

USER'S GUIDE

on the use of PSCAD

PSCAD

Power Systems Computer Aided Design



244 Cree Crescent, Winnipeg, Manitoba, Canada R3J 3W1

Copyright © 2005 Manitoba HVDC Research Centre Inc. All rights reserved.

Information in this document is subject to change without notice. No part of this document may be reproduced in any form or by any means, electronically or mechanically, for any purpose without the express written permission of Manitoba HVDC Research Centre Inc.

PSCAD is a registered trademark of Manitoba HVDC Research Centre Inc.

EMTDC is a trademark of Manitoba Hydro, and Manitoba HVDC Research Centre Inc. is a registered user.

Microsoft, Windows 95, Windows 98, Windows NT, Windows 2000, Windows ME, NT, XP, Developer Studio are the registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

DEC and DEC Fortran are trademarks of Digital Equipment Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company.

Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation in the United States and other countries.

MATLAB and Simulink are registered trademarks of The MathWorks, Inc.

Compaq and the names of Compaq products referenced herein, are either trademarks and/or service marks or registered trademarks and/or service marks of the Compaq Computer Corporation.

WinZip is a registered trademark of WinZip Computing, Inc.

Other product and company names mentioned herein may be trademarks or registered trademarks of their respective holders.

Print history:

- February 3, 2003 - Version 4.0.0, first printing.
- March 21, 2003 - Version 4.0.1
- July 18, 2003 - Version 4.0.2, second printing.
- December 23, 2003 - Version 4.0.3
- April 30, 2004 - Version 4.1.0, third printing.
- April, 2005 - Version 4.2.0, fourth printing.

Foreword

The evolution of the power system simulator is born from many forces. Some are driven by the need to solve a problem while others have come directly from scientific research. Over the last decade, the manipulation of increasingly complex systems has been the driving force behind the graphical interface and its working environment. Every year there is an increasing demand for more flexibility and speed, taxing the software in every direction.

I once read a quote, "To do the difficult task easily, you must do the simple tasks perfectly." Nowhere has this been more applicable as in the latest rendition of PSCAD. The fundamentals have been revisited once again to ensure that they perform precisely and consistently every time. Built on the strong foundation left by the original creators of EMTDC some twenty years ago, the software has solidly reached the ranks of world class. This by no means indicates that the tasks are complete. Like the wave on a beachfront, there are always more to come.

As I look back on the last decade of development, the question of the next decade moves into the light. Solutions are found in a montage of tools specifically targeted for their domain. The distinction of these domains fades as the drive toward a unified tool grows. Convergence forces us to view it not as waves on rocks, soil and trees, but to consider it as a beach where all these forces interact. Thus, tackling this convergence and creating a more powerful set of tools is the next great challenge for the team and those that partner with us.

The many hours and hard work that has gone into the software over the last two decades is hard to imagine. It has become and will continue to be the bedrock of development for the future. In the next years, the many tools will form a nexus point and change the nature of those solutions and how they are used once again. I am proud to be a part of the last decade, working alongside all the great contributors and look forward in earnest to the next.

Craig Muller, P. Eng.
Software Development Manager
September 2005

*This masterpiece is dedicated to all the hardworking,
programmers, engineers, students, professors and support
staff who made this program possible.*

Table of Contents

About This Guide.....	xxvii
Organization	xxvii
Documentation Conventions	xxviii
References	xxviii
Notes	xxix
Text Boxes.....	xxix
Chapter 1: Welcome to PSCAD	1
What is PSCAD?	2
Typical PSCAD Studies.....	3
What's New in PSCAD V4?	3
V3/V4 Comparison Tables.....	6
Appearance and Terminology	6
Chapter 2: Installation and Setup	19
Hardware and Software Requirements	19
Fortran Compilers.....	20
Licensing	20
Multi-User Licensing.....	20
Single-User Licensing	22
Trial Licensing	23
Hardware Locks (Dongles).....	23
License Key.....	24
License Manager Software.....	25
Standalone License Manager	25
Maintenance and Support	25
Local License Manager	25
Trial License Manager.....	26
PSCAD Usage Log	26
Installing V4 For The First Time	27
Installing PSCAD.....	27
Installing the License Manager	33
Possible Messages	35
Troubleshooting	40
Installing EGCS/GNU Fortran	42
Installing LiveWire/LiveWire Lite	42
Upgrading Your V4 Software	42
Upgrading a Previous Installation with a Patch.....	42

Table of Contents

Modifying a Previous Installation.....	44
Possible Messages	46
Repairing a Previous Installation.....	48
Possible Messages	50
Uninstalling.....	51
Possible Messages	52
Finalizing Your Installation.....	53
Windows 98/ME Only.....	53
Windows NT4/XP/2000 Only.....	54
Adding a Shortcut to the PSCAD Executable	54
Running PSCAD For The First Time	54
Failed to Connect to License Manager Server.....	54
Running a Simulation Generates a 'make -f' or 'nmake -f' Error.....	56
Setting Compiler Environment Variables.....	57
EGCS/GNU Fortran 77 Compiler.....	57
Digital Fortran 5 Compiler.....	58
Compaq Fortran 6 Compiler.....	58
Intel Fortran 9 Compiler	59
Trial Licenses	60
Requesting a Trial License.....	60
Installing a Trial License.....	64
Adding/upgrading Stand-alone License Manager Licenses	66
Adding/upgrading A Single-user License	67
Changing Active Licensing Settings	68
Changing License Manager Host.....	68
Changing Active License Type	69
Viewing Active License Information.....	69
Getting Licensing Information	70
Standalone License Manager Information	70
Active Licensing Information	71
Manually Starting The License Manager.....	71
Manually Stopping The License Manager	72
Chapter 3: Workspace Settings	73
Workspace Settings Overview	73
Projects	74
Initialization	74
HTML Browser	75
Views.....	75
Circuit View	76
Script Editor.....	77
Graph and Control Panel Style	77
Runtime	77

