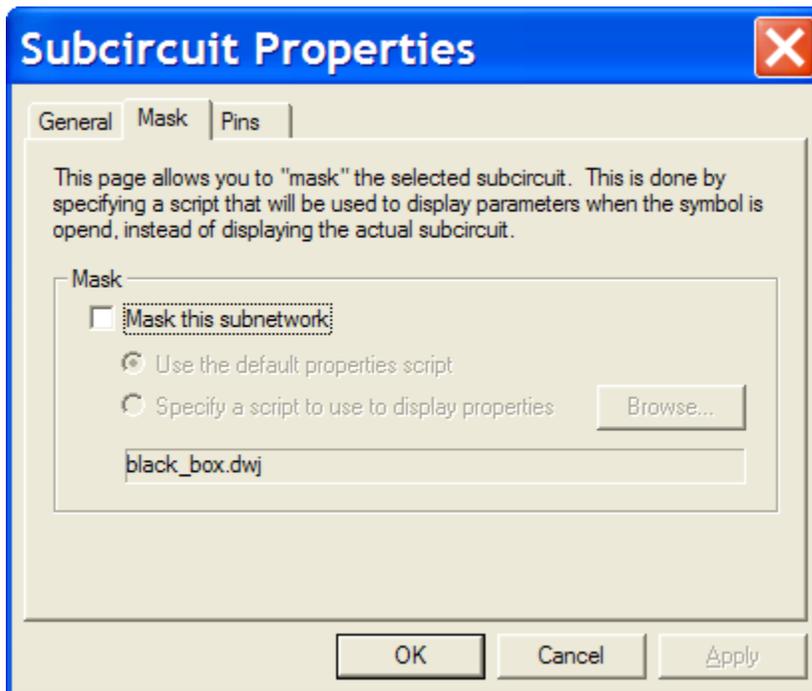
### 8.3 Subcircuit Properties

When a subcircuit is created and right-clicked, the "Subcircuit Info" menu becomes available:



The "New Type Name" input field allows changing the type name. This will change the Part attribute of the subcircuit. The second radio button ("Make the selected...") is similar to the "Make Unique Type" command available from "Options>Part Type". If you select this option and keep the same type name and if other subcircuits of the same type are available in the design, then EMTWorks will use the PartTemp attribute to distinguish this device.

The Mask tab allows masking the subcircuit and selecting a mask name. Masks are explained in [Masking](#).

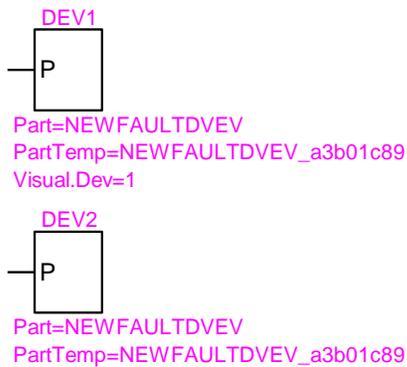


The Visual tab has been disabled starting in version 2.1. The Visual.Dev attribute has been kept for compatibilities with previous versions only. It is not recommended to use this attribute for new subcircuits.

If you copy or duplicate a visual subcircuit you must remember that all its signals are visible and thus parallel connections may result. It is recommended to visually distinguish visual subnetworks by changing the subnetwork symbol or by showing the attribute Visual.Dev. This attribute becomes 1 when the visual option is turned on.

Beside the attribute Visual.Dev there is no other indication in EMTPWorks for a subcircuit being visual. It is actually in the EMTP code, that subcircuit signals are brought to the parent level, by simply deleting the subcircuit path name. Only one path level is deleted, since the visual subcircuit may be nested in another subcircuit.

When there are two subcircuit devices of the same type and one of them is made visual, EMTPWorks keeps the same type, as shown in the example below.

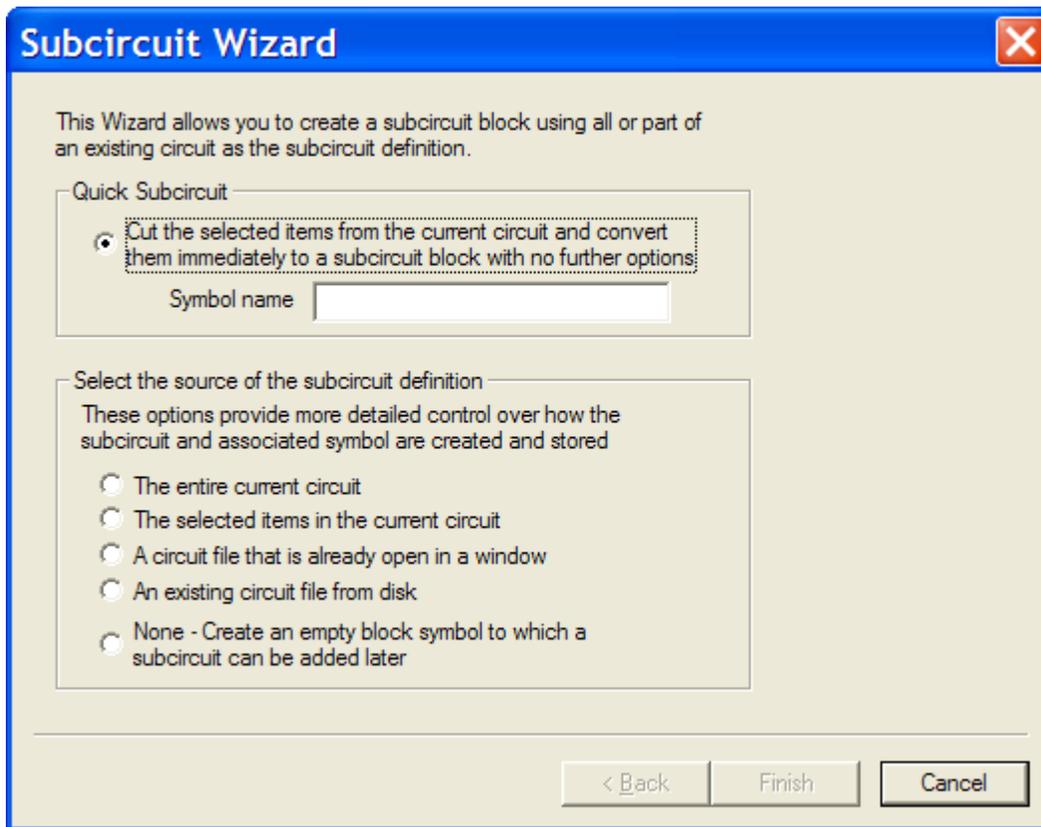


DEV1 is visual, but has not been physically separated from DEV2, so that any changes in DEV2 are reflected back into DEV1 and vice-versa.

## 8.4 Subcircuit creation methods

EMTPWorks offers a large number of methods for creating and maintaining subcircuits.

The starting point is the "Subcircuit Wizard" opened from "Options>Subcircuit>Create Subcircuit Block":



The "Quick Subcircuit" option is activated only when a device or several interconnected devices are selected in the design that will become the parent of the subcircuit.

Subcircuits can be created "top down" or "bottom up".

Working top down means creating the highest level of the design first and then working downward to more detailed levels. In practice, this means that you create and place the symbol for a hierarchy block first, before you necessarily even know what the internals of the block will look like. Once you have completely created the higher-level circuit including the symbols for all the hierarchy blocks it uses, you then proceed to define the internal circuits of the blocks.

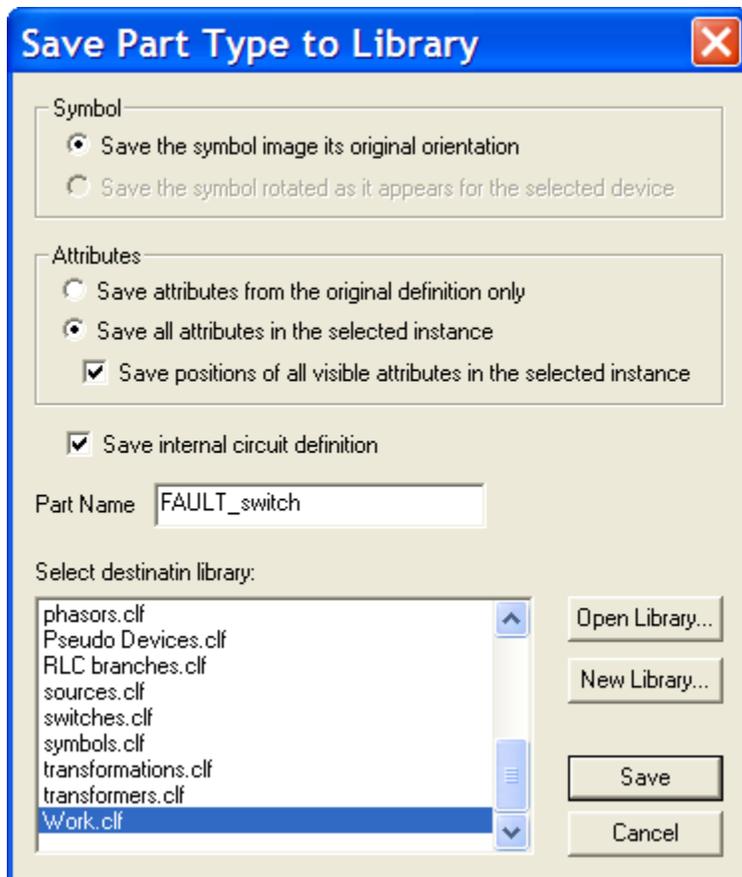
The top down approach is illustrated in [Tutorial 4: Creating a Subcircuit Top Down](#).

In the "bottom up" approach any part of an open design or entire designs can be made into the subcircuit of a hierarchical block symbol. The block is automatically created by the program, but can be modified using the Symbol Editor.

The bottom up approach is illustrated in [Tutorial 3: Creating a Subcircuit Bottom Up](#).

## 8.5 Saving Part Type to Library

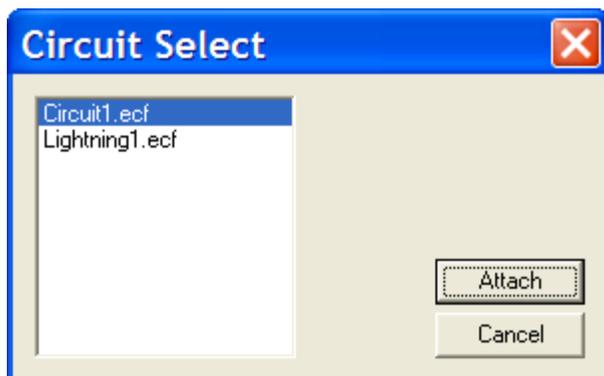
After creating a subcircuit you can save it into a library for later usage or for sharing with other users. To place a subcircuit into a library, simply select the subcircuit symbol and select the menu "Options>Part Type>Save to Lib":



Since the subcircuit is a part with an internal circuit, you must select "Save internal circuit definition". If this option is not checked the subcircuit contents will not be saved with the symbol!

## 8.6 Attaching Subcircuit

The "Attach Subcircuit" command (from Options>Subcircuit) allows you to select an open design to attach as a subcircuit to the selected device. When this command is selected, the following box will appear:



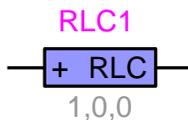
## Reference

The listed circuits are those currently open in EMTPWorks.

Clicking Attach on the Attach Internal box will cause the following actions to be taken:

- 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. This allows the definition of this type to be temporarily modified by the addition of the subcircuit. The other devices of the same type will be updated when you close the attached subcircuit.
- 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.
- 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.
- 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.
- The title of the internal circuit is updated to reflect its hierarchical position in the master design.
- 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.

It is noticed that you can attach a subcircuit to any device, including predefined devices from built-in libraries. The RLC device, for example, can be dragged in and modified to become a hierarchical block.



When you attach a subcircuit to an existing device, it automatically inherits the device pins for the underlying subcircuit. It is your responsibility to make the necessary connections and add the necessary ports.

## 8.7 Detaching Subcircuit

The "Detach Subcircuit" command (from Options>Subcircuit) makes the currently displayed subcircuit into a separate design and redefines the parent device as having no internal circuit.

This operation permanently removes the subcircuit 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 Type" command to isolate the selected device first.

The Detach operation cannot be Undone!!!

In particular, Detaching performs the following operations on the subcircuit displayed in the front most window:

- The circuit is unlinked from its parent device, making it into a separate design.
- The title of the subcircuit is set to a default name.

- The internal circuits of all other devices of the same type in the design are removed.

## 8.8 Discarding a Subcircuit

The "Discard Subcircuit" command (from Options>Subcircuit) discards the subcircuit associated with a parent device. To perform this action:

- ❑ Locate and select the parent device in its circuit.
- ❑ Select the "Discard Subcircuit" command in the Subcircuit menu.

All data associated with the subcircuit is destroyed by this command. This cannot be undone!

The "Discard Subcircuit" command redefines the selected device (and all others of the same type) as having no internal circuit.

## 8.9 Locking and Unlocking Subcircuits

A subcircuit can be locked to prevent accidental opening. This is intended for cases where the subcircuit is derived from a library or is used only to implement a simulation model which should not be edited as part of the parent design.

To lock and unlock a subcircuit from the schematic:

- ❑ Locate the subcircuit's parent device symbol in the schematic and select it.
- ❑ Select the Properties command in the subcircuit right-click menu.
- ❑ Check or uncheck the "Lock opening subcircuit" box, as desired.
- ❑ Click OK

## 8.10 Subcircuit decoding

For each subcircuit in EMTPWorks, EMTP assigns a unique instance code (tag). Each instance code corresponds to a subcircuit Pathname. This is valid for unique and non-unique subcircuits.

You can see the list of Subnetwork pathnames in the case Web. The Case Web is available at the end of a simulation. An example of listing is shown below:

Subnetwork pathnames	
Instance code	Pathname
-999	Substation_A/
-998	Substation_C/
-997	Substation_B/
-996	Ser_C_1/
-995	Ser_C_1/DEV1/
-994	Ser_C_2/
-993	Ser_C_2/DEV1/
-992	YgDD_np1/
-991	YgDD_np1/xfmrA/
-990	YgDD_np1/xfmrB/
-989	YgDD_np1/xfmrC/

## Reference

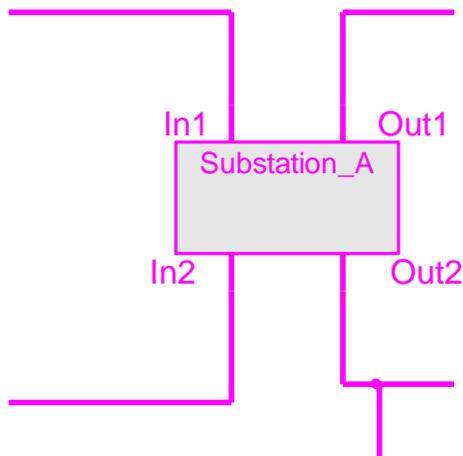
-988	YgDD_np2/
-987	YgDD_np2/xfmrA/
-986	YgDD_np2/xfmrB/
-985	YgDD_np2/xfmrC/
-984	YgYgD_np1/

The instance code is an internal number. All subcircuit devices are tagged with this number. If, for example, a device named RLC1 is located in the subcircuit Substation\_A, it will be internally given the name RLC1\$-999.

Only the top level circuit devices and signals are not given an instance code.

The same method is applied to subcircuit signals, but signal decoding is more complex and the user must be more careful to avoid illegal or undesirable connections. All signal names are visible only within the scope of the subcircuit. A signal named BU1 appearing Substation\_A is tagged as BU1\$-999 and is different from the signal BU1 appearing in Substation\_B, in which case it is tagged as BU1\$-997. This is conformal to hierarchical methods in EMTP.

The extra complexity rises from the fact that a subcircuit may have pins which allow propagating signals between subcircuits. In this example, the signal (hidden name s37) connected to the pin In1 propagates into Substation\_A and keeps its name. Any signal in Substation\_A connected to the pin In1 automatically inherits the name s37. A signal can propagate down through pins from the top circuit level or from a subcircuit containing one or more subcircuits.

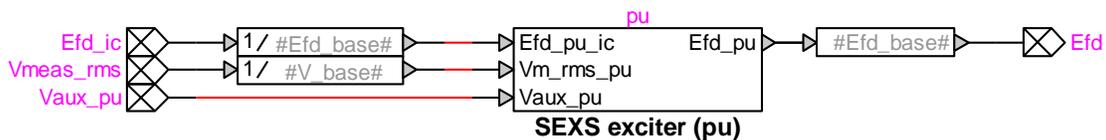


Some signal names can become swapped (shorted) due to internal connections in the subcircuit. Although most of these connections are perfectly legal, EMTP shows the 'Shorted (redundant) signals' in the Case Web. An example of listing is shown below.

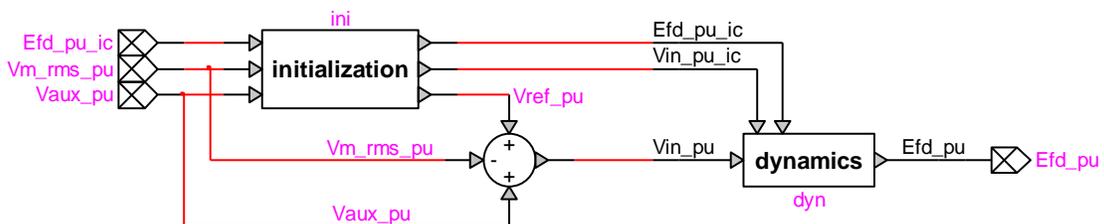
<b>Shorted (redundant) signals</b>	
<b>Signal name</b>	<b>Actual name (connected to)</b>
GND	Ground, 0 volts
Efd_pu_ic_2\$-973	Efd_pu_ic\$-974

Efd_pu_ic_2\$-967	Efd_pu_ic\$-968
Efd_pu_ic_2\$-961	Efd_pu_ic\$-962
Efd_pu_ic_2\$-955	Efd_pu_ic\$-956

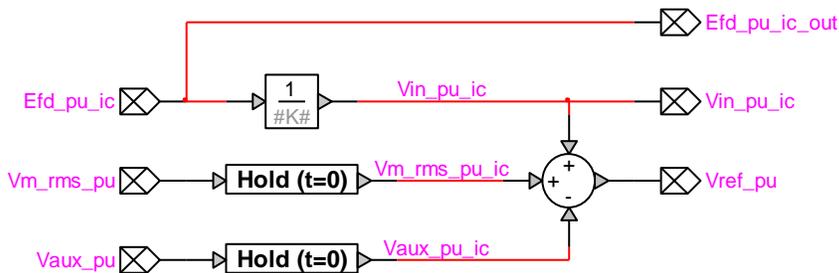
The GND node is always listed to recall that any signal named GND is automatically connected to ground. In this case it is indicated that the signals Efd\_pu\_ic\_2\$-973 and Efd\_pu\_ic\$-974 are connected together. The subcircuit \$-974 is the top level subcircuit sexsExciter\_1 shown below.



It contains the subcircuit tagged as \$-973 (AVR\_1/sexsExciter\_1/pu/)



in which the subcircuit "ini" (shown below) is connecting the input from Efd\_pu\_ic to the output top pin.