Output Channel Warnings	78
System Memory Load	78
Fortran	79
Fortran Compiler	79
Matlab®	80
Interface Settings	80
License	81
License Host	81
Trial License	81
Single User License	82
Available License(s)	82
Active License	82
Associations	82
Customize File Associations	83
Creating a New Association	83
Removing an Association	84
Invoking an Associated External Application	84
Chapter 4: The PSCAD Environment	85
Terminology And Definitions	85
Components	85
Definitions	86
Instances	86
Modules	86
Projects	86
Case	87
Library	87
Tutorial: My First Simulation	87
Starting PSCAD	87
Title, Menu and Main Tool Bar	88
Title Bar and Active Project	88
Menu Bar and Menu Items	88
Toolbar Buttons	89
Workspace and Output Windows	89
Loading a Case Project	90
Running a Simulation	92
Printing the Circuit	93
The Workspace	94
Moving and Resizing the Workspace Window	95
The Projects Section	95
Main Page Branch	96
Definitions Branch	97
Active Project	101
The Master Library	101

Table of Contents

The Runtime Section.....	102
Modules View.....	103
Runtime Objects	105
Observers	109
Groups View	110
Radio Links View	111
TLines/Cables	112
The Files Section	114
Transmission Lines and Cable Files	116
RTP/COMTRADE Recorder Files.....	117
The Output Window.....	118
Moving and Resizing the Output Window	118
Errors and Warnings	119
Build	120
Runtime.....	120
Locating the Problem Source.....	120
Search.....	120
The Design Editor.....	121
Viewing Windows	121
Circuit.....	122
Graphic	122
Parameters	122
Script.....	122
Fortran	122
Data	122
Other Important Features	122
Menus	122
Main Menu Bar.....	123
Right-Click Pop-Up Menus.....	123
Library Pop-Up Menus	124
Toolbars.....	125
Main Toolbar	125
Status Bar	126
Rotation Bar	126
Runtime Bar	127
Electrical Palette	127
Control Palette	128
Graphic Palette	129
Graphic Filter Bar	130
Parameters Bar.....	130
Script Bar	132
Keyboard Shortcuts.....	132
General	132

PSCAD

Wires	134
Plotting	134
Window Settings	135
Tab Navigation	135
PSCAD Temporary Directories	136
PSCAD On-line Help System	136
Look and Feel	137
The Help Toolbar	137
The Navigation Pane	137
Contents	138
Index	138
Search	139
Favourites	139
The Main Viewer	140
Tutorial Projects	140
Voltage Divider	141
Fast Fourier Analysis	141
Simple AC System with a Transmission Line	141
Use of Control Arrays	141
Use of Slider, Switch, Button, and Dial	141
Interpolation	141
Generating a Legend and using PSCAD Macros	141
Chatter Elimination	141
Multiple Run	142
Chapter 5: Features and Operations	143
Projects	143
Creating a New Project	143
Loading a Project	144
Recent Files	145
Opening and Viewing a Project	145
Navigating Through an Open Project	145
Scroll Bars	145
Arrow Keys	146
Panning (Dynamic Scroll) Mode	146
Moving In and Out of Modules	146
Forward/Back Buttons	147
Tabs	147
Zooming	148
Refresh	149
Setting the Active Project	150
Compiling and Building a Project	150
Compile	151
Make	151

Table of Contents

Viewing Errors and Warnings.....	152
Finding the Message Source	152
Running a Simulation	153
Changing the Run Duration.....	154
Changing the Simulation Time Step.....	154
Changing the Simulation Plot Step	154
Taking a Snapshot.....	154
Manual	155
Pre-Defined.....	155
Starting from a Snapshot	155
Saving Output to File.....	156
Saving Project Changes.....	156
Save Project As.....	157
Saving as an Older PSCAD Format.....	158
Editing Project Settings	159
Cleaning The Temporary Directory	160
Unloading A Project.....	161
Components And Modules	162
Selecting Objects	162
Adding Components To A Project.....	162
Adding Multiple Instances of a Component	164
Moving Or Dragging An Object	165
Cut/copy/paste	165
Rotate/Mirror/Flip	165
Deleting Objects.....	166
Undo And Redo.....	166
Connecting Components Together.....	166
Invalid Component Connections	167
Transferring Data Into And Out Of Modules	168
Data Signals.....	168
Hard Wiring.....	168
Wireless Connections	170
Electrical Nodes	171
Wire Mode.....	173
Editing Module Settings	173
Display Options.....	174
Show Connection Grid.....	174
Show Signal Locations	174
Show Virtual Control Wires.....	174
Fortran Code Sequencing.....	174
Assign Execution Sequence Numbers	
Automatically	174
Show Sequence Numbers on Circuit Schematic.....	175

Editing The Page Setup	175
Editing Component Parameters	176
Changing Parameter Values	178
Editing Definition Properties	178
Viewing/Saving Component Parameters	179
Viewing Component Parameters	179
Column Descriptions	180
Formatting Viewed Parameters	181
Saving Component Parameters to a File	181
Changing To Single-Line View	182
Creating A New Component Or Module	182
Using the Component Wizard	182
Creating a Default Module	186
Creating The First Instance of a Definition	187
Compiling an Individual Module	187
Importing/Exporting Definitions	188
Import a Definition	188
Export a Definition	188
Copy As Meta-file Or Bitmap	189
Printing Selected Objects	189
Printing a Module Page	189
Tutorial: Creating A New Project	190
Adding a New Case Project	190
Saving the Project	192
Opening the Project Main Page	193
Opening the Master Library	193
Assembling a Voltage Divider Circuit	194
Locating the Single-Phase Source	194
Add and Assemble	196
Editing the Remaining Component Parameters	196
Plotting Results	197
Adding a Graph Frame	197
Adding Another Overlay Graph and Curve	199
Editing the Graph Properties	200
Running the Project	201
Drag and Drop	202
Create a Component Instance	202
Copy a Component Instance	203
Adding a Curve to a Graph	203
From an Output Channel	203
From a Runtime Object	204
Adding a Meter to a Control Panel	204
From an Output Channel	205

Table of Contents

From a Runtime Object.....	205
Adding a Control Interface to a Control Panel.....	206
From a Controls Component	206
From a Runtime Object.....	206
Move/Copy Meters and Curves Between Graphs/Panels...	207
Accessing The Online Help System	207
Tool Tips	208
Searching	209
Component And Signal Tables	210
Component Table	210
Signal Table.....	211
Control Settings.....	213
Storing Settings to a Template	213
Synchronizing Default Settings	214
Duplicate, Rename or Delete Template	214
Template Settings Viewer.....	215
Features and Functions	215
Global Substitutions	216
Unit System	218
Enabling the Unit System.....	218
Unit Format	219
Base Units.....	219
Exceptions	222
Aliases	222
Prefixes	222
Target Units.....	223
Unit Conversions.....	224
Compound Units	225
Verifying Unit Conversions.....	228
Backward Compatibility.....	229
EMTDC Output Files	229
Creating Output Files	230
Multiple Output Files	230
Column Identification and the Information File.....	231
Chapter 6: Online Plotting and Control	233
Preparing Data For Control Or Display	233
Channeling Output Data.....	234
Controlling Input Data	234
Graph Frames	235
Adding a Graph Frame.....	235
Moving and Resizing a Graph Frame	236
Cut/Copy Frames	236
Paste Frame.....	237

Adjusting Frame Properties.....	237
Adjusting Horizontal Axis Properties	238
Graphs.....	239
Adding Graphs to a Graph Frame.....	240
Graph Order	240
Cut/Copy Graphs	240
Paste Graph.....	241
Copy Data to Clipboard.....	241
Overlay Graphs	242
Adjusting Overlay Graph Properties	242
PolyGraphs	244
Adjusting PolyGraph Properties.....	244
Curves.....	245
Adding a New Curve to a Graph	246
Curve Legends.....	247
Curve Order	247
Cut/Copy/Paste an Existing Curve.....	248
Copy Data to Clipboard.....	248
Adjusting Curve Properties	248
Active Trace	249
Style.....	250
Digital Style	250
Adjusting Channel Settings	250
Synchronizing Output Channel Limits with Those of the Graph	250
Traces	251
Trace Drop Down Menu	251
Adjusting Trace Properties	252
Polymeters	252
Adding a PolyMeter.....	253
Moving and Resizing a PolyMeter.....	254
Cut/Copy PolyMeter.....	254
Paste PolyMeter.....	255
Copy PolyMeter as Meta-File/Bitmap.....	255
Navigate to Channel.....	255
Adjusting Channel Settings	255
Displaying Specific Data	255
Phasormeters.....	255
Preparing Data for Display	256
Adding a PhasorMeter	257
Moving and Resizing a PhasorMeter	257
Cut/Copy PhasorMeter.....	258
Paste PhasorMeter	258

Table of Contents

Copy PhasorMeter as Meta-File/Bitmap	259
Navigate to Channel.....	259
Adjusting Channel Settings	259
Displaying Specific Data	259
Adjusting Phase Angle Input Format.....	259
XY Plots.....	260
Adding an XY Plot.....	260
Moving and Resizing an XY Plot Frame	261
Cut/Copy XY Plot Frame.....	261
Paste XY Plot Frame.....	262
Copy Data to Clipboard.....	262
Adjusting XY Plot Frame Properties.....	262
Adjusting Plot Properties.....	264
Polar Grid.....	266
Dynamic Zoom.....	267
Tool Tips In Plots	267
Dynamic Aperture Adjustment.....	268
Adjusting and Controlling the Aperture in XY Plots	269
Markers	270
Show/Hide Markers.....	271
Graph Frame Marker Legends.....	272
XY Plot Marker Legends	273
Changing the Active Curve.....	274
Adjusting Marker Positions.....	274
Toggle Time Difference Frequency/Delta	275
Locking/Unlocking Markers	275
Setting Markers	275
Using Markers as Bookmarks	276
Preferences.....	277
Zoom Features.....	278
General Zoom In and Zoom Out.....	278
Box Zoom.....	278
Vertical Zoom	279
Horizontal Zoom.....	279
Zoom Previous/Next.....	279
Zoom Extents	280
Zoom Limits.....	280
Resetting All Extents and Limits.....	281
Dynamic Zoom in XY Plots	281
Cross Hair Mode	281
Pop-up Tool Bars.....	282
Online Controls And Meters	283
Control Panels.....	284