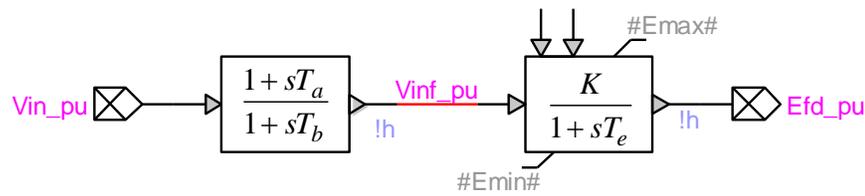


EMTP also messages floating signals found in subnetworks. A floating signal is a signal that enters a subnetwork through a pin but does not connect anywhere (not directly and nor virtually). Some floating signals can be used as variables in control blocks. This is the case of the subcircuit "dyn" example:

## Reference

$E_{fd\_pu\_ic}$   (used as i.c. for  $E_{fd\_pu}$ )

$V_{in\_pu\_ic}$   (used as i.c. for  $V_{in\_pu}$ )

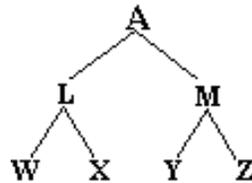


## 8.11 Printing Hierarchical Designs

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

### 8.11.1 Determining Print Page Order

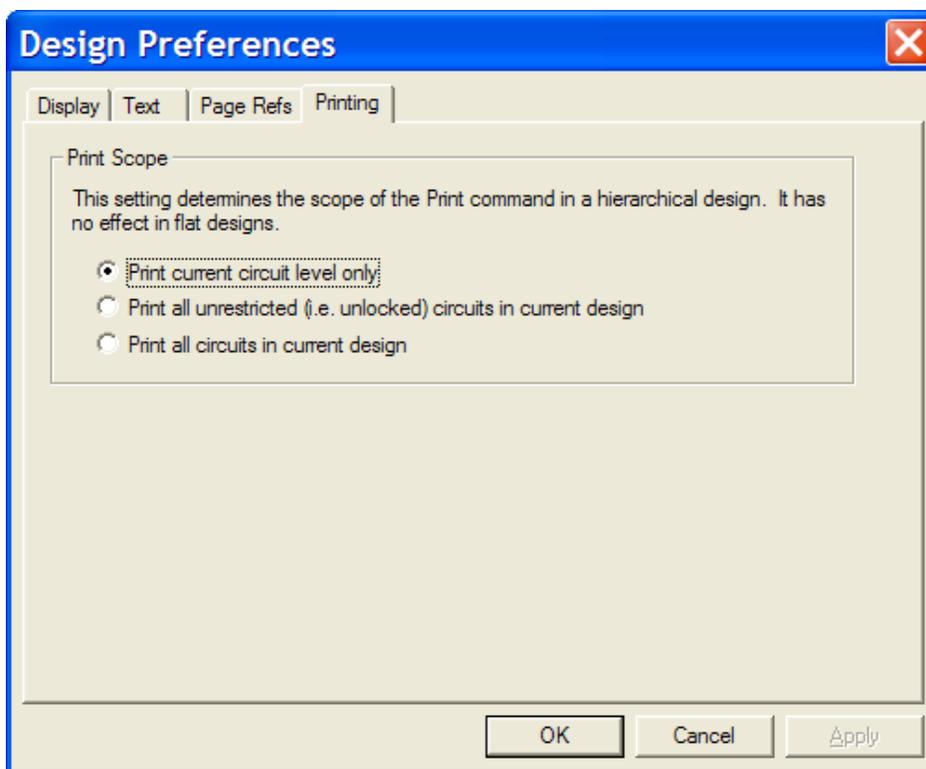
When a design is printed, the master circuit of the design is printed first in its entirety. Then, each hierarchical block is printed, followed by all hierarchical blocks it contains. For example, given the following structure of nested hierarchical blocks:



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. W and X could be reversed above.

### 8.11.2 Design Preferences: Printing

To set the scope of printing in a hierarchical design, choose the "Design Preferences" command in the "Options>Design" toolbar and select the Printing tab.



Three options are available:

- **Print current circuit level only:** With this setting, the Print command will print only pages belonging to the circuit level in the current window.
- **Print all unrestricted circuits in current design:** 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 Properties command for the parent device.
- **Print all circuits in current design:** Print the design's master circuit and all internal circuits, regardless of "locked" status.

### 8.11.3 Printing Sequential Page Numbers in a Hierarchical Design

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

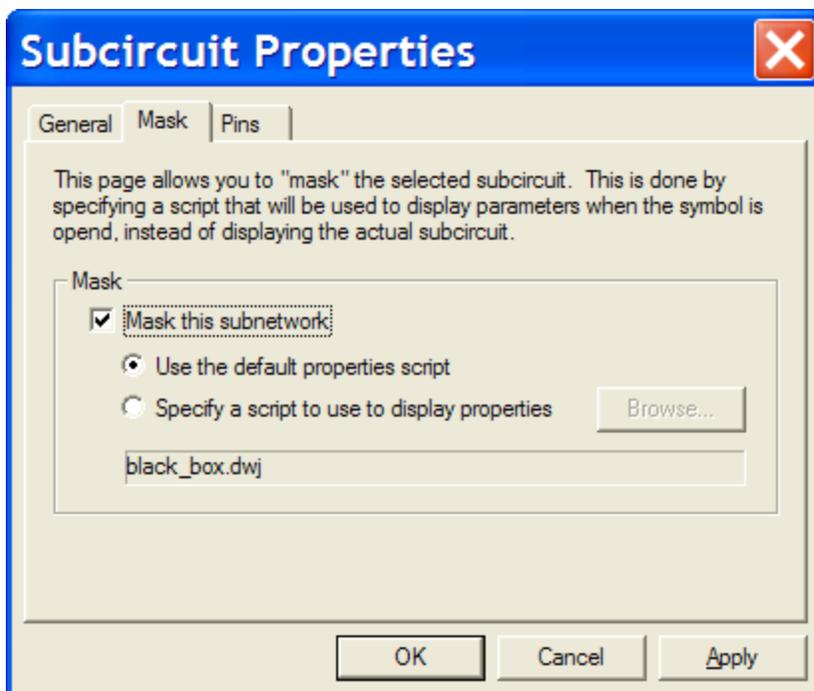
This text variable only applies during a Print operation. When it is drawn on the screen, it is interpreted the same way as the \$PAGENUMBER variable, i.e. it gives the page number within the circuit.

- ❖ See more information on text variables in [Using Text Variables](#).

## 8.12 Masking

### 8.12.1 Masking

After creating the subcircuit the right-click device menu item "Subcircuit Info" becomes enabled. The "Subcircuit Properties" has a Mask tab with a default mask:



The default mask becomes selected when you click on the checkbox "Mask this subnetwork". If you select to specify another scripts "Specify a script to use..." then you must use the Browse button to locate and select that script. The available scripts listed below can be found in the EMTPWorks program folder "Info Scripts" directory. This directory is on the script search path of EMTPWorks and that is why scripts from this directory can be named without entering a path name.

Masking is a powerful feature for data hiding and encapsulation. It provides a high-level access to subcircuit contents and allows creating user defined models.

Help on masking methods is available after selecting this option and looking into the Help tab of the default mask (see [Mask Scripts](#)).

The default mask script can be changed from the initialization file EMTP.INI (see [Drawing] section).

Masking is based on scripting. The currently available generic mask scripts are:

- `black_box.dwj`: generic mask, allows entering data in a textarea and sending it to subcircuit devices. It is the default mask. It creates a simple and quick encapsulation usable in a large variety of cases.
- `script_black_box.dwj`: allows entering mask data and rules. This is a more sophisticated script since it allows programming using JavaScript.
- `no_script_box.dwj`: this is the simplest mask; it just provides the user the ability to create a help panel for just documenting the subcircuit contents and functions.

Mask data sent to EMTP (using the Netlist) is saved into the device ModelData attribute. All subcircuit data is completely resolved before saving into the ModelData attribute.

When a device is masked, its Mask.Dev attribute changes (becomes a code number) to reflect this condition.

### 8.12.2 Unmasking

You can choose to cancel the Mask of a device which is called unmasking. This is achieved by unclicking the "Mask this subnetwork" checkbox shown in the Mask tab in "Subcircuit Properties".

Unmasking is automatically detected by the device script specified in Mask.Dev.Script. This script will clear the contents of ModelData to eliminate the mask data transmitted to the subcircuit contents. You must be careful that this action does not result into undetermined data left in the subcircuit.

If you just unmasked and decided to go back, then you can still recuperate the mask by masking again with the same previous mask script. This is because the available mask scripts also save their basic data into the FormData attribute. This is no more applicable if you unmask and select a new mask or change the mask.

It must be remembered that unmasking a subcircuit of a given type will not unmask all other subcircuits of the same type. The subcircuit contents will however remain linked and the user must create a new unique type if it is desired to unlink subcircuit contents.

## **9 Device Symbols and Libraries**

### **9.1 About Device Symbols**

Device symbols are an important resource in your design creation process. Whether you primarily use the symbols provided with EMTPWorks, or you create special libraries for your own use, the completeness and accuracy of this data has a major effect on your design flow. Library files generally outlast any one design and are used for many years across many projects. In addition, many EMTPWorks features rely on specific steps being taken while creating a symbol. For these reasons, EMTPWorks provides a variety of features for creating and editing the symbols themselves and for maintaining symbol library files.

Creating a device symbol can involve much more than just drawing the graphics that represent the device. Many EMTPWorks features rely on correct text attributes being associated with each symbol.

### **9.2 Terminology Note**

It is important to distinguish between the definition of a symbol that is stored in a library, and an instance or usage of it in a schematic. In this manual we use several different terms to refer to the definition of a device symbol in a library. If we use the term symbol, part, part type, or just type, we are referring to the definition of a device symbol in a library. The term symbol is used when we are primarily interested in the graphical representation of a device, but a symbol is always stored with related pin definition information and text attributes.

### 9.3 Working with Symbol Libraries

The symbols and related parameters for EMTPWorks devices are stored in data files called symbol libraries. 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 attributes, position, orientation and type of each pin, etc.
- A picture representing the symbol for this part type.
- A polygon outlining the symbol, used for highlighting and erasing the symbol.
- An optional internal circuit definition.

### 9.4 Designs and Libraries

Whenever select a device symbol from a library and use it in a design, all information needed to display and edit that device is retained with the design. (Of course, only one copy is kept, regardless of how many times you use the same symbol in one design.) No further access to the library itself is required. This is done to ensure that a design file is always a complete entity and that future changes to a library will not inadvertently render an old design incorrect.

However, there are frequently times when you would like to update a symbol in a library and then copy the changes into one or several designs. Conversely, you may have edited a symbol or its subcircuit on a single design and wish to save the changes back to a library for use elsewhere.

For these reasons, EMTPWorks does retain some information with each symbol about its "home" library and provides a number of features to allow transfer and updating of symbols between designs and libraries. Each time a new symbol is used in a design (i.e. one that hasn't been used in this design before) the library name, file path and "last modified" date are stored in the attribute fields LibName, LibPath and LibDate, respectively. These are used by the "Options>Part Type>Update from Lib" command to locate the original library and determine if it has changed since we used it.

In addition, whenever the definition of a device symbol is modified, a new "checksum" value is calculated. This is also used as a check to verify the equivalence of two symbols whenever one is loaded from a library. This checksum cannot be set by the user, but it can be dumped in report using the \$CHECKSUM script command (see [Export Script Language](#)). This can be used to tell if inadvertent modifications have been made to a symbol, but not to another that appears similar.

The following sections deal with the creation and maintenance of library files. Later parts of this chapter deal with editing the symbols themselves.

### 9.5 Creating a New Library

To create a new and empty symbol library file:

- Right-click anywhere in the Parts palette or use "Home>Libraries>Library Operations". In the pop-up menu that appears, select the "New Library" command.

In either case, a standard file save box will appear. Enter the desired name for the library and select a disk directory. The library is created and opened automatically so it appears in the Parts palette.

If you wish to have the library opened automatically at startup when you enter EMTPWorks in the future, see [Automatically Opening Libraries at Startup](#).

### 9.6 Manually Opening a Library

To open any library file on your disk:

- Right-click anywhere in the Parts palette or use “Home>Libraries>Library Operations”. In the pop-up menu that appears, select the "Open Library" command.

A standard file open box will appear. Locate the desired file in the usual way. The library will be opened and appear in the Parts palette. A small amount of memory is occupied by each open library file.

Libraries with the same file name, but with different locations on your disk, can be opened.

## 9.7 Automatically Opening Libraries at Startup

Libraries can be opened automatically when the program starts. See [Opening Libraries Automatically](#) for more information on this topic.

There are also initialization keywords in the initialization file (see [\[Libraries\]](#) section).

By default, when EMTPWorks is installed, a folder called Libs is created containing the initial libraries.

## 9.8 Manually Closing a Library

To close any open library file, either:

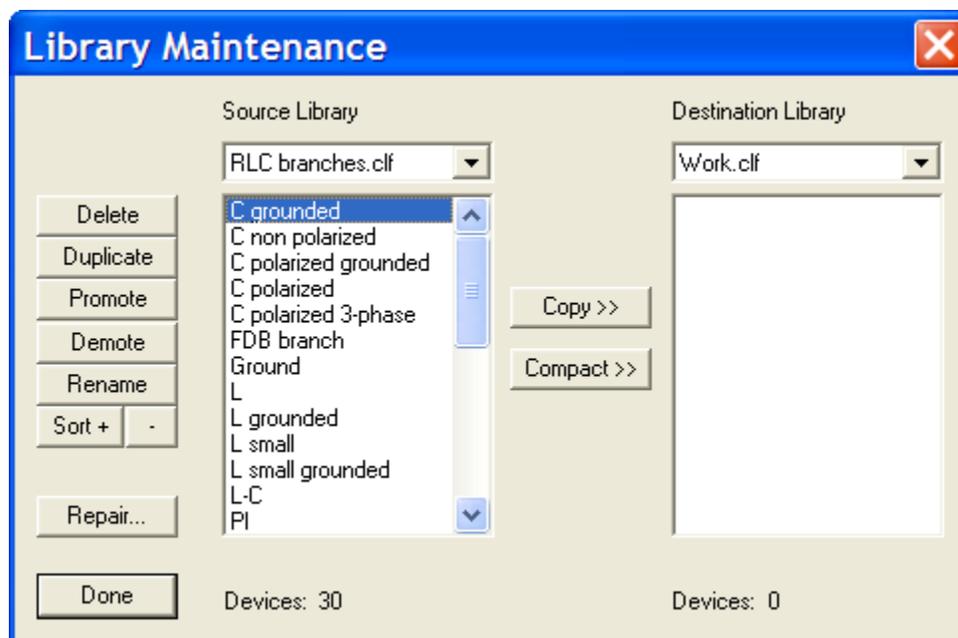
- Right-click anywhere in the Parts palette or use “Home>Libraries>Library Operations”, in the pop-up menu that appears, select the "Close Library" command.

In either case, a list of the open library files will appear. You can use the Shift and Ctrl keys to select multiple files to be closed in one operation and then press the Close button. Alternatively, you can simply double-click on the name of a single library.

Any information required for symbols used in any open designs will be automatically retained in memory. Once you have used a symbol in a design, all information required has been copied into the design's data. No further access to the library itself is required.

## 9.9 Lib Maintenance

The "Lib Maintenance" command is available from the right-click menu in the Parts palette or use “Home>Libraries>Library Operations”.



## Reference

Most of the commands appearing in this panel are self-explanatory. The Shift and Ctrl keys can be used for selecting more than one symbols in some operations.

- Delete: Deletes one or more symbols from the "Source Library". This operation cannot be undone.
- Duplicate: Duplicates symbols within the "Source Library". You will be prompted for each selected item to enter a new name. Names must be unique within a library.
- Promote: Causes the selected items in the "Source Library" to be moved up the list.
- Denote: Causes the selected items in the "Source Library" to be moved down the list.
- Rename: Used to rename a single selected symbol in the "Source Library".
- Sort: Sorts the entire list alphabetically.
- Repair: Useful for fixing library problems.
- Copy: this button is used to copy one more symbols from one library to another.
- Compact: 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. You can compact into a new library and delete the previous library after verifying the new library.

It is noticed that the items displayed in the Parts palette are always sorted alphabetically and the Promote, Denote and Sort procedures will not affect this order. These procedures change the order in which the symbols are indexed in the library file, which affects ordering in some internal operations.

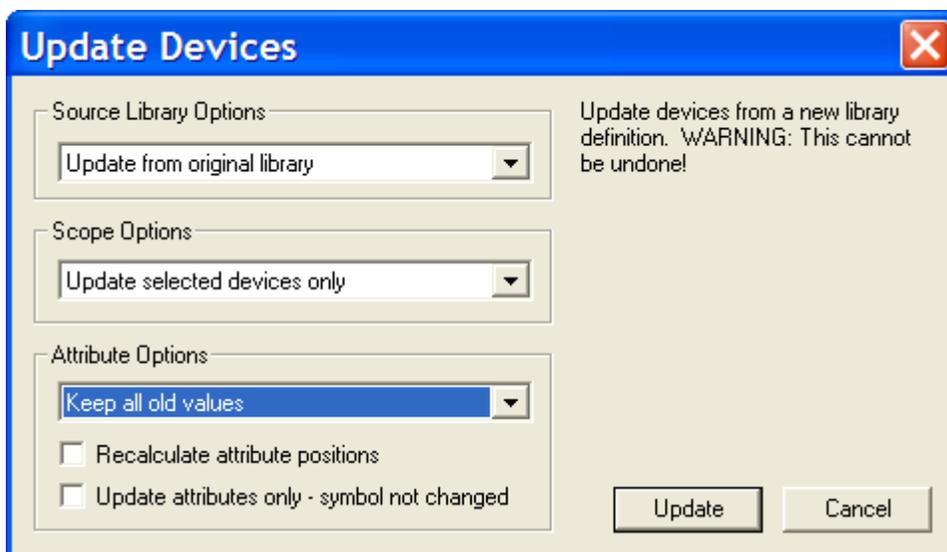
### 9.10 Update from Lib

This command is available from "Options>Part Type".

The "Update from Lib" command updates the definition of the selected device from its original source library. There are a variety of options available to determine how attributes and subcircuits are updated and how a source symbol definition is selected.

Depending upon the options selected, this command may replace the entire internal circuit of the selected device. This cannot be undone!

Selecting the "Update from Lib" command in the "Part Type" sub-menu of the "Options>Subcircuit" toolbar, causes the following box to be displayed:



The box that appears may be simpler than that shown above. Only options that are relevant to the specific device selected are shown.

The first option pop-up menu allows you to select how the source library will be located:

Update from original library	This selection will locate the original library that the device was derived from using information stored in its attribute fields. As long as the library can be located and a part with the same name can be located in the library, you will not be prompted to locate a source.
Select a new source library	This selection will cause a sequence of two prompt boxes to be displayed requesting a library and then a part within the library. This new symbol will be used to update the selected device.

Next, you need to select which devices on the schematic you wish to update. This is only when two or more devices of the same type are available in the design. EMTPWorks will detect similar devices through all hierarchy levels. The choices are:

Update selected devices only	Only the selected device will be updated. 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	This will update all devices in the design that were originally derived from the same part type. Note that this is determined by the internal logical linkage between the device instances and the part definition, not by the type name. If you have multiple type definitions with the same name in your design, this option will only update the devices

Reference

	linked to the same type as the selected one.
Update all devices with the same type name	This option will update all devices derived from a type with the same name as the selected one, even if the definitions have become separated by some previous operation. This is specifically intended to remedy the case where a design inadvertently contains multiple definitions of similar devices.

The "Attribute Options" dropdown menu allows you to choose how attribute values are updated in the event that the new symbol definition has different values than the existing device instance:

Keep all old values	This selection tells EMTPWorks to maintain all attribute values exactly as they are in the current instance, even if the new symbol definition has different values.
Keep all old values modified in the instance	This selection indicates that you wish to keep any attribute values in the device instance that was changed from the default value in the original definition. Any values that still have the default specified in the old symbol will be updated to the new default value in the new symbol.
Use all new non-null values	This item indicates that you wish to use the value specified in the new definition, if there is one, but keep any old values that are not overridden by new ones.
Use all New values-all old values will be discarded	In this case, the updated instance will have exactly the values in the new definition, even if they are empty. All attribute data present in the old definition will have been replaced by new values.

It is noted that the Name attribute is not modified by this operation. It is also recalled that all device simulation data, as seen in EMTP side, is saved into attributes, so the user must be careful with the updating process since it can affect previously entered data and change the numerical behaviour of a device. The first attribute updating option is the less dangerous.

Two other options appear in the Attributes section:

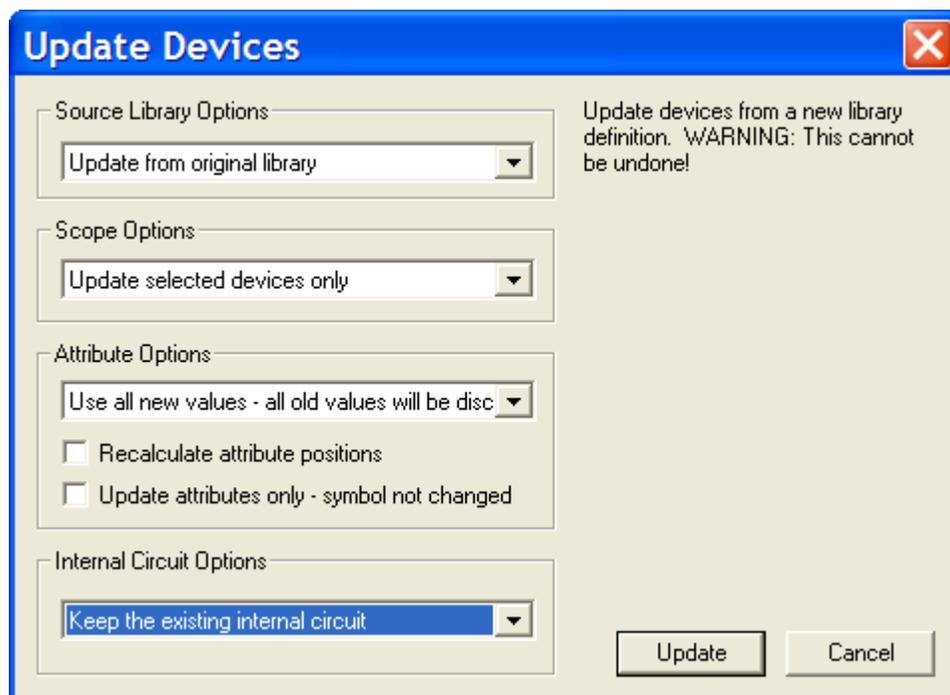
Recalculate attribute positions	This checkbox allows you to determine whether displayed attribute values will be left at their old positions or recalculated to fit with the symbol. If the symbol has not changed in shape appreciably and you have manually repositioned the displayed values, you may wish to turn off this option.
Update attributes only - symbol not changed	If this box is checked, only attribute values are affected, the device symbol, pins, subcircuit and other aspects of the part type will not be affected.

In some cases the symbol of the updated device may significantly different from the one used for its update. The distance between its pins may have changed. In such cases it can become very

difficult for EMTPWorks to readjust the signals to maintain correct signal paths and even connectivity. That is why in some cases it is preferable to check the last option.

## 9.11 Internal Circuit Options

Finally, if either the existing instance or the definition in the library, or both, have subcircuits, then you need to choose which circuit you wish to keep.



Update the internal circuit from the library

If the selected device has an internal circuit, it is removed. If the new library part definition has an internal circuit, a new one is created.

Keep the existing internal circuit

Any existing internal circuit is kept and only the symbol is updated. Note that if you make this selection, the part definition in the design will be marked as having been changed since it is not the same as the library definition.

### Extra Notes

The "Update from Lib" menu is disabled when the circuit is locked.

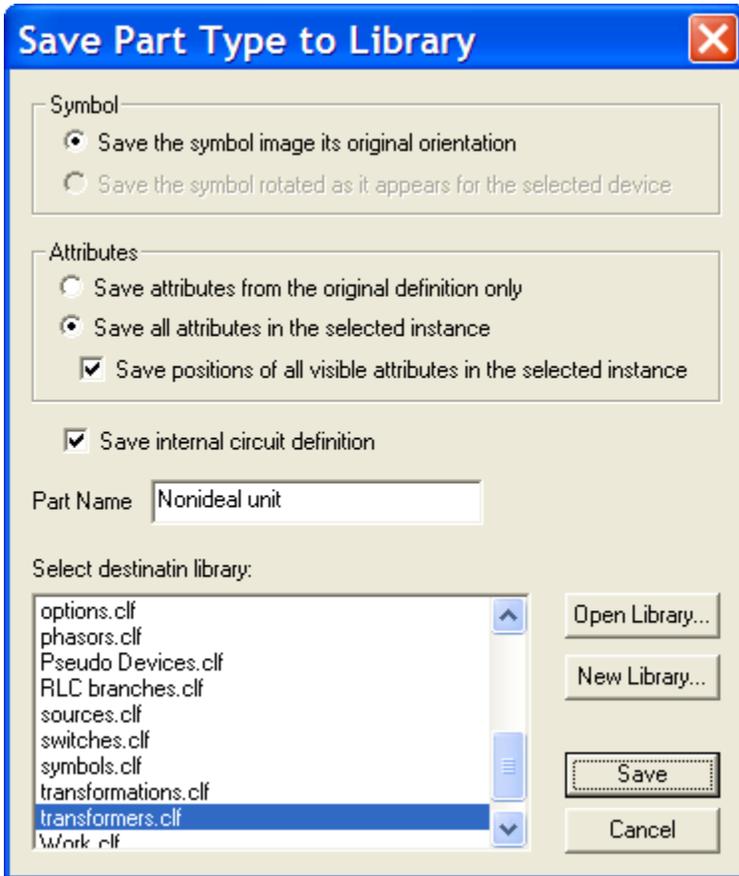
When updating devices located in subcircuits, there could be one or more copies of the parent circuit. In these cases EMTPWorks is able to locate other devices through the parent circuits. You must also remember that if the parent is not unique then changes to its contents will affect all other instances of the parent.

When two parents are identical (of the same type), then updating an internal device and selecting to update all devices of the same type, will find only the number of devices in one parent.

When two parents are not of the same type, then updating an internal device and selecting to update all devices of the same type, will find the total number of devices from both parents.

### 9.12 Save to Lib

The "Save to Lib" (from "Options>Part Type") command saves a type definition for the selected device to a library. The following box will be displayed:



This table summarizes the options available.

Save the symbol image its original orientation	Will save the device in its original orientation, as originally created.
Save the symbol rotated...	As the device appears in the design.
Save internal circuit definition	If this box is checked, the internal circuit attached to the selected device will be saved with the part definition. It is very important to remember to check this box if the device has a subcircuit, since it will be not be saved to the library otherwise.
Save all attributes from the original definition only	If this option is selected, only attributes that were originally defined for the library part will be saved.
Save all attributes in the selected	If this option is selected, all the attribute

instance	values associated with the selected device will be made part of the saved library part.
Save positions of all visible attributes in the selected instance	If this box is checked, then a ".Pt" position field will be saved for each attribute that is visible on the instance and for which the associated ".Pt" field is defined in the design's attribute table. This option allows saving the desired position of a visible attribute after adjusting it with the mouse pointer.
Part Name	The part name under which the new library entry will be saved. This is the name that appears in the library listing.
Open Library	Opens a library which is not listed.
New Library	This button will display a standard file save box allowing you to create a new, empty library.

It is noticed that built-in libraries are normally locked and you can only save new devices into your own libraries. The only built-in library that is not locked is Work.clf.

It is strongly recommended to create user libraries in a directory different from the default EMTPWorks directory Libs (see [Opening Libraries Automatically](#)).

### 9.13 Circuit to Library

To make a library containing all the symbol definitions used in a circuit:

- Ensure that the circuit whose symbols you want to save are in the front most window.
- Ensure that the library you wish to write the symbols to is selected in the Parts palette. If it is not, either select it from the library selection menu in the Parts palette, or use the "New Lib" command to create one.
- Select the "Circuit to Library" command in the "Part Type" submenu ("Options>Part Type").
- Click on the OK button.

This command will proceed to write all the unique symbol definitions in the circuit to the selected library. Here are some points to note on the operation of this tool:

- In a hierarchical design, only symbols in the current circuit level are written.
- Pseudo-devices (i.e. things like power and ground symbols and page connectors) are included in the items written out. Depending on your requirements, you may wish to delete these from the final library.
- The symbol written out will include all attribute values defined in the first instance of the symbol that is encountered, except for Name and InstName. The symbol that is written out is not necessarily identical to the definition used in creating the circuit in the first place.

One of the purposes of "Circuit to Library" is to allow you to use the attribute editing tools like the Browser to work on large numbers of symbols at once. You can use "Library to Circuit" to create a design containing one of each symbol in a library, then editing the resulting devices and write

the modified versions back out. This can be very convenient, for example, if you need to add or edit the same attribute value to all the symbols in a whole library.

### 9.14 Library to Circuit

There are a number of cases where it is useful to create a design that contains one of each symbol in a library, such as:

- To test a library to ensure that all symbols appear, print and Netlist correctly.
- To perform mass modifications on part attributes using the schematic tools such as the **Property** Browser.

To perform this operation, follow these steps:

- ❑ Create a new, empty design using any desired template.
- ❑ Open or select the desired library so that it is the current one displayed in the Parts palette.
- ❑ Select the "Library to Circuit" command in the "Part Type" submenu of the ("Options>Part Type").
- ❑ Click the OK button.

The Library to Circuit command will proceed to place symbols on the schematic in columns based on the sizes of the symbols. If it runs out of room on the page, it automatically creates a new circuit page. The naming of the devices and display of attributes are determined by the template's settings in the same way as if you had placed the parts manually.

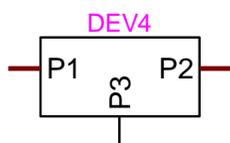
This process can require a large amount of memory. Because every device in the resulting design has a unique symbol definition, a large amount of data is loaded into memory while the design is open.

## 10 Symbol Editor

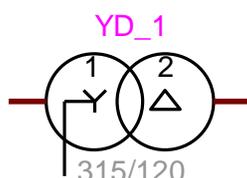
### 10.1 About the Symbol Editor

Symbols are created and edited using the device Symbol Editor tool (from device right-click menu). In addition to drawing symbols, the device symbol editor can also be used for general graphics (e.g. title blocks or simple mechanical drawings) for use on EMTPWorks schematics. It provides a complete, object-oriented drawing environment with standard drawing tools, as well as specific functions tailored for symbol creation.

One of the most useful applications of the Symbol Editor is the customization of the subcircuit symbols. When a subcircuit is created the first time, EMTPWorks uses a default rectangular symbol. You can right click on the symbol and select "Edit Symbol" to open the Symbol Editor window and modify the device drawing. A device that is originally given the default symbol



can be transformed to become



The Symbol Editor has its own window in EMTPWorks. It also has its own tab showing the Part name of the edited device.

### 10.2 Using the Clipboard in Device Symbol Editor

The standard Edit menu commands Cut, Copy and Paste can be used to move objects inside and between the symbol editor window, EMTPWorks circuit windows, and other applications. Some types of graphic objects, notably bitmaps, created by other programs are not supported by the current version of the symbol editor and will not appear if pasted into the editor's drawing area.

### 10.3 Editing existing Parts (Devices)

#### 10.3.1 Editing an existing part in a library

To edit an existing part in a library:

- Right-click on the desired item in the Parts palette, then select "Edit Part" from the pop-up menu.

In response to either of these operations, a copy of the symbol definition is loaded into a device symbol editor window. No changes to the source library will be made until you save the symbol back to its original library.

## Reference

Editing a symbol in a library does not automatically update designs that have used that symbol. You must use the [Update from Lib](#) command to perform updating actions on already placed devices.

You can decide to save an edited part under a new name using the "File>Save As" command.

### 10.3.2 Editing an existing part in the design

To edit the symbol of a device that is already placed in a schematic without modifying the library it came from, right-click on the device and select the "Edit Symbol" command. This actually saves the symbol to a temporary library, and then opens it in the symbol editor. You can then make any desired changes to the symbol and close the editor. At that point, you will be asked to confirm that you wish to replace the original symbol, and, if applicable, any others of the same type in the schematic.

This process is similar to using the "Update from Lib" command to update an existing device instance from an updated library and has many of the same options.

You can decide to save an edited part under a new name using the "File>Save As" command.

#### Important note

If you decide to move from the Symbol Editor window back to the design window without closing the Symbol Editor window or saving ("Symbol Editor>File>Save" or Ctrl+S), then the connection with the Symbol Editor will be lost and any changes in the Symbol Editor will not be reflected back to the edited device. Going back to an open Symbol Editor window after saving, will not reestablish the connection with the device drawing.

The normal practice is to always close the Symbol Editor window if it is desired to keep the changes and reopening the Symbol Editor window for subsequent changes.

## 10.4 Modifying subcircuits

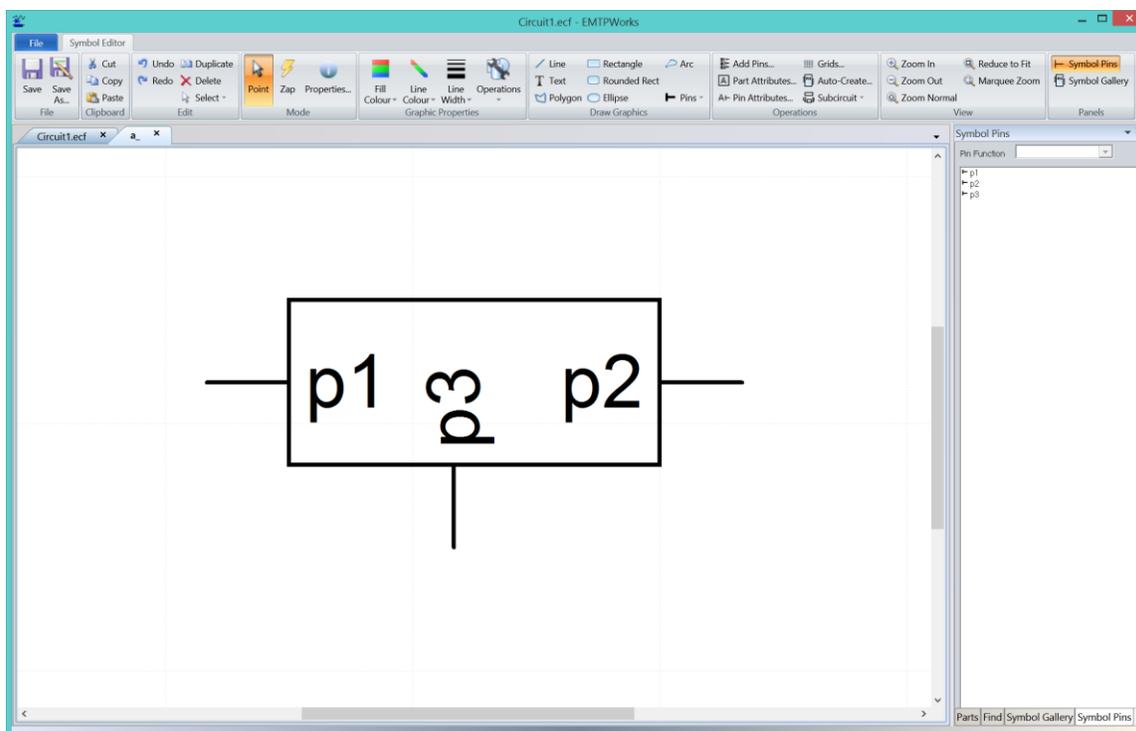
The most important techniques for manipulating existing subcircuit pins and for creating subcircuits from scratch through the "Auto Create Symbol" command (Operations toolbar), are presented in [Tutorial 4: Creating a Subcircuit Top Down](#) and [Tutorial 5: Advanced Subcircuit Topics](#).

## 10.5 Creating a new EMTP Part

In addition to redefining subcircuits, and existing parts, you can create new EMTP devices. This is available through the EMTP DLL based functions.

## 10.6 Editor functions

The editor window is shown below. It is used here to modify a subcircuit based device appearing in a circuit.



In addition to the main editor window, there are two new panels on the right: Symbol Gallery and Symbol Pins. These panels can be turned on and off using the Panels toolbar of the “Symbol Editor”.

At the top of the window, the toolbar buttons provide a quick access to various functions.

## 10.7 Symbol Gallery

### 10.7.1 Using elements from the Symbol Gallery

To add an item displayed in the “Symbol Gallery” list to the symbol editing window, simply drag and drop it at the desired location. Here are some additional notes on this operation:

- The complete item dragged from the list will be added to the current symbol as a grouped graphic item. If you then wish to edit the individual elements of the new item, you can right-click on it and select the Ungroup command.
- The placed item can be rotated to a new orientation by right-clicking on the item and choosing one of the Rotate or Flip commands.
- If the placed item contains one or more pins, these will be added to the pin list or matched with existing, unplaced items in the list. The program attempts to use the name that was assigned to the pin when it was added to the element, but will add or increment a numeric portion of the name in order to create a unique name.

### 10.7.2 Adding elements in the Symbol Gallery

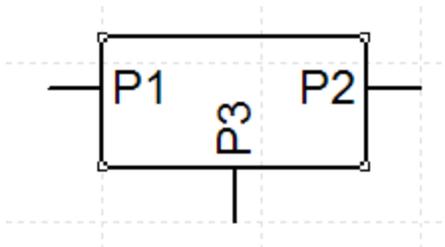
The “Symbol Gallery” is simply a normal symbol library file which is selected to have its contents displayed in the Symbol Gallery list. Thus, all the usual methods for creating, editing and updating symbols can be used on items in the library.

The Symbol Gallery file is not normally open by default, so it must be explicitly opened before you can save items to it or edit it. To open this file, right-click in the parts list and select the “Open Library” command and use the file open box to select the “Symbol Gallery” file. You must have write permissions to modify this file or you can modify a copied version and then copy back.

In a default EMTPWorks installation, this file is called "symbol\_gallery.clf" and is located inside a folder called "Data Files" inside the EMTPWorks program folder. If the EMTPWorks initialization file has been modified to specify a different file, you will have to locate that file.