Adding a Control Panel.....	284
Moving and Resizing a Control Panel.....	284
Cut/Copy Panel.....	284
Paste Panel.....	285
Adjusting Panel properties	285
Control Interfaces.....	285
Adding a Control Interface to a Control Panel	286
Control Interface Order	287
Cut/Copy Control Interface	288
Paste Control Interface	288
Navigate to Control	288
Adjusting Control Interface Properties	289
Using the Control Interfaces	289
Meters	291
Adding a Meter to a Control Panel.....	292
Meter Order.....	293
Cut/Copy Meter.....	293
Paste Meter.....	294
Navigate to Channel/Control.....	294
Adjusting Meter Properties.....	294
Grouping Of Runtime Objects	294
Creating a Runtime Object Group.....	295
Viewing Runtime Object Groups	296
Displaying Group Name on Graph Frames and Control Panels.....	296
Frame/panel Minimization	297
Moving and Organizing Minimized Frames/Panels.....	298
Displaying Plots And Controls In Reports	298
Setting Panel Style.....	298
Copy as Meta-File or Bitmap.....	299
Chapter 7: Project Settings	301
Project Settings Overview.....	301
General.....	302
Description	302
File Path	302
Persist As	303
Revision Tracking.....	303
File Version	303
First Created	303
Last Modified.....	303
Author	303
Runtime	303
Time Settings	304

Table of Contents

Duration of Run (sec).....	304
Solution Time Step (us).....	304
Channel Plot Step (us).....	304
Start-up Method	305
Standard	305
From Snapshot File.....	305
Save Channels to Disk?.....	306
Timed Snapshot	306
Single (Once Only).....	306
Incremental (Same File).....	306
Incremental (Many Files)	306
Multiple Run	307
Miscellaneous	308
Remove Time Offset When Starting From a Snapshot..	308
Send Only the Output Channels That are In Use	309
Start Simulation Manually to Allow Use of an Integrated Debugger	309
Enable Component Graphics State Animation.....	309
Network	309
Process Communication	309
Allow Simulation to Run Ahead at Most	310
Use Idle Time Polling if Network is Large (200+ Nodes)	310
Network Solution Accuracy	310
Interpolate Switching Events to the Precise Time.....	311
Use Ideal Branches for Resistances Under	311
Numerical Chatter Suppression	311
Detect Chatter that Exceeds the Threshold	311
Suppress Effects When Detected	312
Diagnostic Information	312
Echo Network and Storage Dimensions	312
Echo Runtime Parameters and Options	312
Echo Input Data While Reading Data Files.....	312
Dynamics.....	312
Signal Storage.....	313
Store Feed Forward Signals for Viewing	313
Signal Flow.....	315
Compute and Display Flow Pathways on Control Wires	315
Buses	315
Treat Multiple Buses with Matching Names as the Same Bus	315
Unit System.....	316

Enable Unit Conversion and Apply to Parameter Values	316
Mapping.....	316
Network Splitting	317
Split Decoupled Networks into Separate Matrices	317
Combine Isolated Non-Switching Local Networks	317
Matrix Optimizations.....	317
Optimize Node Ordering to Speed Up Solution	318
Move Switching Devices to Speed Up Solution	318
Fortran.....	319
Additional Source (.f) Files	319
Runtime Debugging	320
Enable Addition of Runtime Debugging Information	320
Checks	320
Array & String Bounds	321
Floating Point Underflow.....	321
Integer Overflow.....	321
Argument Mismatch	321
Uncalled Routines.....	321
Uninitialized Variables	321
Link.....	322
Additional Library (.lib) and Object (.obj) Files	322
User Library Path Method	322
Absolute Path Method.....	323
Entering Multiple Files.....	324
Matlab	324
Link This Simulation with the Currently Installed Matlab Libraries	324
Chapter 8: Transmission Lines and Cables	325
Constructing Overhead Lines.....	326
Building an Overhead Line (/w Remote Ends).....	327
Building an Overhead Line (Direct Connection Style).....	328
Editing the Line Main Parameters	329
Constructing Underground Cable Systems	330
Building an Underground Cable System.....	330
Editing the Cable Main Parameters	331
The T-line/cable Properties Editor	332
Editing a Transmission Line Segment Cross-Section	333
Adding a Tower Component.....	335
Editing Tower Properties	337
Editing a Cable Segment Cross-Section.....	337
Adding a Cable Cross-Section Component	339
Editing Cross-Section Properties.....	341

Table of Contents

Selecting the Proper Line Model	341
The Bergeron Model	342
The Frequency-Dependent (Mode) Model.....	342
The Frequency-Dependent (Phase) Model.....	342
Adding a Line Model	343
Editing Line Model Parameters.....	343
Line Constants Files.....	344
Solving the T-Line/Cable Constants Manually	344
Viewing Line Constants Files	345
Input (*.tli/*.cli) File	345
Constants (*.tlo/*.clo) File.....	347
Log (*.log) File.....	347
Output File.....	347
Phase Data	348
Series Impedance Matrix (Z):	348
Shunt Admittance Matrix (Y):.....	348
Long-Line Corrected Series Impedance Matrix (ZLL):	349
Long-Line Corrected Shunt Admittance Matrix (YLL):	349
Sequence Data	349
Sequence Impedance Matrix (Zsq):.....	350
Sequence Admittance Matrix (Ysq):	351
Load Flow RXB Formatted Data	351
Detailed Output Viewer	352
Creating Detailed Output.....	353
Invoking the Detailed Output Viewer	353
The Viewer Environment.....	354
Spreadsheet Data Viewer	354
Column Organization	354
Switching between Spreadsheets.....	355
Copying Data Rows	355
Detailed Output Toolbar	356
Curve Viewer	356
Invoking the Curve Viewer.....	356
Element Identification	357
Chapter 9: Designing Components	359
The Design Editor.....	359
Editing a Component or Module Definition.....	360
Returning To Circuit View	361
The Graphic Section.....	361
Navigating and Zooming	361
Scroll Bars.....	362

Arrow Keys.....	362
Panning (Dynamic Scroll) Mode	362
Zooming	362
Graphic Objects	363
Adding Graphic Objects	363
Rotate, Flip, Mirror and Resize Graphic Objects	364
Changing Graphic Object Properties	365
Changing Arc Object Properties.....	368
Cut, Copy, Paste and Delete Objects	371
Text Labels	371
Adding Text Labels.....	371
Rotate, Flip or Mirror Text Labels.....	372
Changing Text Label Properties.....	373
Cut, Copy, Paste and Delete Labels	374
Linking a Text Label to an Input Field	374
Connections	375
Adding Connections.....	375
Changing Connection Properties.....	376
Cut, Copy, Paste and Delete Connections.....	378
Electrical Node Types	378
Undo and Redo	379
Saving and Reloading Graphics.....	379
Adjusting Graphic Page Size	380
The Parameters Section.....	381
Categories.....	381
Adding a New Category	382
Viewing Categories	383
Ordering Categories.....	384
Previewing Categories	384
Changing Category Properties.....	385
Duplicating a Category.....	386
Deleting a Category	386
Text Field.....	387
Adding Text Fields.....	387
Moving and Resizing Text Fields	387
Changing Text Field Properties.....	388
Cut, Copy, Paste and Delete Text Fields	389
Input Field	389
Adding Input Fields	389
Moving and Resizing Input Fields	390
Changing Input Field Properties	390
Cut, Copy, Paste and Delete Input Fields	392
Allowing Signal Names	392

Table of Contents

Choice Box.....	393
Adding Choice Boxes.....	394
Moving and Resizing Choice Boxes	394
Changing Choice Box Properties.....	395
Cut, Copy, Paste and Delete Choice Boxes	396
Adding Choices to a Choice Box	396
Field Alignment.....	398
Make Same Width.....	398
Space Evenly	399
Undo and Redo	399
Find Symbol	399
Saving and Reloading Parameters	400
Conditional statements, Layers and Filters	401
Conditional Statements	401
Layers	404
Viewing Layers.....	404
Setting Layers	405
Graphic Filters.....	406
The Script Section	407
Segments.....	407
Managing Segments	408
Viewing Segments	409
Segment Types	409
Computations.....	409
Branch.....	411
Fortran	415
DSDYN	418
DSOUT	418
Checks	418
Help.....	419
Comments.....	419
FlyBy	420
Transformers.....	421
Model-Data	424
Matrix-Fill	424
T-Lines	427
Undo and Redo	428
Syntax Colouring	428
Internal Output Variables	429
Outputting EMTDC Measured Voltages and Currents	429
CBR	429
VDC	430
Adding A Reference To A Source File.....	430

Interfacing To C Code.....	431
Using the GNU Compiler.....	431
Using a Fortran 90 Compiler.....	433
Tutorial: Creating A New Component.....	434
Adding Graphics.....	435
Adding a User Interface.....	436
Adding Conditional Statements to the Graphic Objects.....	438
Pre-Processing User Input.....	439
Defining the Electrical Branch.....	440
Adding Code to Define the Source.....	441
Setting the Branch Source.....	442
Sanity Checks.....	442
Testing Your New Component.....	443
Adding Internal Output Variables.....	444
Running Your Component.....	446
Chapter 10: PSCAD Script.....	447
Substitutions.....	447
\$ Value Substitution Prefix Operator.....	448
Valid Context and Key Names.....	450
Local Context Short Form.....	450
Substituting Component Constants into Script.....	451
Substituting Component Constants into Text Labels.....	453
% Data Substitution Prefix Operator.....	454
Substituting Text into Commented Script.....	455
Substituting Text into Text Labels.....	455
{ } Braces.....	456
Anonymous Expression Substitution.....	457
Block Processing.....	457
! Comment Indicator.....	458
Script Directives.....	459
#STORAGE.....	460
#LOCAL.....	461
#FUNCTION.....	462
#SUBROUTINE.....	463
#OUTPUT.....	464
#TRANSFORMERS.....	465
#WINDINGS.....	466
#IF, #ELSEIF, #ELSE, #ENDIF Directives.....	466
~ Line Continuation Operator.....	469
#CASE Conditional Directive.....	470
Expression Evaluation.....	472
Mathematical Functions.....	472

Table of Contents

Chapter 11: Debugging and Finalizing	477
Compiling Projects	477
Common Output Window Messages.....	477
Warning: Suspicious isolated node detected	478
Parser: Suspicious substitution of empty key '<name>'.....	478
Signal '<name>' type conversion may lose accuracy	478
Signal '<name>' source contention -> '<source_1>'	
and '<source_2>'.....	478
Signal '<name>' dimension mismatch -> <dim_1> !=	
<dim_1>	478
Signal '<name>' size mismatch at connection '<con>'.....	479
Array '<Name>' cannot be typecasted	479
Invalid breakout connection to ground at '<Node>'.	
Node array elements cannot be individually grounded.	479
Short in breakout at '<Node>' Node array elements	
must be uniquely defined.	479
Branch imbalance between breakouts at '<Node>'.	
Node array elements cannot be shared between signals. ..	479
Find Feature	480
Find What.....	480
Search.....	480
Viewing Search Results	481
Viewing Build And Data Files	481
Fortran and Data Files	482
Map and Make Files.....	482
Component Ordering.....	483
Showing Sequence Numbers.....	483
Manually Setting Sequence Numbers.....	484
Show Signal Locations	485
Control Signal Flow Indicators.....	486
Virtual Control Wires	487
Using An Integrated Debugger.....	488
Project Options to Preset	488
Linking to the Debugger	489
Creating Library (*.lib) and Object (*.obj) Files.....	491
Object (*.obj) Files.....	491
Library (*.lib) Files	491
Creating a Library (*.lib) File	492
Chapter 12: MATLAB®/Simulink®	495
MATLAB Interface Subroutine.....	495
Arguments.....	496
Inputs	496
Outputs	496