## 10.8 Placing objects

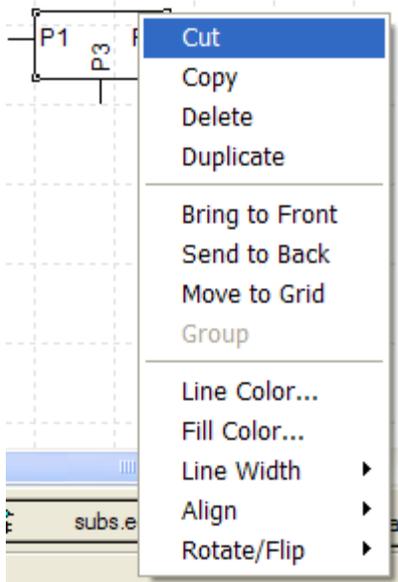
Once an object is placed, it can be clicked upon to be selected and modified. When an object is clicked, it shows selection handles as in standard drawing applications.



You can click-hold and drag the handles to resize objects. You can click anywhere on an object, hold and move the object.

It is in most cases preferable to increase the zoom level and change the Grid settings for improved precision drawing. The Grid settings can be changed from "Operations>Grids". The Snap option can be reduced to one pixel.

Each object placed on the editor pane also has a right-click menu. The right-click menu for drawing objects is shown below.



Most of the above right-click commands are self-explanatory. The "Bring to Front" and "Send to Back" commands are for ordering overlapping objects to get the desired appearance.

The line and fill color settings are explained in [Line and Fill color settings](#).

## 10.9 Symbol Date Stamping

Whenever a symbol in a library is modified, an attribute field called DateStamp.Symbol is automatically updated with the current time. Since all attributes associated with a symbol are

copied into a design when a symbol is used, this can be used to determine whether a symbol has been modified since it was used.

The date stamp is stored using the system's internal integer date format, that is, an unsigned integer representing the number of seconds since January 1, 1970. To create a more human-readable date value, the Export tool has date conversion functions available.

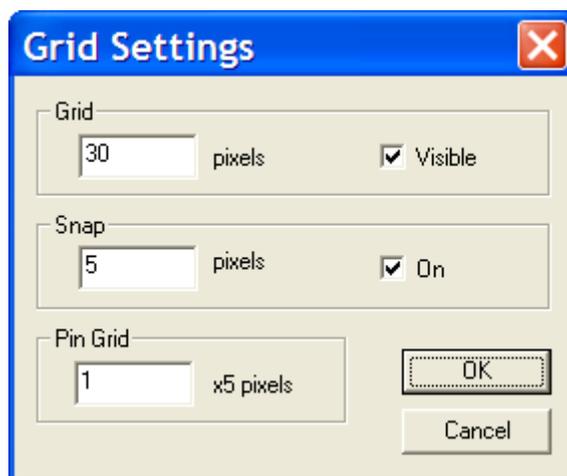
The following operations cause the DateStamp.Symbol field to be updated in the part definition:

- Saving the symbol using a Save or "Save As" command in the device symbol editor.
- Using the "Save to Lib" ("Options>Parts Type" toolbar) command to save a part from a schematic to a library with the "Use attributes in selected instance" option turned on. If this option is off, the date stamp is not updated since the part is not being modified.
- Saving parts to a library using the Circuit to Library command.

Operations that simply copy the symbol definition intact, such as the Copy button in the "Library Maintenance" box, do not change the date stamp.

## 10.10 Setting Grids

The Grids command (Operations) allows you to specify the visible grid spacing and the snap-to grids for objects drawn using the drawing tools.



**Visible (grid)** This check box determines whether visible grid lines are shown in the drawing workspace of the symbol editor window. The spacing between these grid lines is determined by the value in the "Grid" field.

**Snap on** This check box determines whether the corners of objects made with the drawing tools are moved to the nearest snap-to grid point.

**Grid (Spacing)** This number determines the spacing between the visible grid lines. The units are in screen dots at the default zoom level.

**Snap (Spacing)** 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. This

must be a multiple of 5 to meet the EMTPWorks pin grid requirements. The units are in screen dots at the default zoom level.

## 10.11 Line and Fill color settings

The line and fill color setting panels have a "Color Override" option. This option allows locking object color from user or script changes in the design window. In some cases, depending on the device data handling methods, the device color may change through scripts to reflect some device properties. If the device color settings are not overridable, then the script action will be cancelled.

## 10.12 Working with Pins

Once you have placed a pin on the symbol, you will see that an entry in the pin list has been added for each pin.

Pins are listed depending on the order in which you placed the pin symbols. You can check the association of names to pins by clicking on an item in the list and observing which pin on the drawing is highlighted.

To change pin ordering in the pin list, click and drag the pin to the desired position. Most of built-in EMTP devices require that pins appear in a certain order in the output Netlist. Unless you modify attributes or apply special scripts, this order is determined by the order in which the pins are placed on the symbol.

You should never change settings, delete pins or modify pin ordering for built-in EMTP device libraries unless the device documentation is explicitly allowing such actions.

To change the name of a pin:

- Double-click on the pin item in the list
- Enter the new name
- Press the Enter key to terminate editing

The actual names given to the pins are not critical. It is a good idea though to use meaningful names for these reasons:

- These names will be seen in the Schematic tool's (design page) pin info dialog.
- They can be extracted in Netlist output if needed.
- They are used when binding pins to ports in subcircuits to visually recognize transmitted signal functions.
- In the case of bundle internal pins, they are used when connecting bundle pins.

It is not allowed to have two pins with the same name.

Pins listed, but not attached to the device are shown in red. Normally all pins must be attached and red colored pins must be reused (attached) or deleted.

You can also right click on a pin in the pin "Pin List" panel for more options.

## 10.13 Pin Function

In addition to setting pin names, the pin list has a dropdown menu for setting a pin type (function). This is in relation to the type of signal (see [Device pin types](#)).

## 10.14 Deleting Pins

In a device symbol editor window, pins exist both in the graphic representation of the symbol and in the pin list. They have to be deleted from both places before they are completely removed from the part definition.

To remove a graphical pin from the symbol, simply select it by clicking on it and delete it with the Delete key in the usual way. Note that this does not remove the corresponding entry from the pin list and from the device. If you wish to replace the pin, for example to use a different orientation, you can simply select the item in the list and place the appropriate pin graphic from the palette.

To remove the selected pins in the pin list and their associated graphic pins:

- ❑ Select the pin or pins in the pin list that you wish to delete. You can select more pins by holding the Ctrl or Shift keys while clicking in the list.
- ❑ Hit the Delete key on the keyboard. It will display a confirmation box before proceeding to delete the pin entries.

## 10.15 Adding sequential pin names

The basic pin placement procedure requires that the pin name be entered manually for each pin, which can be rather tedious for large numbers of sequentially numbered pins. To solve this problem, the "Add Pins" ("Operations" toolbar) command allows you to enter a sequence of names in a format similar to that used for bundle breakout creation and for the "Auto Create Symbol" command. These pins are added to the pin list, but not placed on the symbol. After the pins have been added to the pin list, you can place the pins sequentially on the graphic with no further typing required.

You can also use the "Auto Create Symbol" command to specify the pin names and auto-generate a rectangular symbol in one operation.

Pin names may be entered into the "Add Pins" box. The pins are created and merged with the contents of the pin list when the Enter key is pressed or when the Add button is clicked. The "Add Pins" command does not create graphic pins, it only adds pins to the pin list.

The created pins are merged with the names in the pin 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" box.

Here are the rules for how a list of pin names is entered:

- Pin names may be up to 16 characters long.
- Normal pin names (that is, not bundle pins or bundle internal pins) 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. The sequence A0..3 defines pins A0, A1 and A3. It is allowed to enter several sequences by separated sequences with a space.
- Bundle pin names are denoted by a name followed by [...]. Bundle pins do not need to have internal pin names defined and may not have pin numbers. Bundle 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 bundle name pin by not placing a bundle name in front of the []. For example, [A B C] adds the internal pin names "A", "B", and "C" to the pin list.
- The pin function (input, output, etc.) can be specified by placing a "|" character followed by a letter denoting the type. All following pins will have the specified type until another "|" specification is found. The allowable pin type characters are:

I        Input

## Reference

O	Output
P	Power

For example:

|I CLK,,D0..7 |O Q0..7

### 10.16 Placing a Bundle Pin

Bundle pins allow bundles to be connected to the part. A bundle pin's functionality is determined by the internal pins it contains. These can be specified when the symbol is created, or modified later on the schematic using the "Bundle Pin Info" command described in [Changing Bundle Pin Connections: Bundle Pin Info](#). It is recalled that it is not allowed to modify built-in device bundle pins, since such actions will result in data corruption.

To add a bundle pin to a symbol you need to use the "Add Pins" command to add the bundle pin to the pin list and specify its bundle internal pins.

In this example:

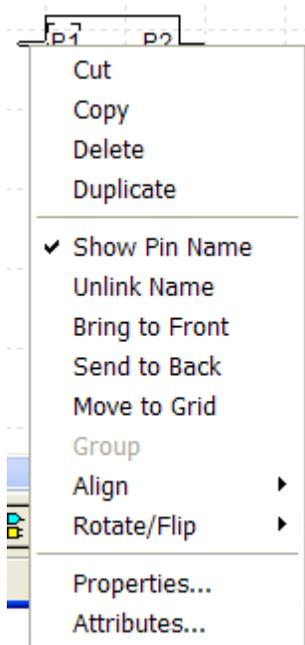
BUN[Trig Reset]

the bundle pin BUN is added with internal pins Trig and Reset.

The bundle pin can be placed first and then you can right click on the pin in the "Pin List" panel and add pins. Bundle pins are marked in the "Pin List" with a bolder pin icon and will be followed by their internal pins.

### 10.17 Pin right-click menu

The pin right-click menu is used to access to several functions for setting pin data and standard editing tools.



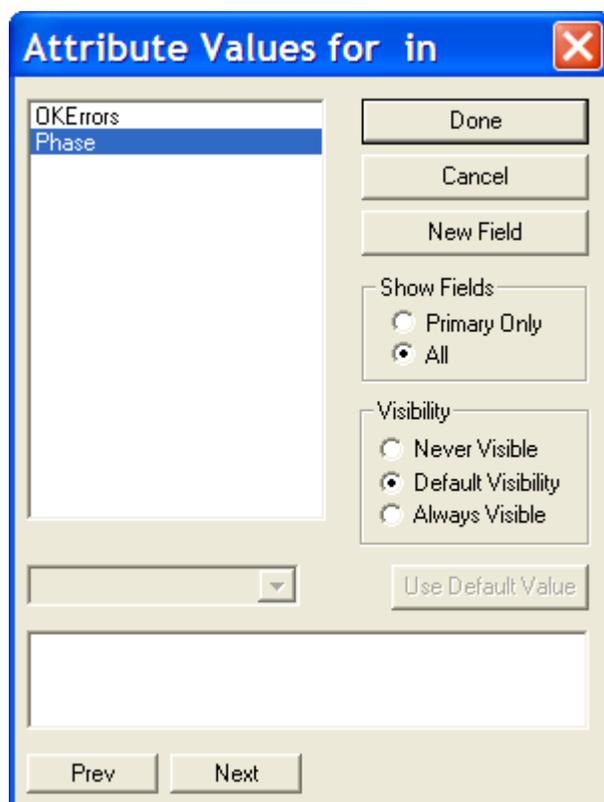
The "Show Pin Name" command shows the pin name near the pin. The "Unlink Name" command allows detaching the pin name from the pin and moving it to another location.

The Properties command is similar to double-clicking on a pin.

The Attributes command is explained in [Setting Pin Attributes](#).

## 10.18 Setting Pin Attributes

The Attributes command from the pin right-click menu or "Options>Pin Attributes" opens the panel:



This is similar to entering pin attributes in the design drawing. It allows setting and saving attributes with the device. Attributes defined while in the device symbol editor will be associated with the part in the library if the part is edited from the library. These will become the default attributes for the part when it is placed in a circuit.

When the Phase attribute is not set, it means that the connected signal does not have a forced number of phases (see [Signal Line Type](#)).

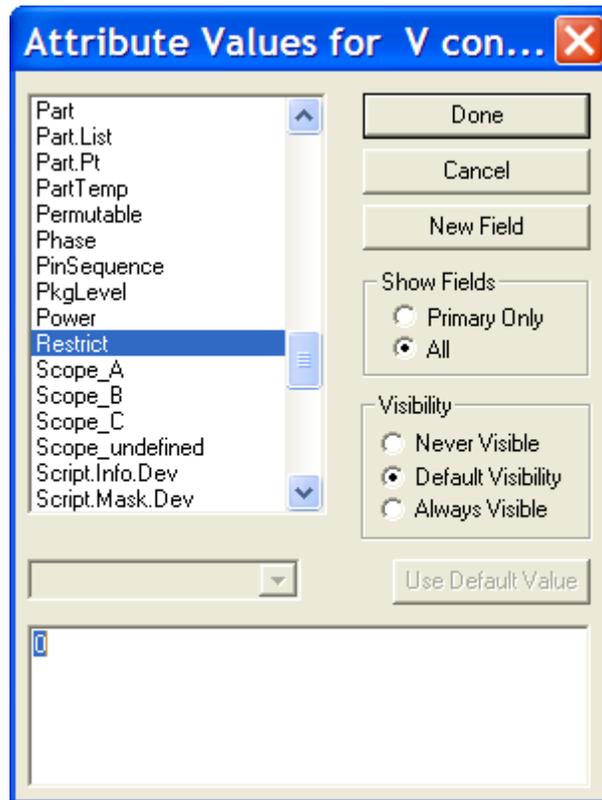
- If it is required to force a 1-phase (general) signal, then this attribute must be given the value "=1"
- If it is required to force a 3-phase signal, then this attribute must be given the value "=3"

The 1-phase setting forces a 1-phase pin and the 3-phase setting forces a 3-phase pin. The control input and output pins can be only 1-phase (general signal).

## 10.19 Setting Part Attributes

To edit the attributes associated with a part select the "Part Attributes" command in the Operations toolbar. To edit the attributes associated with a pin, select a pin in the pin list and then select the Pin Attributes command from the Operations toolbar. The only visible difference

between the part attribute dialog and the pin attribute dialog is the addition of "Next" and "Previous" buttons at the bottom of the window.



Attributes defined while in the device symbol editor will be associated with the part in the library if the part is edited from the library. These will become the default attributes for the part when it is placed in a circuit.

Only attribute fields that have a value are stored with the symbol. When you save the symbol to a library, any fields with null values are stripped off to save storage. This has the side effect that you cannot create a "place holder" field definition in a symbol for future use without putting a value in it.

## 10.20 Attribute Field Definitions in Symbols and Designs

In EMTPWorks, symbol definitions are completely independent of any particular design or library file. For this reason, symbols carry around their own definitions of the attribute fields that are associated with them. By default, when you create a new symbol, you see the attribute definitions for the design that is currently open, or a default set if no design is open. However, the "New Field" button in the "Part Attributes" box allows you to add new field definitions that affect only this one part definition. This operation does not automatically add the same field to the current design.

When you place a symbol in a design for the first time, the program compares the attribute fields defined in the symbol to those defined in the design's symbol table. If there are any new fields in the symbol, you will be asked if you wish to add them to the design's table. Your choices in this case are to either:

Click the OK button and allow the program to automatically add the field definitions to the design's attribute table. The program will attempt to make reasonable settings for the maximum length, default visibility and other settings, based on the current value, or,

You can click Cancel to stop the device placement and add the fields manually to the design's table using the "Define Attribute Fields" command. In general, this is preferred, since it gives you more control over the option settings for the field.

Some visibility options appear in this version of the attributes box that are different from what you will see on a device placed in a schematic:

- |                    |   |
|--------------------|---|
| Always Visible     | This setting indicates that the selected field should always be made visible on the schematic when the device is placed, regardless of the visibility setting for this field in the target design. This setting does not prevent the field from being removed after the device is placed, it just sets the initial state. |
| Default Visibility | This setting indicates that we wish to make the field visible only if it is defined as Visible by Default in the design in which it is placed.  |
| Never Visible      | This indicates that the field value should not be displayed when the device is placed, regardless of the design's Visible by Default setting for this field. This does not prevent the value from being displayed later, it just sets the initial state.  |

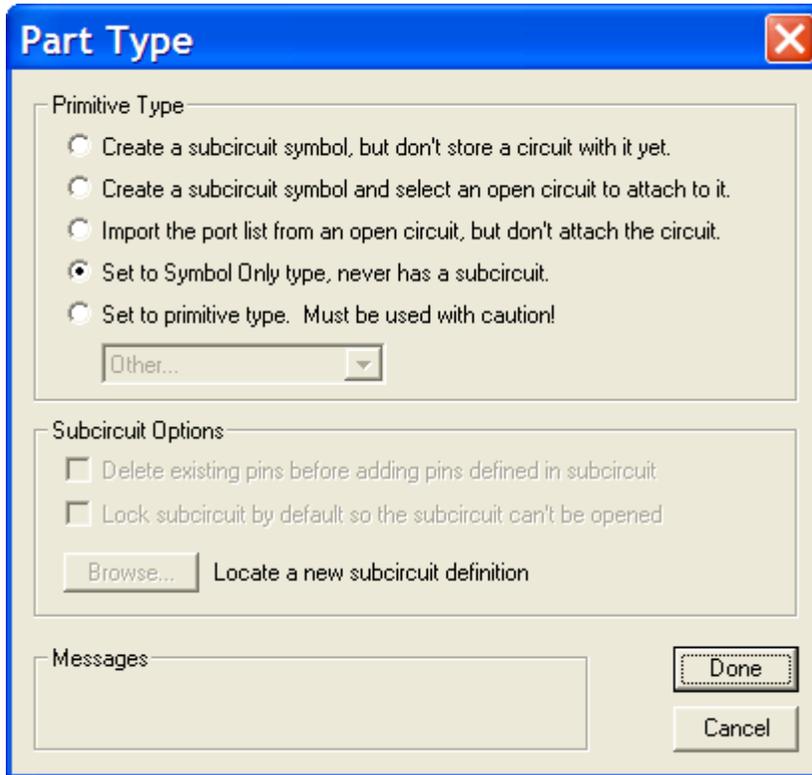
You can set the default visibility independently for each attribute field defined in a design. This is done by selecting the "Define Attribute Fields" command ("Options>Design") menu and setting the "Visible by default" option as desired for each field.

For more information on how to create and modify attributes see [Entering and Editing Attribute Data - Basic Procedure](#).

## 10.21 Auto Create Symbol

This command is available from the Operations toolbar. It is explained in [Tutorial 4: Creating a Subcircuit Top Down](#).

## 10.22 Subcircuit and Part Type command



This dialog allows you to select among several different options:

Create a subcircuit symbol, but don't store a circuit with it yet	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 later by the schematic capture section of EMTPWorks.
Import the port list from an open circuit, but don't attached the circuit.	This is similar to the No Subcircuit option. No subcircuit becomes associated with the part, but a port interface is extracted from a circuit for use with the part. Selecting this option prompts you to pick an open schematic circuit. 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 length of the editing session.
Create a subcircuit symbol and select an open circuit to attach to it.	This option prompts for an open circuit to associate with the part being created as its subcircuit. The circuit definition is saved with the part in the library. This operation also imports the circuit's port interface so that the names of the ports appear in the list and you can place the corresponding graphical pins on the symbol.
Set to Symbol Only	This option is like the No Subcircuit option except it

type, never has a subcircuit.	doesn't allow the schematic capture part of EMTPWorks to associate a subcircuit with the part in the future.
Set to primitive type. Must be used with caution!	This option is used to create special EMTPWorks part types such as pseudo devices (power, ground and port connectors). The primitive type options must only be used with a clear understanding of the effect they will have on program operation.