PSCAD

Simulink Interface Subroutine	497
Arguments.....	498
Inputs	498
Chapter 13: Migrating from Older Versions	507
Converting PSCAD V2 Projects	507
User-Written EMTDC Source Code	508
Conversion Issues.....	508
V2 Control Type Components.....	508
V2 Electrical Interface Components.....	508
V2 Component Libraries	509
Component Definitions.....	510
Importing V2 Draft and Runtime Batch Files.....	510
Common Warning and Error Messages.....	513
Importing Additional Runtime Batch Files	513
Importing V2 User Libraries	514
Importing Individual V2 Components	516
Manual Revisions to the New PSCAD V4 Project.....	519
Too Many External Connections	519
Import/Export Tags with Array Variables	519
Runtime Module.....	521
Migrating V2 Cable Systems.....	521
Migrating V2 Transmission Line Systems.....	521
Converting PSCAD V3 Projects	522
Conversion Issues.....	522
System Dynamics Component Ordering.....	522
Additional Libraries and Object Files	522
Flyby Windows.....	523
Node Loop Component Output Format.....	523
MATLAB Libraries and Interface	524
Appendix A: References	525
Index	527

About This Guide

The goal of this guide is to provide a comprehensive reference for installing and using the PSCAD software. It is composed of a combination of both newly written, as well as previously available material from the PSCAD V3 Getting Started Manual and Online Help System.

Information specific to certain models is not included here, but can be found in the new PSCAD V4 Online Help System. The online help system also includes this guide and the EMTDC User's Guide.

For information on more advanced topics regarding simulating using EMTDC, please see the EMTDC User's Guide.

ORGANIZATION

The PSCAD User's Guide is organized in the following manner:

- **Chapter 1: Welcome to PSCAD** briefly introduces the PSCAD product, its use and development, what is new, and also provides important information regarding both compiler and PSCAD restrictions.
- **Chapter 2: Installation and Set-up** describes the installation procedure for all platforms on which PSCAD is supported. It also discusses hardware and software requirements, License Manager and licensing, and provides information on setting up workspace preferences once installation is complete.
- **Chapter 3: Workspace Settings** describes all parameters included within the Workspace Settings dialog window.
- **Chapter 4: The PSCAD Environment** describes the new PSCAD look and feel, including new terminology and features. This chapter also discusses using the Online Help System, and provides a tutorial on creating a new project.
- **Chapter 5: Features and Operations** is essentially a 'how to' guide, which outlines most of the basic features and operations that are available in PSCAD V4. This chapter

also includes a tutorial on creating a new project from scratch.

- **Chapter 6: Online Plotting and Control** provides details on creating and using the new PSCAD V4 online plotting tools. Creating and using the online controls and meters is also discussed.
- **Chapter 7: Project Settings** describes all parameters included within the Project Settings dialog window.
- **Chapter 8: Transmission Lines and Cables** provides details on constructing and defining transmission corridors for both overhead transmission lines, and underground cable systems. Related output files are also discussed.
- **Chapter 9: Designing Components** is essentially a manual in itself, which describes how to design components using the Design Editor. This chapter also includes a tutorial on designing a component from scratch.
- **Chapter 10: PSCAD Script** is a complimentary chapter for Designing Components above. It describes all features and coding conventions involved in the PSCAD Script language.
- **Chapter 11: Debugging and Finalizing** describes common warning and error messages associated with running a project, as well as methods for debugging user code, creating library files and using the search feature.
- **Chapter 12: MATLAB/Simulink** provides details on the interface subroutine and how to set-up the interface and connect with MATLAB/Simulink.
- **Chapter 13: Migrating from Older Versions** outlines all procedures for migrating both V2 and V3 projects into PSCAD V4. Important conversion issues are also discussed for migrating each type of project.
- **Appendix A: References** provides references to technical publications sited in this manual, as well as additional readings of interest.
- **Index:** Alphabetical index of keywords with page numbers.

DOCUMENTATION CONVENTIONS

The following general conventions are followed throughout this manual:

References

References are cited using ‘boxed brackets.’ For example, referring to Reference #5 would appear as [5].

PSCAD

Notes

All note text will appear in text boxes, as shown below:

NOTE text.

Text Boxes

All code examples will appear in text boxes, as shown below:

CODE

Welcome to PSCAD

Whether you are a seasoned user upgrading to PSCAD V4, or you have just recently joined our expanding, worldwide user base - welcome!

PSCAD was first conceptualized in 1988 and began its long evolution as a tool to generate data files for the EMTDC simulation program. In its early form, Version 1 was largely experimental. Nevertheless, it represented a great leap forward in speed and productivity, since users of EMTDC could now draw their systems, rather than creating text listings. PSCAD was first introduced as a commercial product as Version 2 targeted for UNIX platforms in 1994. It arrived as a suite of associated software tools that performed circuit drafting, runtime plotting/control and off-line plotting.

When Version 3 for Windows arrived in 1999, it sought to push the envelope by introducing a simulation system that could be built in a modular form. Systems could now be built up using interconnecting drawing blocks, compiled individually and having their own private data space. This modular system improved accuracy and correctness of the simulation. In addition, Version 3 brought some new usability by fully integrating the drafting and runtime systems of its predecessor. This integration produced an intuitive environment for both design and simulation.

PSCAD Version 4 represents the latest developments in power system simulation software. With much of the simulation engine being fully mature for many years, the new challenges lie in the advancement of the design tools for the user. The goal is to produce software that is both powerful and easy to use. Version 4 retains the strong simulation models of its predecessors while bringing to the table an updated and fresh new look and feel to its windowing and plotting. New single-line representations and new compiler enhancements improve both the accuracy and reliability of the simulation. New editors and easier navigation mean that finding your way and maintaining larger systems is far easier to do. Portability to Version 3 means updating to Version 4 is as simple as point, click, and run. These and many other enhancements are why PSCAD has been, and is, the professional's choice for transients simulation.

WHAT IS PSCAD?

PSCAD (Power Systems CAD) is a powerful and flexible graphical user interface to the world-renowned, EMTDC solution engine. PSCAD enables the user to schematically construct a circuit, run a simulation, analyze the results, and manage the data in a completely integrated, graphical environment. Online plotting functions, controls and meters are also included, so that the user can alter system parameters during a simulation run, and view the results directly.

PSCAD comes complete with a library of pre-programmed and tested models, ranging from simple passive elements and control functions, to more complex models, such as electric machines, FACTS devices, transmission lines and cables. If a particular model does not exist, PSCAD provides the flexibility of building custom models, either by assembling them graphically using existing models, or by utilizing an intuitively designed Design Editor.

The following are some common models found in systems studied using PSCAD:

- Resistors, inductors, capacitors
- Mutually coupled windings, such as transformers
- Frequency dependent transmission lines and cables (including the most accurate time domain line model in the world!)
- Current and voltage sources
- Switches and breakers
- Protection and relaying
- Diodes, thyristors and GTOs
- Analog and digital control functions
- AC and DC machines, exciters, governors, stabilizers and inertial models
- Meters and measuring functions
- Generic DC and AC controls
- HVDC, SVC, and other FACTS controllers
- Wind source, turbines and governors

PSCAD, and its simulation engine EMTDC, have enjoyed close to 30 years of development, inspired by ideas and suggestions by its ever strengthening, worldwide user base. This development philosophy has helped to establish PSCAD as one of the most powerful and intuitive CAD software packages available.

TYPICAL PSCAD STUDIES

The PSCAD users' spectrum includes engineers and scientists from utilities, manufacturers, consultants, research and academic institutions. It is used in planning, operation, design, commissioning, preparing of tender specifications, teaching and research. The following are examples of types of studies routinely conducted using PSCAD:

- Contingency studies of AC networks consisting of rotating machines, exciters, governors, turbines, transformers, transmission lines, cables, and loads
- Relay coordination
- Transformer saturation effects
- Insulation coordination of transformers, breakers and arrestors
- Impulse testing of transformers
- Sub-synchronous resonance (SSR) studies of networks with machines, transmission lines and HVDC systems
- Evaluation of filter design and harmonic analysis
- Control system design and coordination of FACTS and HVDC; including STATCOM, VSC, and cycloconverters
- Optimal design of controller parameters
- Investigation of new circuit and control concepts
- Lightning strikes, faults or breaker operations
- Steep front and fast front studies
- Investigate the pulsing effects of diesel engines and wind turbines on electric networks

WHAT'S NEW IN PSCAD V4?

Many individuals reading this chapter may be current users of PSCAD V3 (or even V2), who have recently upgraded to V4. If so, you are probably eager to start simulating.

This section is provided especially for users of older versions who want a jump-start to proficiently using V4, without the hassle of a lot of reading. Of course, this section will also be of interest to new users, as the new features are described.

Chapter 1: Welcome to PSCAD

The following is a list of some of the major new features and improvements in PSCAD V4:

- **New Windowing Interface:** PSCAD V4 boasts a completely new windowing interface, which includes full MFC (Microsoft Foundation Class) compatibility, docking window support and a new integrated design editor.
- **New Drawing Interface:** The drawing interface has been enhanced to provide uniform messaging and core support, as well as a full double-buffered display.
- **All New Online Plotting Tools:** The online plotting facilities in PSCAD V4 have been completely redesigned and are now more powerful than ever. The new advanced graphs come complete with full features, including full zoom and panning support, marker controls, Polymer, PhasorMeter and XY plotting capabilities.
- **Improved Online Control:** New and improved online user interface controls have replaced those used in V3. The new user controls include the basics as in V3, but feature new functionality, insightful design and stunning appearance.
- **Off-Line Plotting Facilities:** With the inclusion of LiveWire, the best data visualization and analysis software package available today, PSCAD output comes to life. LiveWire includes many of the functions that were available in the PSCAD V2 utility called 'Multiplot.' For more on LiveWire, visit www.zsystems.ca.
- **Component Ordering Control:** Users now have the ability to manually order system dynamics (controls) components. This new diagnostic functionality also includes graphical indication of feed-forward and feedback loops, as well as component sequence numbering. PSCAD will order system dynamics automatically, using a sophisticated algorithm, by default.
- **Control Group Names:** Runtime objects, such as online controls and displays may now be grouped according to group name.
- **Control Signal Flow Indicators:** It is now possible to show control signal flow on data wires. The indicators appear as small arrowheads, pointing in the direction of signal flow.
- **Storage Optimization:** Users now have the option to only store data that is required for the next simulation time step (i.e. feedback signals). Depending on the amount of data signals, this has the potential to greatly increase simulation

speed. In addition, users now have the option to 'turn off' unused Output Channels.