There are some subcircuit options in the lower part of the dialog:

Delete existing pins before adding pins defined in subcircuit	This option is only enabled when the selection made above results in a port interface being imported. If this box is checked, when the "Done" button is pressed all of the old pins in the pin list will be deleted. This allows a new port interface to be brought in without any conflicts with the existing pin list. 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 list remain exactly the same.
Lock subcircuit by default so the subcircuit can't be opened	This option has the effect of saying, "Yes. There is a subcircuit, but in general you don't want to go into it". This causes the schematic capture part of EMTPWorks to prompt to make sure it is really OK to enter the subcircuit before doing so.
Locate a new subcircuit definition	This button allows you to replace the subcircuit in a symbol that already has a circuit associated with it.

## 10.23 Opening the Subcircuit Associated with a Symbol

If the symbol currently being edited has a subcircuit already stored with it in the library, you can use the "Open Subcircuit" command in "Operations>Subcircuit" to open it for editing. This opens the subcircuit in a design window as if it was an independent design.

Modifying and saving the design that was opened with this command DOES NOT automatically update the symbol or the library it was read from. If you wish to update the symbol, you must use the Subcircuit and Part Type command and use the "Create a subcircuit symbol and select an open circuit to attach to it" option to reattach the modified circuit to the symbol.

## 10.24 Creating Pseudo-Device Symbols

### 10.24.1 Creating Pseudo-Device Symbols

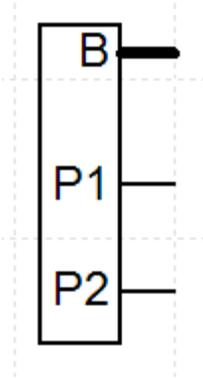
In addition to symbols that are used to represent actual circuit components, EMTPWorks also uses "pseudo-device" symbols to describe various types of signal connections on the schematic. The Symbol Editor also allows you to create customized versions of these symbols for use in your drawings.

### 10.24.2 Breakout primitive type

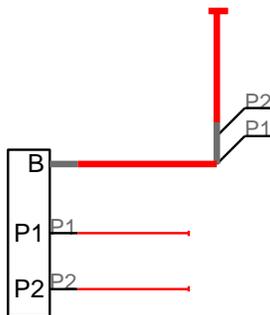
To create a breakout type you need to set the primitive type to Breakout in the "Subcircuit Part Type" dialog box described in [Subcircuit and Part Type command](#).

A breakout is a special device that allows signals to be associated with a bundle. It consists of 1 Bundle 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 bundle. Design breakouts are normally created using the New Breakout command described in [Using Bundle Breakouts: New Breakout](#).

This following device is created by placing the breakout pin B and adding the normal pins P1 and P2.



it results into the following device in the design drawing. The pins P1 and P2 are separately accessible.



Another way to split signals out of a bundle 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 bundles or any combination of bundles and signals.

### 10.24.3 Creating a Signal Connector

To create a signal connector type you need to set the primitive type to "Signal Connector" in the "Subcircuit Part Type" dialog box described in [Subcircuit and Part Type command](#).

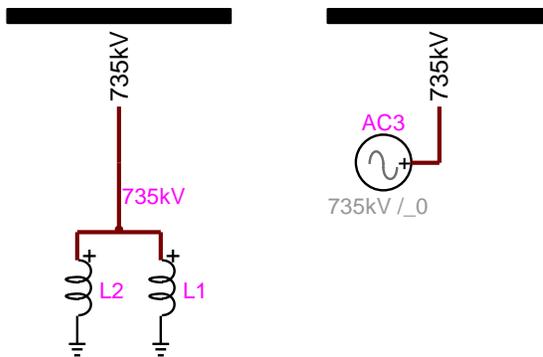
A signal connector is a special type of device which is generally used to represent a power or ground source signal. It can be used as a general connector for simplifying the schematic.

The pin name becomes the signal name and it cannot be changed. This method is used for creating the Ground device which has a pin named GND.

In this example, the signal connector is used for creating a common 735kV bus:



The pin is given the name 735kV and set to be a Power pin. When the device is placed in the design, it is used for connecting to devices with the signal name 735kV:



All devices are virtually connected through the signal name 735kV.

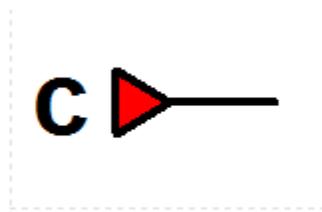
It is noticed that the Signal Connector pin type does not really count since it inherits the pin type from the connected signal.

#### 10.24.4 Creating a Port Connector

Port connectors are used for creating subcircuits. Port connectors have the property of associating a signal or bundle 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 bundles are matched by pin name. The "pseudo devices.clf" library has port connectors for control pins, power pins and bundle pins. The best way to place port connectors for creating a subcircuit is to use the "Options>Subcircuit>New Port connector" dialog.

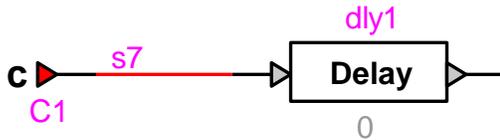
If you want to create your own port connectors (for customization needs) then you must create the port connector drawing in the Symbol Editor and select "Port Connector" for the primitive type in the "Subcircuit Part Type" dialog box described in [Subcircuit and Part Type command](#).

In the example below, a special (customized) port connector is used for input pins.

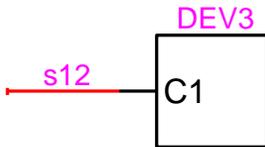


It is used here to create a subcircuit with a control device:

## Reference



The port connector name is automatically given the name of the port connector device based on its Name.Prefix attribute. The corresponding design subcircuit (created with "Options>Subcircuit>Create Subcircuit Block") symbol is given the pin name C1:

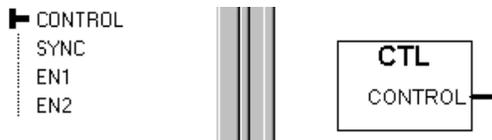


The port connector pin type is set to Output since it is connected to input pins in the subcircuit.

A port connector for a bundle must have a bundle pin which contains pins of the correct type. For example, a bundle pin with three internal pins A (input), B (input), and C (output) would be correct to connect to a parent part's bundle pin which contained pins A (output), B (output), C (input). The name of the port connector bundle pin is not important, but the internal pins must have the same names as the corresponding internal pins in the parent pin. Once the port connector is placed in a circuit its name is important, it must be the same as the parent's pin name.

The easiest way to use bundle port connectors for creating subcircuits is to use the "Options>Subcircuit>New Port connector" dialog. If it is still required to customize such a device, then it can be created in the Symbol Editor as an ordinary pin port connector, only now you must set its internal pins. A bundle port connector is also available in the "pseudo devices.clf" library.

For this example, assume that the following simple device has a bundle pin called CONTROL containing internal pins SYNC, EN1 and EN2.



To create the corresponding Bundle Port Connector using the symbol editor, follow these steps:

- Select the New command in the File menu and choose the Device Symbol document type.
- From the "Operations" toolbar select "Add Pins..."
- Enter the following string

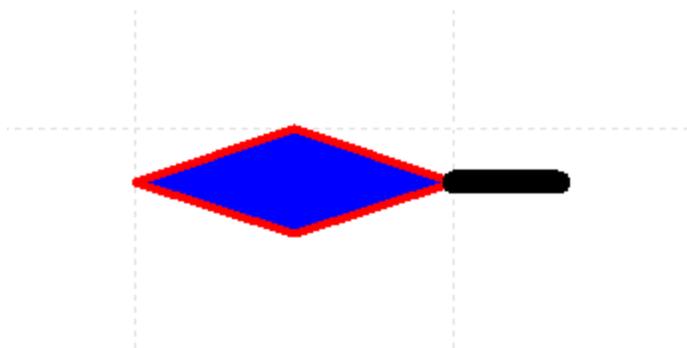
```
PORT[SYNC EN1 EN2]
```

You should now see a pin list like the following:

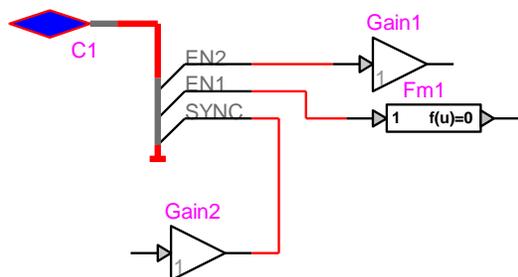


- ❑ For each bundle internal pin set the *pin function* as appropriate, i.e. the opposite of the function of the parent pin. We will assume that the SYNC signal is an input to the block and that EN1 and EN2 are outputs:
- ❑ Select the first internal pin name from the symbol editor's pin list. In this example it is SYNC.
- ❑ Use the pin function pop-up menu at the top of the list to select Output for the pin function.
- ❑ Click on the next pin in the list.
- ❑ For the remaining pins EN1 and EN2 the value Input is correct since the parent pin is an output. Check that the remaining pins are correct. You can also use power pins.
- ❑ Create a symbol for the connector, perhaps a simple rectangle.
- ❑ Select the bundle pin, PORT in the pin name list.
- ❑ Place a bundle pin from the device symbol editor's tool palette.
- ❑ Set the primitive type to "Port Connector" in "Subcircuit Part Type" dialog box described in [Subcircuit and Part Type command](#).
- ❑ From the File menu choose "Save As..."
- ❑ Choose or create a working library to save the part.
- ❑ Close the device symbol editor window.
- ❑ Place your new Bundle Port Connector in your internal circuit.

The bundle port connector in the Symbol Editor is given by:



and can be used to create a subcircuit:



The pin name of the bundle pin itself (in this case "PORT") is not important. The association between the "Bundle Port Connector" and the parent bundle pin is made by the name applied to the "Bundle Port Connector" symbol itself. The same Bundle Port Connector symbol can be used for any bundle with the same internal signal names.

## Reference

The bundle internal pins do not have to be the same type. You can include any combination of names and functions in one bundle pin.

### 10.24.5 Creating a Page Connector

Page connectors are discussed in [Inter-page Connections](#) and three types (bundle, output and power) of page connectors are available in the "Pseudo Devices.clf" library.

A customized page connector can be also created using the Symbol Editor. It is needed to set the primitive type to "Page Connector" in "Subcircuit Part Type" dialog box described in [Subcircuit and Part Type command](#).

For a normal pin, the creation is straightforward. It is noticed that the pin type does not really count since it inherits the pin type from the connected signal.

You can also use a bundle pin (with no internal pins) a page connector. It allows showing the virtual connection between pages through bundles with the same name.

### 10.25 Password protection

The Symbol Editor allows using password protection for devices. This is applicable to ordinary devices and subcircuit devices. To apply password the user must select the menu "Operations>Subcircuit>Subcircuit Password Protection":

The screenshot shows a dialog box titled "Subcircuit Protection". It has a blue header bar with the title and a red "X" button. The main area is divided into two sections. The first section, "Access Password", contains the text: "Specify this password to control access to this part when it is read from a library. If the modification password is specified, this password allows usage, but not modification." To the right of this text are two input fields labeled "Enter password" and "Re-enter password to confirm". The second section, "Modification Password", contains the text: "Specify this password to control modification of the part. If this is not specified, the access password allows modification as well." To the right of this text are two input fields labeled "Enter password" and "Re-enter password to confirm". At the bottom of the dialog are "OK" and "Cancel" buttons.

There are two options:

- Access Password
- Modification Password

When a device is given password protection then it becomes necessary to enter the password when accessing the device from a library, accessing its contents or trying to edit its symbol. When the user enters the password once on a given computer then EMTPWorks remembers the unlocking of the device for the following sessions.

To remove password protection from a device it is necessary to access the above panel and remove (Delete) all password characters manually.

# 11 Templates

## 11.1 Design Templates

A design template is intended to be used as the source for a "File>New>Design" command. When a design template is selected in the "New Design" box, the design template file is opened and all the circuit data and settings it contains become the starting point for a new design.

The standard (default) EMTP design template is "EMTP Default Design.ecf" and it is located in the default directory "EMTPWorks>Templates".

The design template can be also used to program sheet settings.

## 11.2 Creating Design Templates

Do not modify the design templates provided with the software without copying them first! You may wish to return to them later if your application changes. You should base your new design templates on existing design templates.

In EMTPWorks, a design template is simply a normal design file that has its sheet size, attribute fields, hierarchy mode and other settings pre-defined for the application at hand. The simplest way to create a template may be to take an existing design that is set up the way you like it, delete all the circuit elements and extra pages out of it and save it in the appropriate template directory.

When you create a new design using one of the template files listed, EMTPWorks just reads the file in the normal way, then renames it "Circuit1" (or the next available number), and disassociates it from the original file so that it cannot accidentally be saved on top of the template. In all other respects, "New Design" is the same as doing an Open on the template design. All the settings and contents of the design template file become part of the new design.

Generally speaking a design template is a design with predefined attributes for the design, devices, signals and pins. It also has other settings, such as the sheet settings, and background and/or border objects. Specific scripting methods are named in attributes and provided for the targeted tasks.

It is important to specify the design attribute DesignType which is used for naming the template and can be used in scripting techniques.

In the default EMTP template the DesignType is EMTP. The template has a specific set of attributes designed for EMTP simulations.

The name of template file is any valid name. Once the template is created and saved, the user can access it and create designs based on the provided automation.

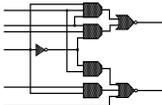
### 11.2.1 Creating a Title Block

In addition to setting attributes, a more basic and typical usage of templates is to create a title block. The title block is saved in the template and automatically appears in all designs based on the template.

EMTPWorks 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 simply using the text feature by clicking the pencil cursor (Text Tool) in the desired location on the diagram. After the text has been typed, go back to the standard pointer mode and click on the text block to select it, then use Ctrl+I to open the Text Object Options box for setting various text options.

## Reference

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

Project Title:
Drawn by:
Orig. Date:
Last Mod. Date:
Page Number:                      of:
Page Title:
Rev. Level:
Mods:
 Capilano Computing

Attributes can be used for automatically updating the various stamp fields. See [Attribute Variables](#) for learning how to show design attributes in text.

### 11.3 Sheet Templates

A sheet template is intended for use with the "Options>Design>Import Sheet Info" command and is used to set only the sheet border, title block, print setup and other settings related to the printable sheet.

There is no reason why a design template cannot be set up with all the sheet settings you need for your design. This eliminates the need to ever refer to a sheet template. However, sheet templates can be used to apply different border styles and settings to different parts of a single design. In addition, separating out the sheet functions can reduce the number of design templates you need to have available. The design template can be used to select the application type, attribute settings, script setup, etc. A sheet template can then be applied to customize the sheet border as a separate operation.

#### 11.3.1 Setting Sheet Sizes and Borders

EMTPWorks does not have any built-in knowledge of sheet sizes. Sheet border information can either be customized for each design or imported from an existing design file or template. This section describes the methods used for setting the sheet size and border for a schematic and the use of various options for adjusting the schematic border to the available output device.

EMTPWorks 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. Various settings allow you to:

- Have the schematic border adjust automatically to the current printer page setup.
- Have the schematic border retain a fixed logical size and have the printer scaling adjust automatically to fit the schematic on a single sheet.
- Have the schematic border retain a fixed physical size and be broken into the required number of printed sheets for output.

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.

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

- Importing an existing layout using the "Import Sheet Info" command from the "Options>DesignD". This is the simplest method of setting the sheet size for a new design or template, but it assumes you have an existing design that has the desired settings.
- Creating a custom layout using the "Drawing>Sheet Size Wizard" 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.

### 11.3.2 Importing Sheet Settings from another Design, Page or Template

If you have another page in the current circuit, or an existing design or template file that has the sheet size and border settings that you want, you can import those settings into the current design using the "Import Sheet Info" command. This command performs this sequence of operations:

- Deletes all text and picture objects marked as "Border" items from the current page or design.
- Copies all text and picture objects marked as "Border" items in the source design or page into the current design or page. The Border switch must be set individually for each border text or picture item using the Properties command from the right-click menu.
- Copies the sheet and border settings (sheet width, grid settings, border text settings, etc.) from the source design or page to the current design or page.
- Optionally copies the printer page setup from the source design to the current design.

Note these important points regarding this operation:

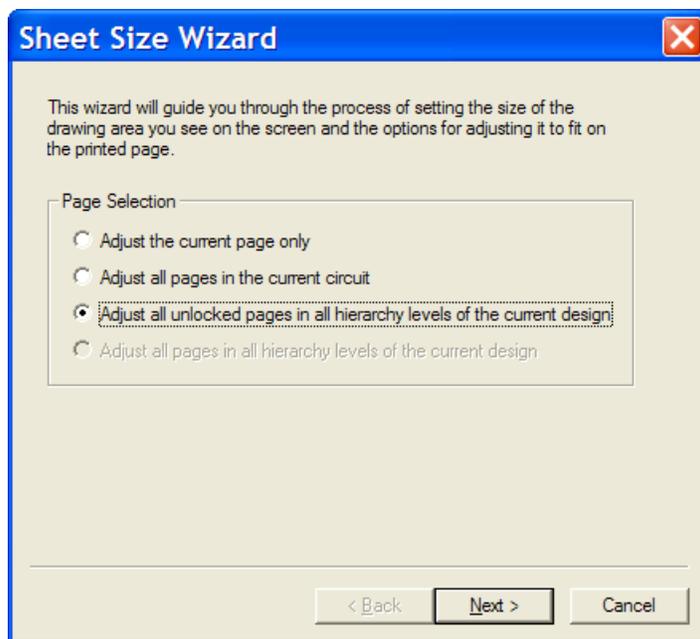
- If the source for the import is a design file, only the items and settings on page 1 of the source design are imported. There is no provision for importing different settings from multiple pages.
- The import operation does not affect any circuit items or attribute settings in the destination design.

### 11.3.3 Custom Sheet Size using Sheet Size Wizard

The "Sheet Size Wizard" command in the "Options>Design" toolbar gives you full control over the size of the drawing area that appears on the screen and how it is printed on the available output device. This section will describe the various options that are available.

#### Selecting Which Pages to Modify

If your design consists of only a single page and no hierarchy levels, the Wizard will immediately display the first page of sheet settings and you can skip this section. If your design contains multiple pages or circuit blocks, you will first see this panel:



Settings can be applied to individual pages or to the entire design.

- If the first panel of the Wizard displays the Page Selection choice, then select whether you want to change only the current page or all pages, then click Next. If the selected range contains a variety of settings, you will be warned that they will be overridden and you will have to click Next again.

### Selecting Fixed or Auto-Scaled Border

EMTPWorks provides two general types of sheet size settings:

- Scale border to match printer setup— Selecting this option causes the current printer setup to be used to determine the sheet size. Any changes in print setup (using the Print Setup command in the File menu) will immediately affect the sheet border. This is the recommended setting for most cases as it provides a guaranteed match between the drawing area and your output device. The only disadvantage of using this setting is that the sheet size will change if you change printers. Most printers have slightly different printable areas, even on the same size paper.
  - Fixed border size— Selecting this option causes fixed values to be used for the border size and position (as it appears on the screen), regardless of printer page setup. The advantage of this setting is that the sheet configuration will remain fixed regardless of changing output devices. The disadvantages are that the border size may not fit exactly to the aspect ratio of the printable area on your printer.
- Choose the desired border scaling method and click Next.

### Working with Scaled border Size

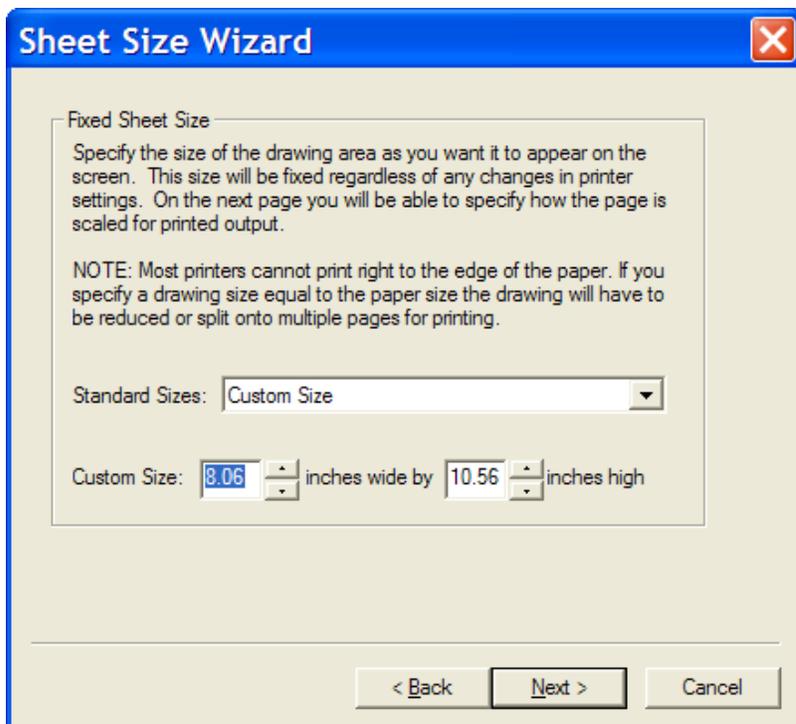
Once you have elected to scale the border to match your printer settings, you will be presented with these options:

- Single printed page— When this option is selected, the drawing border on the screen will always match your current printer page setup. Any change in printer setup will automatically be reflected in the sheet you see on the screen. This is the simplest setting and recommended for most case.

- Auto-expand— This setting also adjusts to match the printer page setup, but automatically expands to use multiple pages if you draw outside the current border. This will allow you to assemble larger drawings by piecing together a number of smaller sheets.
- Advanced— The Advanced setting is similar to Auto-expand, but allows you to specify exactly how many sheets high and wide you want the printed out put to appear on, plus it allows an additional scale factor to be applied to shrink or expand the output.

### Working with Fixed border size

When you select the Fixed border size option and click Next, you will be presented with the sheet size controls:



Using these controls, you can either select one of the standard sheet sizes listed, or enter any desired custom size.

These settings only determine the size that the sheet will appear on the screen and do not select a corresponding paper size for your printer. Printer settings must be adjusted appropriately using the "Print Setup" command from the File menu.

The standard sizes that are listed here are determined by settings in the initialization file emtp.ini. These settings can be modified to suit your requirements.

- ❑ Enter the desired sheet size and click Next.

The next step in the Wizard allows you to choose how the page will be adjusted for printing. Since you have specified that you wish the drawing to appear a fixed size on the screen, the program needs to know what to do if the visible page does not exactly match the printable area available on the paper. You have two general options:

- Scale to fit printed page: In this case, EMTPWorks will scale the drawing to fit into the available printed area for the selected printer. You can use the controls to specify that you want the drawing to be stretched onto multiple pieces of paper.

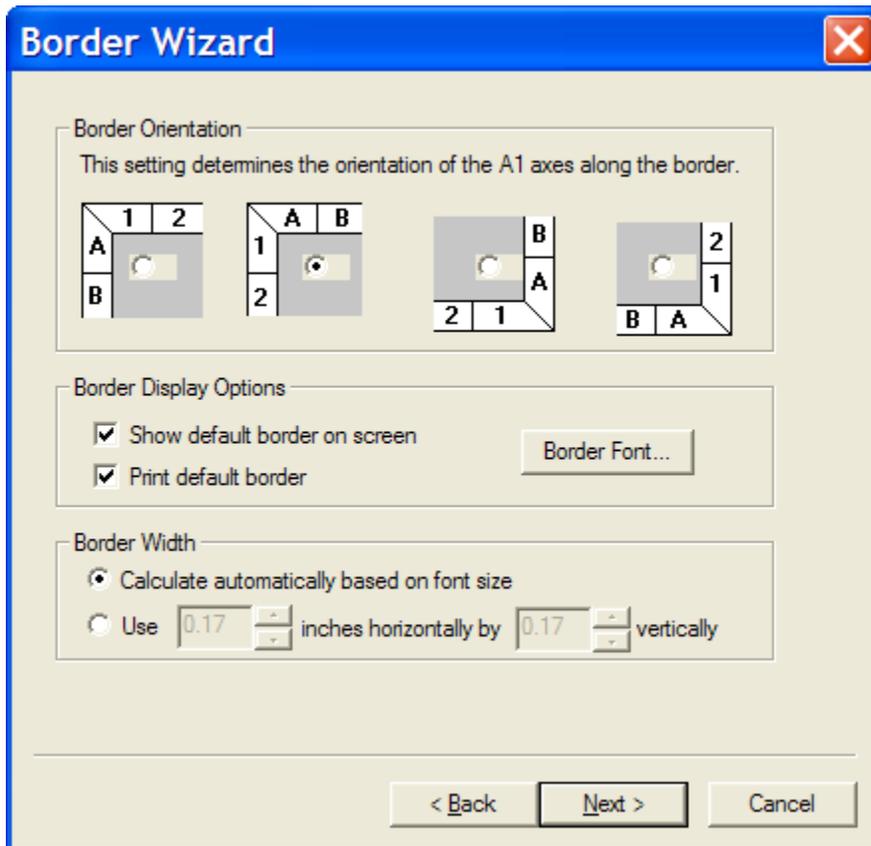
## Reference

- Fixed scale factor: With this setting active, the program will scale the drawing by a fixed amount (default 100%) for printing. This guarantees that the drawing will be a known size on paper, but may require that it be broken into multiple printed pages, depending on the settings.

### 11.3.4 Border Wizard

The Border Wizard command in the “Options>Design” toolbar allows you to set these parameters:

- The size and visibility of the default drawing border.
- The layout of letter and numbers in the reference grid that is used to locate items on a sheet.
- The size and visibility of the background grid.



This table summarizes the first page of Border Wizard controls and options:

Border Orientation	These switches determine the usage of letters and numbers for sheet grid references. This setting will affect the position displayed on Page Connectors, in the Pin List box and in report output.
Show default border on screen	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.
Border Font	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.
Border Width	These switches allow you to specify an exact width for the border or have it calculated automatically based on the text size setting.

When you click Next, you will be presented options for the layout of the major grid lines that are drawn in the background. The major grid lines are the darker lines drawn in the background. You can select later how many minor (lighter) grid lines are drawn in between the major lines.

Even if you choose not to display the drawing grid, these settings affect how the letter/number reference grid is calculated. This information is used and displayed by a number of reports and commands.

Specify the number of grid sections	If you choose this option, you can use the controls to specify exactly how many grid sections are displayed. The size of the grid sections will be adjusted automatically to match any changes in sheet size.
Specify the size of the grid sections	This setting forces the grid sections to be a fixed size, regardless of changes in overall sheet size. In this case, the grid section width may not exactly divide into the sheet width, so you must also specify how to handle any leftover space.

The last page of border options includes these settings:

Show background grid on screen	If this switch is on, the background grid determined by the previous settings is drawn on the screen.
Print background grid	This switch determines whether the background grid is included in printed or graphics file output.
Number of minor grids per major grid	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.

### 11.3.5 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 Properties command (or hit Ctrl+I) in the right-click menu.

## Reference

- .enable the "Make Background" and "Make Border" switches.
- .click OK.

The object will now be treated as part of the sheet background.

See [Background Objects](#) and [Border Objects](#) for more information on background objects and how to select them.

### 11.3.6 Creating Multipage Templates

EMTPWorks 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 Pages command (in "Options>Design") and clicking on the "New Page" button. You can create a multipage template and reuse it for all new designs based on this template.

More information on using pages in designs can be found in [Adding, Deleting and Titling Circuit Pages](#).

### 11.3.7 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.

## 12 Initialization file EMTP.INI

### 12.1 INI File General Format

The EMTPWorks initialization file is found in the top EMTPWorks directory, it is named EMTP.INI.

The EMTP.INI file is divided into sections. Each section starts with a section heading contained within square braces, e.g. [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 "//".

The initialization file is also called the INI file.

In addition to these notes, you can simply open the INI file and browse through its sections which are explained through comments.

Although some changes in the INI file may become automatically active, most changes become apparent and operational only after restarting the program.

**IMPORTANT:** It must be remembered that patches and new installers may change the EMTP.INI. If you make changes to this file, you should create a backup copy in a personal folder. Before copying back your own version of INI file, you must compare the contents and include your changes in the latest release of this file.

### 12.2 Specifying File and Folder (Directory) Names

A number of the INI file keywords are used to specify file and folder locations for various purposes. This section describes a number of features available to make it easier and more flexible to specify the locations of items on disk.

Folders can be specified using either an *absolute* (i.e. starting with a disk name) or *relative* path. If the path begins with a drive or server name, the path is absolute; otherwise it is relative to the location of the EMTPWorks application. The following are valid statements:

Directory = Libs

specifies a folder called Libs but does not give its absolute location. In this case, the program looks into the folder containing the application itself.

Directory = d:\4000 Series

specifies a folder called "4000 Series" on disk D.

Directory = ..\Libs\7400 Counters

specifies a folder called "7400 Counters" which is contained inside a folder called Libs which is in the same folder one level up from the application.

You can also use file path variables. The following are available:

%my documents% %program files% %windows% %desktop%

#### 12.2.1 Use Environment and Registry Variables

Most places in the INI file (as well as Export scripts) that allow you to specify a file or folder name, also allow references to Windows environment variables and registry entries as sources of information about file location. This is primarily intended for use in cases where references are needed to other applications or data files that may not be installed with EMTPWorks and are not in a fixed location.

It is beyond the scope of this section to discuss how environment or registry entries are set up. Please refer to the appropriate Windows documentation.

## Reference

The following example makes reference to an environment variable called "DataDir":

```
menu= XML, "Generate XML Netlist", "%DataDir%xml_netlist.rfm"
```

The value of the variable is placed in the resulting name strictly by text substitution without any checking. It is the user's responsibility to ensure that this results in a valid file name with appropriate name separator characters (e.g. backslashes between directory names), etc. For example, in the above case, the variable DataDir should contain a terminating backslash character in order to create a valid file name when it is combined with the other text in the file name. Also note that some commonly-used system environment variables such as PATH may contain a list of alternate directories to search. When used in this context, EMTPWorks DOES NOT automatically break out such lists and search all the specified directories.

Here is a sample that refers to a registry entry in order to locate an outside application to execute in response to a custom menu (note that all of the following must be on one line):

```
menu= Simulate, "Run Simulator",  
"%HKEY_LOCAL_MACHINE\SOFTWARE\JoesApps\SuperSim\AppPath%"
```

Since the Windows registry allows multiple values to be associated with a separate name for each value, this format also allows a value name to be specified by appending a colon ":" and the name, as follows:

```
Library =  
"%HKEY_LOCAL_MACHINE\SOFTWARE\HQ\EMTPWorks:InstallPath%\Libs\thing.cif"
```

### 12.3 [DevEditor] Section

This section contains items affecting the device symbol editor tool.

#### Default Font

```
Font = "font_name" font_size [BOLD ITALIC]
```

This statement specifies the default font for text items appearing in a device symbol editor 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 symbol editor's 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.

#### Default Pin Name

```
PinName = PIN1
```

```
BusName=BUS1
```

These entries set the default name used for normal pins and bus pins, respectively. When a new pin is added to a symbol, the exact name given is tried first. If this name is already in use, a



## Reference

Bus_Selected_Col	When the bus (bundle) is selected
BusPin_Col	The bus (bundle) pin color
BusPin_Selected_Col	When the bus (bundle) pin is selected
Pin_Col	Normal pin color
Pin_Selected_Col	When the pin is selected
PinNumber_Col	
PinNumber_Selected_Col	
Device_Col	Default line color for devices
EMTPBus3_Selected_Col	3-phase bus selected color
EMTPBus3_Col=DkRed	3-phase bus color
EMTPConn3_Selected_Col	3-phase signal color when selected
EMTPConn3_Col	3-phase signal color
EMTPPhaseA_Selected_Col	Phase A signal
EMTPPhaseA_Col	
EMTPPhaseB_Selected_Col	Phase B signal
EMTPPhaseB_Col=Blue	
EMTPPhaseC_Selected_Col	Phase C signal
EMTPPhaseC_Col=Green	
EMTP_BUS3_WIDTH = 3	3-phase bus width
EMTP_SIG3_WIDTH = 2	3-phase signal width
EMTP_PHASE_WIDTH = 1	Phase signal width
BlobScale	This is a blob scale for a blob appearing when connecting to a bus
EMTP_BUS3_BLOBS	Turns on (true) the blob or turns off (false)

The available colors are (letter case is not significant in color names):

Red

Green

Blue

Cyan

Magenta

Yellow

Black

DkGray

Gray

LtGray

White

VLtGray

LtBlue

LtGreen

LtRed

DkBlue

DkGreen

DkRed

Violet

Orange

### Templates

Templates = folder

The Templates keyword specifies the location of the design template files that will be displayed in the New Design box. This keyword can be specified multiple times to point to multiple template folders, if desired. See [Design Templates](#) for more information on the usage of templates.

### Examples

Examples = folder

The Examples keyword specifies the location of the example files that will be displayed when the Examples button in the New Design box is clicked. This keyword can be specified multiple times to point to multiple example folders, if desired.

### Default Mask Script

DEFAULT\_MASK\_SCRIPT = script\_black\_box.dwj

This is default subcircuit mask script. See [Masking](#).

### Attribute Probe Scripts

This section specifies the location of scripts used to create the text displayed by the "Attribute Probe" cursor. Either a report (Export) generator script (.rfm) or JavaScript (.dwj) function can be used. The current setup is

ProbeScriptDev=Utility Scripts\Probe\_Dev.dwj

ProbeScriptSig=Utility Scripts\Probe\_Sig.dwj

ProbeScriptPin=Utility Scripts\Probe\_Pin.dwj

These are located in the folder "Utility Scripts". The probe can be used for devices, signals and pins. See [Device Attribute probe](#) and [Signal and Pin Attribute probe](#) for more information on this function.

### Popup menus

Allows specifying parameter menu items in the object specific popup menus for devices, signals, pins and circuits. See [Creating Custom Menus](#).

### Backup

Backup = On

This statement specifies that you want an automatic backup created each time an existing circuit file is resaved. The default is On.

## Reference

AutoSave = 15

This statement specifies that you want to be prompted to save your work whenever the specified number of minutes has passed without a save. The default is 0, i.e. no auto save.

See also [Backup Procedures](#).

### **Device date stamping**

This statement controls the application of a date stamp to devices whenever they are edited.

NODATESTAMP

This keyword does not take any value. If the keyword appears, date stamping will be disabled. If it does not appear, date stamping is enabled.

See also [Device Date Stamping](#).

### **Subcircuit Port Prefix**

The keywords of this section allows specifying the default prefixes for different types of ports when subcircuits are created (see [Subcircuit creation methods](#) and [Subcircuits and Hierarchy](#)).

The current settings are:

SubcctPortPrefixIn = "i"

SubcctPortPrefixOut = "o"

SubcctPortPrefixPower = "p"

### **Sheet sizes**

The SheetSize keyword is used to specify the standard sheet sizes that are available in the Sheet Size Wizard command using the following format:

SheetSize = name width height

Where:

name is the text that you want to appear in the sheet size list. This should be surrounded by double quotes if it contains any non-alphanumeric characters.

width and height are decimal numbers expressed in internal units:

1 inch = 1008

1 cm = 397

For example:

SheetSize = "ANSI A Portrait - 8-1/2W x 11H" 8568 11088

SheetSize = "ANSI A Landscape - 11W x 8-1/2H" 11088 8568

SheetSize = "ANSI B - 17W x 11H" 17136 11088

SheetSize = "ANSI C - 22W x 17H" 22176 17136

SheetSize = "ANSI D - 34W x 22H" 34272 22176

SheetSize = "ANSI E - 44W x 34H" 44352 34272

See also [Setting Sheet Sizes and Borders](#).

**Scales**

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**

BREAKOUT = dth dtv;

The BREAKOUT keyword lets you control the creation of bundle 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 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 keyword indicates the number of levels of undo which should be maintained. A value of zero means that there is no undo. The default value is 10.

UNDO = n

Undo data is held in memory during program operation. A higher value here will increase the memory space occupied by the program.

### **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.

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.

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. The second statement increases what is checked.

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.

## **12.5 [Export] Section**

### **Predefined Script Variables**

ScriptVar = path, "d:\mydocs"

The ScriptVar keyword allows you to predefine variables that can be accessed from within an Export script. This can be used to set values such as the location of certain directories that may vary from user to user but will be fixed on that one computer. The format is:

ScriptVar = varName, value

- varName is the variable name, consisting of any letters, numbers, underscore or period, up to 16 characters long. This variable can be referred to within the script using the &varName format.
- value is any text. If the value contains any non-alphanumeric characters, the value should be enclosed in double quotes.

### **Creating Custom Menus**

This section is for standard menus appearing on the top bar. It is documented in the INI file and in [Creating Custom Menus](#).

### **Location of Export (Report) Scripts**

ScriptFolder = Export Scripts

This statement defines where to look for the scripts to display in the Export script list. The specified location is relative to the directory containing the EMTWorks application. This keyword can be specified multiple times to add multiple directories, if desired.

Scripts = "Report Scripts"

This statement specifies the location of Export (Report) scripts.

## 12.6 [Find] Section

### Search Script Location

```
ScriptFolder = dir_name
```

This keyword specifies the location of search scripts that will be available to the user in the Find tool. All files with the ".RFM" extension that are located in the specified directory will be added to the Find tool's list. This keyword can be specified multiple times (one per line) if desired, to add more directories.

## 12.7 [Jscript] Section

### Script paths

This keyword is used to name the paths (folders) that must be searched by EMTPWorks when a JavaScript file is invoked from anywhere in the design and in the INI file menus.

```
Scripts = Folder name
```

Several such lines can be present for indicating several locations. If a script is called without a path, then EMTPWorks searches the named directories to find it. Both ".dwj" and ".js" files can be used.

### Panel folder

This keyword is used to specify the location of the panel folder. The panel is a web page called by the EMTP computational engine during the design simulation process. This is reserved for advancing programming and should not be modified by users.

## 12.8 [Libraries] section

### Specifying the Default 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 EMTPWorks executable, or if you prefer to specify a complete file "library\_path" in each library statement.

### Specifying a 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. For example:

```
LIBRARY = Lib1.clf
LIBRARY = lib\74LS00.clf
LIBRARY = \mylibs\blocks\controls.clf
```

### Opening 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. This keyword can be specified several times.

## 12.9 [System] Section

### Tools Directory

```
ToolFolder = c:\dw\tools
```

This statement defines where to look for the external code modules. If this value is not specified then modules are loaded from the tools subdirectory under wherever the EMTPWorks executable is located.

### Default Font

```
Font = "font_name" font_size [BOLD ITALIC]
```

This statement specifies the default font which the EMTPWorks system will use when no other font has been specified. Certain program 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
```

### Printer Line Scaling

```
PrinterScaleLines = 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, i.e.:  $\geq 300$  dots per inch, this will cause thick screen lines (busses/bundles) to be reproduced as printed lines which do not appear to be much thicker than 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 than 1 pixel will be scaled when printed. The result is signal lines being drawn very finely, but busses/bundles appearing as thick lines.

### Printer Color Mapping

```
PrinterColors = Yes or ToBlack
```

This keyword controls translation of colors to black and white during printing. This is used in cases where the printer driver being used does not properly handle color requests. If PrinterColors=ToBlack is specified then all colors are translated to black for printing. If this keyword is missing or an invalid option has been specified then no translation of colors will occur.

### Clipboard Color Mapping

```
ClipboardColors = Yes or ToBlack
```

This is used to control translation of colors to black and white when placing an image on the clipboard. If ClipboardColors=ToBlack is specified then all non-white colors are translated to

black. If this keyword is missing or an invalid option has been specified then no translation of colors will occur.

## 12.10 [TextEdit] Section

### Additional Text Document Types

You can specify additional file extensions to be handled by the built-in text editor. For example, if you frequently create a text report from your designs that has the file extension "rpx", you can add this extension to the list of handled file types. This means that this file type will appear in the New and Open commands, allowing easier access to those files.

These additional file types are not registered with the Windows system. You will not be able to open them in EMTPWorks by double-clicking on a file in the Windows Explorer.

For example, the following line adds handling for report script files having the .RFM extension.

TextDoc =

```
"\nReport Script \nReport Script\nReport Script Files (*.rfm)\n.rfm"
```

All of the above text must appear on one line in the INI file. Notes on this format:

- The first section (before the first \n) specifies the default name for a new, untitled document.
- The second section provides the name that will be used to show the file type in the New document type list. If this is omitted, then it will not be possible to create this file type using the File>New command.
- The third section provides the string that appears in the "Files of type" selection list in the Open file box. If not specified, this type will not appear in "File>Open".
- The last section specifies the default file extension applied when a "Save As" is done on this file type.

## 12.11 Creating Custom Menus

EMTPWorks allows adding your own menu items in several locations:

- Standard menu items appearing on the top of the design page (top bar)
- Right-click menu items (popup) on devices
- Right-click menu items (popup) on signals
- Right-click menu items (popup) on pins
- Right-click menu items (popup) on circuit

The standard menu items are created in the **[Menus]** section.

The syntax is:

menu= menuName, script, script argument, tooltip, icon, type of button

- menuName: is the title of the menu.
- Script: is the name of the script file or other function to run. This can be given in full path or relative to the program folder.
- script argument: is the optional arguments strings that will be passed to the called function. If you are using a report generator (or Netlist generator), for example, these allow you to use one report script for different purposes by passing different arguments. In Export type scripts, the value of script arguments is accessible as &ARG1.
- Tooltip: is the text displayed as tooltip when this menu item is highlighted

## Reference

- icon: is the shown command icon file, this file is located in the "Toolbar Icons" folder
- type of button: specifies the type of button to be used (button\_large, button\_small, popup\_large)

It is possible to create toolbars with submenus (commands) by naming the toolbar first and following it with command buttons. In this example:

```
menu= "Design\Device Operations\Show/Hide", "", "", "Device name
operations", "Toolbar Icons\hide.png", popup_large
menu="Design\Device Operations\Show/Hide\Hide Device Names", "info
scripts\shownames\hide_device_names.dwj", "", "Hide the names of
the selected devices", "Toolbar Icons\hide.png", button_small
```

The Design ribbon uses a toolbar named "Device Operations" and a menu named "Show/Hide". The "Show/Hide" menu is given a submenu item named Hide Device Names.

The right-click (popup) menu items are created in the **[Drawing]** section.

This is used to specify parameter menu items in the object specific popup menus, for devices, signals, pins and circuits.

- The first field is a keyword for indicated the target object: PopupScriptDev, PopupScriptSig, PopupScriptPin or PopupScriptCct.
- The following item on each line is the menu text. This can contain the "|" character to indicate submenus or "-" to indicate a separator.
- The third item is either the name of an attribute field or the actual name (full or relative path) of the script to run. If the string given here ends in ".dwj" or ".rfm" it is taken as JavaScript or Export script respectively. Otherwise it is the name of an attribute field. If an attribute field is given, the program looks first in the object itself for a field with the given name. If no value is found, it then looks into the design for the same field.

Any value containing blanks or commas must be enclosed in single or double quotes.

The named scripts are also search in the folders named in [Jscript] section.

This is currently used to provide extra menu items for devices, signals and pins. You can start from the existing examples in the INI file and modify to accommodate other needs.

## 13 Keyboard Shortcuts

### 13.1 Keyboard Shortcuts

The following table lists key combinations used in the schematic drawing and symbol editor windows.

The "+" or "-" character appearing between keys is to indicate that the keys must be pressed simultaneously.

Command	Schematic	Symbol Editor
AddPins	-	Ctrl+E
Auto Create Symbol	-	Ctrl+J
Bring to Front	-	Ctrl+B
Close	Ctrl+W	Ctrl+W
Copy	Ctrl+C	Ctrl+C
Create Subcircuit Block	Ctrl+Shift+Q	
Cut	Ctrl+X	Ctrl+X
Design Attributes	Shift+D	
Design Preferences	Ctrl+J	-
Draw Bundle	Ctrl+Shift+B	-
Draw Signal	Ctrl+Shift+S	-
Duplicate	Ctrl+D	Ctrl+D
Draw Text (Text pointer)	Ctrl+E	-
Enlarge	Ctrl+Shift+E	Ctrl+Shift+E
	Alt+PgUp	-
Export	F2	-
Export Again	Ctrl+F2	-
Find	Ctrl+F	-
Go to selected	Ctrl+G	-
Properties	Ctrl+I	-
Group	-	Ctrl+G
Magnify	Ctrl+Shift+M	
Move to Grid	-	Ctrl+Shift+G
New	Ctrl+N	Ctrl+N
New Breakout	Ctrl+B	-
Normal Size	Ctrl+Shift+N	Ctrl+Shift+N
	Home	-
Open	Ctrl+O	Ctrl+O
Orientation (for device)	Ctrl+Q	-
Part Attributes	-	Ctrl+I
Paste	Ctrl+V	Ctrl+V
Paste Special	Ctrl+Shift+V	
Pin Attributes	-	Ctrl+Shift+I

## Reference

Pop Up (to parent circuit)	Ctrl+Shift+U	-
Print	Ctrl+P	-
Push Into (subcircuit)	Ctrl+Shift+I	-
Redo	Ctrl+Shift+Z	-
Redraw	End	-
Reduce	Ctrl+Shift+R	Ctrl+Shift+R
	Alt+PgDn	-
Reduce to Fit	Ctrl+Shift+F	Ctrl+Shift+F
	Ctrl+Home	-
Revert	Ctrl+M	-
Rotate Left	Ctrl+R	
Rotate Right	Ctrl+Alt+R	
Save	Ctrl+S	Ctrl+S
Save As	Ctrl+Shift+A	-
Select All	Ctrl+A	Ctrl+A
Send to Back	-	Ctrl+Shift+B
Subcircuit and Part Type	-	Ctrl+Q
Undo	Ctrl+Z	Ctrl+Z
Text Font	-	Ctrl+T
Ungroup	-	Ctrl+U
Zap	Ctrl+H	-

## 13.2 Other Key Usage

Key	Function
Delete	Deletes selected items
Arrow Keys	Flip and rotate items while placing a device or pasting
Arrow Keys	Move selected items in small steps
Ctrl or Shift or Alt	Path controls over signal lines while drawing signals. Press these keys after starting to draw with the mouse pointer.
Ctrl+Shift	Control device alignment while placing devices. Press these keys while a device is stuck to the mouse pointer.
Ctrl+Shift+Tab	To move between design windows in the main EMTPWorks window.
Page Up (PgUP)	Scroll Up
Ctrl+PgUp	Scroll Left
Page Down (PgDn)	Scroll Down
Ctrl+PgDn	Scroll Right
Spacebar or Esc	Exits and moves back to the normal pointer when a signal pointer is active. Also stops the sticky action of placed or duplicated objects. If you are working with the text tool pointer, then you must first exit the text and then click the Spacebar. To exit the text, click on an empty space in the design.

If you hold down the Ctrl button while holding down and moving the mouse pointer, you get a quick way for navigating around a drawing.



## 14 FAQ & Tech Notes

### 14.1 Installation issues

#### 14.1.1 Can I install more than one version of EMTP on my computer?

You can install several version of EMTP/EMTPWorks on your computer. If you decide to do so you must be aware of the following issues:

- If your first version was installed under the default standard directory "C:\Program Files (x86)\EMTPWorks", during a subsequent installation you need to ask the installer to use another directory for creating an independent version.
- The default version is the last version. It is also the automatically launched version when a design file is double-clicked to be opened.
- To access previous version you need to start the related program file, "C:\Program Files (x86)\EMTPWorks\EMTPWorks.exe", for example. You can also create a shortcut to each version executable file.
- If you decide to uninstall EMTPWorks, using Windows, then only the last version will be uninstalled.

### 14.2 Schematic Editing

#### 14.2.1 Modifying a Bundle Pin on an Existing Device

The Bundle Pin Info command is available from the right-click menu of a pin.

There are a few points to keep in mind when using this command:

- Adding or deleting bundle internal pins is making a structural modification to the device definition. If this symbol has been used in multiple places on the diagram, you will be asked whether you want to update all devices of the same type, or only this one.
- It is important to verify that device simulation data supports adding pins. In several cases this may result into a data error.
- Modifying the device definition in this way makes it different from the symbol definition that is stored in the original library. If you use the "Update from Lib" command or any other mechanism that causes the device to revert back to its original definition, these changes will be lost.
- If this is a subcircuit device, the bundle pin must correspond to the bundle port connector in the subcircuit. The "Bundle Pin Info" command attempts to make corresponding modifications to the port connector, but it is up to you to verify these changes and make sure the new internal pins connect to an appropriate signal in the internal circuit.

#### 14.2.2 Setting Device Color

I want to change the color of a device on a schematic to mark its status without changing the symbol itself. How can I do this?

To set the color of a device on a diagram, simply right-click on the device and select the "Fill Colour" or "Line Colour" command. The "Fill Colour" command sets the color used for the interior of closed objects like rectangles, ovals, etc. The "Line Colour" command sets the color for lines and for the line border part of filled objects.

Note the following points regarding these settings:

- Device symbols can be arbitrarily complicated with any number of graphic elements and different colors. Applying a single color setting using one of these commands may not always give the desired result. For this reason, the color settings applied when creating a symbol have an "overridable" tag associated with them, allowing you to select which elements will change color in response to settings on the device. Most color settings are overridable by default. This means that when the "Fill Colour" or "Line Colour" commands are used to apply a new color, those settings will be changed to the new color. In the symbol editor, any color setting can be made "not overridable". This means that the "Fill Colour" and "Line Colour" commands will have no effect on these specific graphic elements. It is therefore possible to create a device symbol that will be immune from these override commands and the color settings will appear to have no effect on these devices.
- The "Fill Colour" and "Line Colour" commands do not change the symbol definition and affect only the single device selected when the command is applied. These settings are stored with the device instance and are applied "on the fly" as the device is drawn.

### 14.2.3 Attributes

#### 14.2.3.1 Making an Attribute Field Visible on All Devices

How do I show or hide the value of an attribute field on all the devices in a circuit at once, without having to click on each one?

The easiest way to change the visibility of attributes on a large collection of devices (or signals or pins) at once is to use the Browser tool. Follow these steps:

- ❑ If you want to change a specific group of devices, select them in the schematic first. If you want to change all devices, make sure nothing is selected by clicking in an empty area of the diagram.
- ❑ Display the Browser panel by using the Browser command in the View menu.
- ❑ If the attribute field that you want to modify is not already showing in the Browser's spreadsheet area, display it by double-clicking on it in the field list.
- ❑ Click on the column heading for the desired field to select all rows of that field.

Go to the Browser's Edit menu (not the program's main Edit menu) and select the "Make Visible" or "Make Invisible" command, as desired.

#### 14.2.3.2 Preventing a Name from Being Made Visible

I have created a special purpose device symbol and I don't want the name to be made visible when I place it on a diagram, even though the Auto-assign setting for the design is "visible". How do I do this?

To prevent the auto-assigned name from being made visible on a specific symbol, you can set the visibility options on the "Name" attribute while you are editing the symbol. To do this:

- ❑ Open the symbol in the symbol editor.
- ❑ Select the "Part Attributes" command in the "Options" menu.
- ❑ Click on the "Name" field in the field list.
- ❑ Set the Visibility option to "Never Visible".
- ❑ Enter any text (e.g. "X") for the value of the "Name" field.
- ❑ Click "Done", then close and save the part.

**IMPORTANT:** You must enter a value for the "Name" field even if it is going to be changed as soon as the symbol is placed in the schematic. If you do not enter a value, the field is not saved with the symbol and the visibility setting will be lost.

### 14.2.3.3 Preventing a Name from Being Assigned

I have created a special purpose device symbol and I don't want a name to be auto-assigned to this device when I place it on a diagram, even though the Auto-assignment is enabled for the design. How do I do this?

To prevent the auto-assigned name from being made visible on a specific symbol, you can set a special value for the "Name.Prefix" attribute while you are editing the symbol. To do this:

- Open the symbol in the symbol editor.
- Select the "Part Attributes" command in the "Options" menu.
- Click on the "Name.Prefix" field in the field list.
- Set the value of this field to be exactly "\$NONAME".

Click "Done", then close and save the part.

### 14.2.3.4 Deleting Attribute Fields from a Symbol (Device)

How do I delete an attribute field definition from a symbol stored in a library? There is no delete button and when I remove the value and come back, the field name still appears in the list.

There is no explicit command in the symbol editor to delete an attribute field from a symbol. If you have defined a value for a field and now wish to remove it, simply use the "Part Attributes" or "Pin Attributes" command to set the value to empty, i.e. delete the value completely from the text box. When the symbol is saved, any field that has no value is discarded.

You can however delete some attribute fields from the design which will also eliminate them from all devices appearing in the design. This could be a dangerous action if devices need a given attribute to transmit data. To delete attribute fields from a design go to the "Options>Define Attribute Fields" menu.

NOTE: For convenience, the symbol editor remembers all attribute field names used during a session in any symbols it encounters and displays these names whenever you set a value in any symbol. When you follow the above instructions to delete a field, the field name will still appear in the list even if it is not stored with the symbol.

## 14.3 Symbols (Devices) & Libraries

### 14.3.1 Deleting Attribute Fields from a Symbol (Device)

How do I delete an attribute field definition from a symbol stored in a library? There is no delete button and when I remove the value and come back, the field name still appears in the list.

There is no explicit command in the symbol editor to delete an attribute field from a symbol. If you have defined a value for a field and now wish to remove it, simply use the "Part Attributes" or "Pin Attributes" command to set the value to empty, i.e. delete the value completely from the text box. When the symbol is saved, any field that has no value is discarded.

You can however delete some attribute fields from the design which will also eliminate them from all devices appearing in the design. This could be a dangerous action if devices need a given attribute to transmit data. To delete attribute fields from a design go to the "Options>Define Attribute Fields" menu.

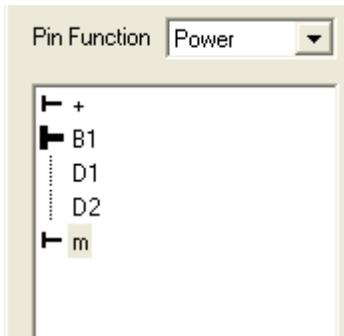
NOTE: For convenience, the symbol editor remembers all attribute field names used during a session in any symbols it encounters and displays these names whenever you set a value in any symbol. When you follow the above instructions to delete a field, the field name will still appear in the list even if it is not stored with the symbol.

### 14.3.2 Adding Internal Pins to a Bundle Pin

I have an existing bundle pin on a symbol and I want to add more internal pins in the same bundle. How do I do this?

You can add more internal pins to an existing bundle pin using the "Add Pins" command. The association of internal pins with a bundle pin is simply by order, so you have to be careful to move the new items to the correct position in the pin list.

For example, supposing you have a symbol open in the symbol editor with the following list of pins:

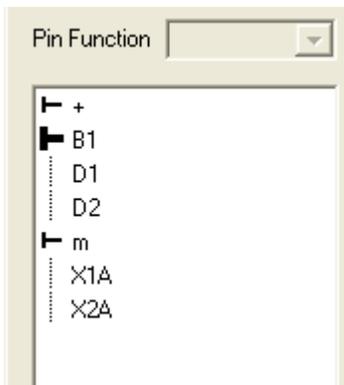


To add two more pins (say X1A and X2A) to the bundle B1, follow these steps:

- Select the Add Pins command in the Operations toolbar.
- Enter the text

[X1A X2A]

You should now see the following pin list:

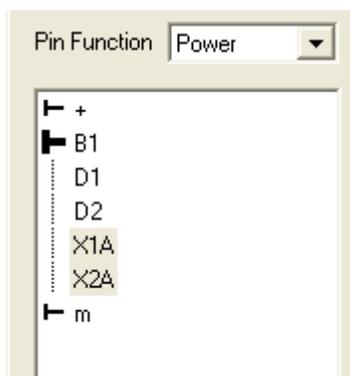


The square brackets around the list of names is important as this signifies bundle internal pins. If you don't include these, you will get normal visible pins and they will not be associated with the parent bundle.

**IMPORTANT:** This order is invalid because you have bundle internal pins following a non-bundle pin. The symbol editor allows this while you are working but will not save the symbol in this state.

- Click and drag the items X1A and X2A into the position immediately after D2 (or you can position them anywhere between B1 and m, if you prefer a different order).

You should now have the following list, which has a valid pin order:



### 14.3.3 Opening Libraries Automatically

I have created my own library and want it to open automatically each time I use (start) EMTPWorks. How do I do this?

Each time it starts up, the EMTPWorks program looks in a number of places for libraries to open. Depending on your requirements, you can do any of the following:

- Use the "Auto-open Libraries" menu from "Options>Libraries>Library Operations"
- Put your library in the default folder where the common libraries are located.
- Put a "shortcut" to your library in the default folder.
- Modify the EMTPWorks initialization file to specifically locate your file or folder.

The first option is the most convenient and simple to use. We'll give a quick rundown on the following two options here. For information on the last option, please refer to [\[Libraries\]](#) section.

#### 14.3.3.1 Placing Libraries in the standard "Libs" Folder

When EMTPWorks is installed, a folder called "Libs" is created in the same folder that contains the program. All libraries inside this folder will be opened automatically when the program starts up. The only disadvantage to placing items in this folder is that they may become mixed up with the standard libraries provided and it becomes more difficult to track and backup your work. You need administrator privileges on your computer to perform this task.

#### 14.3.3.2 Placing a Shortcut in the "Libs" Folder

An extension to the above method is to place a Windows shortcut in the "Libs" folder, rather than the file itself. This has the advantage that you can leave your own files in a separate location where they won't get accidentally deleted when updating or de-installing the software. The methods for creating shortcuts vary with different versions of Windows, but the simplest method is to right-click on the file in the Windows Explorer and select the "Make Shortcut" command. Then, drag the shortcut file to the "Libs" folder. It can be renamed if desired.

### 14.3.4 Adding an Item to the Symbol Gallery

How do I add my own elements to the "Symbol Gallery" list in the symbol editor window?

The "Symbol Gallery" is simply a normal symbol library file which is selected to have its contents displayed in the "Symbol Gallery" list. Thus, all the usual methods for creating, editing and updating symbols can be used on items in the library.

The "Symbol Gallery" file is not normally open by default, so it must be explicitly opened before you can save items to it or edit it. To open this file, right-click in the parts list and select the "Open Lib" command. Use the file open box to select the "Symbol Gallery" file.

In a default EMTPWorks installation, this file is called "symbol gallery.clf" and is located inside a folder called "Data Files" inside the EMTPWorks program folder. If the EMTPWorks initialization file has been modified to specify a different file, you will have to locate that file.

You must have administrator rights on your computer to modify files in the program folder.

### 14.3.5 Creating a Fixed-Phase Pin on a Symbol

How do I add a pin to a symbol that is fixed as a single-phase (3-phase) connection and cannot connect to a 1-phase (single phase) line?

Normally, when you add a pin to a symbol and set the pin function to Power, that pin can be connected to either a 1-phase or a 3-phase power signal. In some case the user may decide to force the phase setting to maintain a given set of data or for making a device readily available in the library. In some cases, models or subcircuit blocks only make sense with a certain fixed pin configuration.

To force a fixed phase setting on a pin, follow these steps:

- Open or create the symbol in the symbol editor.
- In the pin list, select the pin that you wish to set.
- Verify that the "Pin Function" setting is Power.
- Select the "Pin Attributes" command in the Operations toolbar.
- Select the Phase attribute field.
- Enter the text "=1" (that is, an equals sign followed by the single digit 1) for a force 1-phase pin, or "=3" to force a 3-phase pin.
- Save and close the symbol in the usual way.

Some devices may not accept a change in pin type, so you must verify that forcing a given pin type is acceptable for the device definition.

To test the symbol, you can place one on a circuit diagram and attempt to connect an incompatible type of signal to it.

It is actually also possible to make this setting on a single instance of a device on a diagram without modifying the symbol definition. To do this, simply enter the same values given above into the pin attributes on the device.

## 14.4 Report Generation

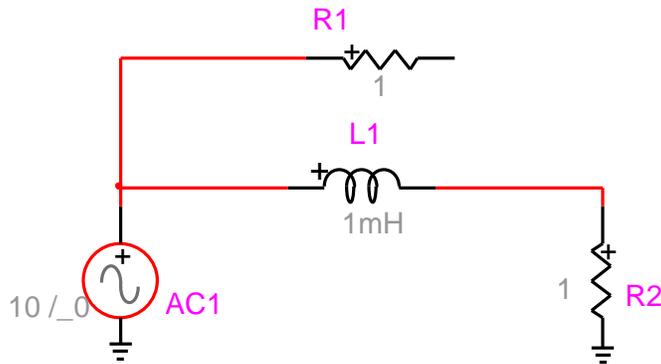
### 14.4.1 Creating an Information Report

How do I create a text report listing information about devices in my design?

The Export command in EMTPWorks can be used to generate text reports containing essentially any information in your design. The format of a report is determined by a script ("report form", or RFM file) that contains commands describing the information you want in the output file. A complete description of the script language is contained in a [separate manual](#). The EMTP Netlist file is generated using the RFM file "EMTP Netslist.rfm" located by default in "EMTPWorks\Report Scripts"

The purpose of this note is to give an example by describing only the commands needed to generate a simple "bill of materials" or device listing report.

For this example, suppose we have created the following circuit ([report\\_case.ecf](#)):



To create the simplest possible report format script from scratch, follow these steps:

- Select the New command in the File menu
- Select the Report Script option
- Type (or Copy and Paste) the following text into the script:

```
$DEVICES$DEVNAME
```

- Save the script file in any convenient location and close the script file. The file must have the ".RFM" extension. Use "[Report\\_Script\\_1.rfm](#)", for example.
- With the circuit you are interested in open and active, select the Export command.
- Choose the "Text Formats" option and click Next
- Use the Browse button to locate the script file we just saved.
- Turn on the "Open Finished Report on Completion" switch, if it is not already on.
- Click Finish. You will be prompted for a location to save the finished report. Save it in any convenient location.

Once the file is saved, you will see a report like the following (obviously, the exact list will depend on the circuit):

```
AC1
```

```
R2
```

```
L1
```

```
R1
```

The script we created indicates that we wish to create a list of devices (\$DEVICES) with the device name (\$DEVNAME) given on each line.

We'll now create a somewhat more sophisticated script that illustrates most of the features you need to create useful reports.

- Enter or copy the following script into the script (.RFM) file ([Report\\_Script\\_2.rfm](#)):

```
$CREATEREPORT($DESIGNNAME_Report.txt)
```

This is the device report:

```
$SORT $DEVICES &Part &Value
```

```
$DEVICES$DEVNAME$COL(30)&Part$COL(45)&Value
```

This will generate a report like the following:

This is the device report:

R2	RLC	1
R1	RLC	1
L1	RLC	1mH
AC1	Vsine	10 /_0

There are a few basic points about the operation of the script language:

- Any item starting with a \$ is a command to the report generator, for example \$CREATEREPORT.
- Any item starting with & is a reference to an attribute field. For example &Part means "output the contents of the Part attribute field". You can easily change this format to output your own attribute fields by adding &fieldName on the \$DEVICES line.
- Anything that is not part of a command is passed through directly to the output file. For example, the "This is..." line appears directly in the output without modification.

Now we'll look at the specific commands used in this script:

```
$CREATEREPORT($DESIGNNAME Report.txt)
```

This line asks to create a report output file using the name of the design followed by "\_Report.txt" as the file name. The file will be created automatically without prompting the user for a name. If there is an existing file with that name, it will be overwritten with no comment. If you want the system to prompt for a new name before proceeding, you can add \$PROMPT to the end of the line.

```
$SORT $DEVICES &Part &Value
```

This says "sort the devices by the Part attribute, then by the Value attribute". In other words, if two devices have the same Part attribute, they will be sorted by Value, otherwise Value is ignored.

```
$DEVICES$DEVNAME$COL(30)&Part$COL(45)&Value
```

This is the heart of this script and causes the actual listing to be generated. To take the commands one at a time:

\$DEVICES – generate a device listing. This will cause one line to be generated for each device, with the format of each line determined by the following commands.

\$DEVNAME – puts out the names of all devices referred to by this line, in this case, all devices having the same Part and Value attributes.

\$COL – specifies a fixed column to advance to, i.e. put out blanks until we reach the given column. Another alternative would be to insert a tab character here.

&Part – means to put out the value of the Part attribute field for the device(s) in question.

\$COL and &Value are similar to above.

## 14.5 Creating Scripts for the Find Tool

Find ("Home>Find>Advanced Find" menu) commands can read a report file generated by the Export tool in a specific format which lists the objects of interest. It reads this file and then displays the results to the user in an interactive manner. Creating an error script thus requires some knowledge of the particular format read by Find and the scripting techniques required to generate such a file.

### 14.5.1 Find Data File Format

The Find module reads a text file containing a list of items to display for the user. All the items in the list usually correspond to objects (e.g. devices, signals or pins) in the design. Some potential

uses of the Find tool may call for the display of data stored in design attributes or even read from external files or software.

For each item, the following information can be supplied:

- The item's title to display in the item list. This should uniquely identify this item in as compact a form as possible, generally no more than 30 - 40 characters. This data is required.
- The object's locator string. This is used to uniquely locate the object in the schematic, even in hierarchical designs. If this is not provided, the "show in schematic" function will not operate.
- A text message to be displayed in the "item info" box when this item is selected in the list. This will normally indicate to the user the type of error found and any other relevant information specific to this item.

An error bit number that can be set to mask out this error. This is used to implement the "Mark as OK" function on the Find control panel. If the user clicks this button, the given error bit will be set in the OKErrors attribute field. It is the script writer's responsibility to check this bit when looking for errors and to skip over this object if it comes up. More information on the Mark as OK function can be found in the [EMTPWorks Report Script Language Reference](#) (Export Tool).

In addition to the information provided per object, the file can also contain a title line that places a title in the Find results box. This will normally indicate the type of error check done, the number of errors found, and other status information.

Script developers should note that if you use the "Mark as OK" (see Find panel opened using "Advanced Find") function in your scripts, you should provide a "Clear OK Errors" script to clear these settings in all marked objects. There is no user interface provided in Find for clearing these flags.

### Example

The following text illustrates a simple Find data file.

Although the format of the command keywords is similar to a Export command script, a Find data file is not a script file. The Export tool will normally be used to generate the data file, as described below.

```
$TITLE Found 2 devices matching C
$OBJECT DD5
$OBJTITLE C1
$MESSAGE Name: C1 - Type: C grounded - Page: 1
$OBJECT DD25
$OBJTITLE C2
$MESSAGE Name: C2 - Type: C grounded - Page: 1
```

This table describes the keywords used in the data file:

\$TITLE	This sets the title displayed at the top of the Find results box. It should be set once at the beginning of the file. The title text starts immediately following the \$TITLE keyword and can continue until any other \$ keyword is found. This means that the title text can be multiple lines, obviously limited by the space available in the output box.
\$OBJTITLE	This sets the short title string used to display the item in the results list. The text follows the keyword on the

	line.
\$OBJECT	This indicates that start of the data for a new object, i.e. an item in the results list. (This does not have to correspond to an object in the design.) Following the keyword is an optional object locator string. The locator will be used to find the object in the design when the user elects to show this object. The locator string is generated using the \$DEVLOC, SIGLOC and \$PINLOC script commands.
\$MESSAGE	This sets the message displayed in the "item info" box on the Find panel. The format is the same as for \$TITLE, that is, the message text starts after the \$MESSAGE keyword and continues for any number of lines until the next \$ command is found.
\$ERRBIT	This sets the bit number used to mask this error using the "Mark as OK" function. If this is not supplied, the "Mark as OK" button will be disabled. This must be a decimal integer between 0 and 31.

Note that all keywords must start at the beginning of a new line and must appear with all upper case letters, exactly as shown.

### Generating a Find Data File Using the Export Tool

When the user selects a script from the list in the initial Find window, these actions take place:

- Find passes the name of the script to the Export tool and asks it to execute it.
- Export executes the script, which must generate a data file in the format described earlier in some known disk location and must pass the full name of the file, including directory path name, back to Find.
- Find opens the data file and displays the title and the list of results.

The execution of the Export tool in this mode is a bit different than operating it directly from the File menu. In this case, when the script starts executing, there is already a "report output file" open which is actually a memory buffer in Find. Any output generated by the script will be interpreted by Find as the file name. The script must contain an explicit \$CREATEREPORT command to generate the disk data file, followed by a \$CLOSEREPORT, followed by commands to generate the name of the file to pass back.

Here is a simple example to look for devices with a given name:

```
$NOTES
Find Devices by Name
```

Enter the desired device name in the text box and click search.

All device symbols having a name starting with the given characters will be located. You can use wildcards \* (any number of characters),  
 ? (any single character), and # (any numeric characters)

```

$ENDNOTES
$TEMPPATH\find.output
$CREATEREPORT($TEMPPATH\find.output)
$FIND $DEVICES $DEVNAME(&ARG1\*)
$SORT $DEVICES $DEVNAME
$title Found $DEVCOUNT devices matching &ARG1
$DEVICES\$OBJECT $DEVLOC$NEWLINE\$OBJTITLE
$DEVNAME$NEWLINE\$MESSAGE Name: $DEVNAME - Type: $TYPENAME - Page:
$PAGE
$CLOSEREPORT

```

Note these points:

- The \$ keywords required by Find in the output file should always be prefixed with an escape character "\" so that Export does not attempt to interpret them as commands.
- You must be careful to ensure that only a single line containing the file name is generated outside the \$CREATEREPORT/\$CLOSEREPORT pair.
- Everything between \$CREATEREPORT and \$CLOSEREPORT is output to the data file. The format of this file must meet the specifications given earlier.
- It is best to specify a complete directory path name when creating the data file and describing its location to Find. This will ensure that the file does not end up in an unexpected location due to the setting of the system's "current directory". You can use the script keywords \$DESIGNPATH, \$SCRIPTPATH, \$TEMPPATH and \$PROGPATH to generate complete path names to known locations without specifying the absolute name of your disk. Note that \$DESIGNPATH will be null if the design has not been saved.
- The "\$TEMPPATH\find.output" line is passed to the Find tool so it knows where to find the data file to open.
- When the user clicks the Script Help ("?") button on the Find panel, the contents of the \$NOTES section (i.e. anything between the \$NOTES and \$ENDNOTES keywords) are displayed in a Help box. This should always be included to describe the function of the script, how it uses the text input box, and any other relevant information.

As long as you keep these points in mind you can use any of the Export capabilities outlined in the [EMTPWorks Report Script Language Reference](#) (Export Tool) to produce the data file.

### Using the User Text Entry Box

The script can optionally make use of the contents of the Search Text box on the Find panel as part of its search. In the script, the contents of this box is passed as an argument to the script and is accessible using the &ARG1 variable. For example, the following script line will locate all devices with names matching the contents of the box:

```
$FIND $DEVICES $EQ(&ARG1, $DEVNAME)
```

### Where Search Scripts are Placed

In order to appear in the Find tool's script list, search scripts must be located in a folder specified by the SCRIPTFOLDER keyword in the initialization file EMTP.INI, normally "Find Scripts".

## 14.6 Printing

### 14.6.1 Fitting a Printout to a Single Sheet

How do I get my circuit to print out on a single sheet instead of being broken into multiple sheets when printing?

EMTPWorks allows you to create a "virtual" drawing area on the screen that can be of any size, independent of the size your printer can actually handle. If this on-screen sheet size is bigger than the printable area, then the diagram must be either reduced to fit on one piece of paper, or broken into multiple sheets for printing.

To change these settings:

- Select the "Sheet Size Wizard" command in the "Options>Design" toolbar.
- Depending on the other settings in your design, you may see a warning page about applying settings to multiple pages. If so, make the appropriate choice and click Next until you see the Border Scaling selection page.

If the Border scaling option is set to "Scale border to match printer setup", then follow these steps, otherwise skip to the next section:

- Click Next. You should see the "Printer Page Mapping" options.
- Select the Advanced option and click Next.
- The Advanced page allows you to select how many pages your circuit prints on. Set the size to 1 x 1 printed pages to force the circuit to print on a single page. If this creates a drawing area that is too small, you can adjust the scale factor to a smaller number to create a larger on-screen drawing area.
- Click Finish to view the results.

If the Border scaling option is set to "Fixed border size", then proceed with these steps:

- Click Next to view the sheet size. Unless you have some reason to want to change this setting, click Next again.
- On the Printed Output page, select the "scale to fit printed page" option and set the size to 1 x 1 pages. This forces the program to scale the drawing to fit on one printed piece of paper, regardless of the on-screen size.
- Click Finish. This change should have no visible effect on the drawing area displayed on the screen.

### 14.6.2 Printing in Black and White

When I print diagrams on my black and white printer, lines that are colored on the screen appear as shades of gray on the printer. How do I get everything to come out in black or white?

When printing, EMTPWorks normally transmits all the color information associated with a drawing to the printer driver, which then makes its own decision how to use it. In some cases, black and white printers attempt to mimic colors by creating shades of gray. If this is not the desired result, you can make a change in the EMTPWorks initialization file EMTP.INI to request that colors are always converted to black or white when printing.

To change these settings, edit the EMTP.INI file (see below) and locate the "PrinterColors" keyword in the [System] section of the file. The following paragraphs describe the possible settings for this keyword.

PrinterColors=Yes

This specifies that all color information should be transmitted to the printer. Some printer drivers may have their own color options that could override this one. If you have specified Yes for this

option and are still getting black and white output, check the "Print Setup" options for your selected printer.

`PrinterColors=ToBlack`

If `PrinterColors=ToBlack` is specified then all non-white colors are translated to black when printing.

If the "PrinterColors" keyword is missing or an invalid option has been specified then "Yes" is assumed.

### Notes on Editing the INI File

All of the options described here are modified by editing the file `EMTP.INI` which is located in the same directory as the program files. The ini file can be edited using NotePad or any basic text editor. If you use a word processor such as Microsoft Word, make sure to use a Text Only option when saving the file.

The changes will take effect only after restarting `EMTPWorks`.

The INI file is a simple text file and the options are divided into various sections, each of which starts with a `[SectionName]` marker. The option you are modifying can appear anywhere after its associated section marker.

### 14.6.3 Line Width Scaling When Printing

When I print diagrams, the signal lines come out too fat or too thin. Can I control this?

Printer drivers vary in whether they consider a line width of "1" to mean "1 screen-dot-width" or "1 printer-dot-width", so line widths that appear in your printouts can vary from what you see on the screen. The `PrinterScaleLines` option in the initialization file `EMTP.INI` lets you specify how you want line widths to be scaled.

This option selects one of three possible ways of setting line width when a diagram is printed. The option that is best for your application will depend on the printer you are using, the scale factor and your own preferences. If this keyword is missing or an invalid option has been specified then NO scaling of lines will occur.

`PrinterScaleLines=None`

This indicates that no scaling should occur. A line that is displayed 3 dots wide on the screen will be displayed three dots wide in the printer's native resolution.

`PrinterScaleLines=All`

This indicates that all lines should be scaled. A line should appear on the printout approximately the same width as it appears on the screen.

`PrinterScaleLines=OverOnePixel`

This specifies that lines one pixel wide (i.e. most signal lines on the screen) will be printed one dot wide in the printer's native resolution. All other lines will be scaled. With some printers, this option makes the best use of fine lines for representing signal traces.

### 14.6.4 Notes on Editing the INI File

All of the options described here are modified by editing the file `EMTP.INI` which is located in the same directory as the program files. The ini file can be edited using NotePad or any basic text editor. If you use a word processor such as Microsoft Word, make sure to use a Text Only option when saving the file.

The changes will take effect only after restarting `EMTPWorks`.

The INI file is a simple text file and the options are divided into various sections, each of which starts with a [SectionName] marker. The option you are modifying can appear anywhere after its associated section marker.

## **14.7 Copying**

### **14.7.1 Copying into Word**

Copying a circuit into Word is simple. You need to select the desired circuit or any part of the circuit or any part of a design page, copy it using Ctrl-C or "Home>Edit>Copy" and pasting into a Word document. The highest resolution results are achieved when copying from EMTPWorks at the highest zoom level. The copied circuit can be resized afterwards in the Word document.

If the selection in EMTPWorks has text objects, then you must select "Edit>Paste Special" in Word and then select "Picture (Enhanced Metafile)".