- **Multiple-Trace Curves:** Output channels will now directly accept both REAL and INTEGER arrays. This translates to a multiple-trace curve, which can be inserted directly to graphs.
- **Control Setting Storage:** A new feature has been added, which allows users to store the current setting of Dial, Switch and Slider control components to a template file. Multiple templates can be stored for each Project. This is similar to having different Runtime batch files for a single PSCAD V2 project.
- **Single-Line Diagram Support:** PSCAD now includes the ability to construct circuits in a convenient and space saving single-line format. This new feature includes fully adaptive wires and connections using dimension propagation logic. All three-phase electrical components in the Master Library can be adjusted easily to display a single-line equivalent view.
- **New Smart Subsystem Logic:** Now, electrical subsystems no longer need to be separated into multiple page modules, and electrical nodes on the main page no longer restrict matrix splitting.
- **Enhanced Project Navigation and Organization:** Most plotting and online control display design, as well as navigational operations can now be performed directly from a central project database called the Workspace. Similar to the Project Tree in PSCAD V3, the Workspace provides a wealth of Project related information in a neatly organized tree type environment.
- **Drag and Drop:** PSCAD now supports Drag and Drop functionality for component Instances, as well as display related features, such as Curves and Control Interfaces.
- **New Wires:** Multiple segment Wires are now possible. These new and improved Wires can be manipulated as a group, and may be merged or decomposed at will. Older Wires will be automatically replaced with the new Wire component upon upgrade.
- **New Data Management Interface:** New record support has been added for improved speed and flexibility, along with data export portability to offline analysis tools.
- **New Online Help System:** The Online Help system has been completely redesigned and updated. Based on the

Chapter 1: Welcome to PSCAD

Microsoft HTML Compiled help format, the new system provides a search engine, TOC and index. Also included are the complete PSCAD and EMTDC manual set.

- **New 2D Navigation:** Now not only can you navigate into and out of sub-pages and components, but also view Graphics, Fortran Files, Data Files and signal information, all with the click of the new multi-window tab features. Component definition graphics, parameters and script are also included.
- **Improved Menus and Toolbars:** The most commonly used menu functions have been added as buttons on various new toolbars. Users can now add the most common Master Library components directly using convenient toolbar buttons.
- **MATLAB®/Simulink® Interface:** Now interface PSCAD to both your MATLAB® and/or Simulink® files.
- **Backwards Compatibility:** Both library and case projects created in PSCAD V4, can easily be saved as PSCAD V3 type format. Backwards compatibility to PSCAD V2 is not available.
- **Frequency Scan of a Network:** Users now can get a impedance profile of a network over a frequency range by using Interface to Harmonic Impedance Solution.
- **New Models:** There are many new models and improvements to existing models. See the section entitled New Master Library Models below for a list of available models and features.

V3/V4 Comparison Tables

With PSCAD V4, we have tried our best to incorporate the best features of both PSCAD V2 and V3. However, there are, of course, some differences in appearance, function and terminology. The following tables compare PSCAD V3 with V4, providing a quick and convenient reference.

Appearance and Terminology

There have been some changes to the appearance and terminology used in PSCAD V4. The following is a list of most of them:

PSCAD

Category	PSCAD V3	PSCAD V4
Project Organization	Project Tree	Workspace. The Workspace contains information on loaded projects. The organizational structure of information in each individual project is now referred to as the project tree for that particular project.
Messaging	Message Tree	Output Window. This window also categorizes Build and Runtime messages using the new 2D navigational tab features, as well as providing search results.
Graphical View	Draft window	Design Editor - Circuit Tab. Imbedded graphical display editor.
Viewing Fortran	Fortran file viewed in text editor	Design Editor - Fortran Tab. Imbedded text viewer.
Viewing Electrical Data	Data file viewed in text editor	Design Editor - Data Tab. Imbedded text viewer.
Component Design	Component Workshop	Design Editor - Graphic, Parameters and Script Tabs. In V3, component design was performed in a separate utility called the Component Workshop. Access to component definitions is now included directly into the Design Editor window, using the new tab features.
Component Workshop: Graphics	Graphics category in Component Workshop	Design Editor - Graphic Tab. Imbedded graphical display editor.

Chapter 1: Welcome to PSCAD

Component Workshop: Parameters	Parameters category in Component Workshop	Design Editor - Parameters Tab. Imbedded text editor
Component Workshop: SectionsNode	Code section category in Component Workshop	Design Editor - Script Tab. Imbedded text editor.

Keyboard Shortcuts

There have been some minor changes to the keyboard shortcuts (especially plotting related shortcuts) from that used in PSCAD V3. The following tables summarize these changes:

General:

Shortcut	PSCAD V3	PSCAD V4
Dynamic Scroll	Ctrl + left mouse hold	Ctrl + Shift + left mouse hold. This change was made in order to avoid conflicts with the new online plotting tools.
Iconify Plot	I	No longer used. See Frame/ Panel Minimization for more details.

Plotting:

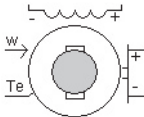
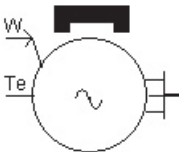
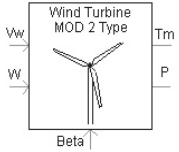
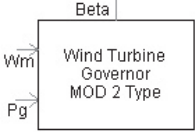
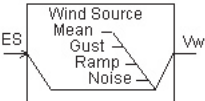
Shortcut	PSCAD V3	PSCAD V4
Zoom to Time Frame	Z	Ctrl + left mouse hold
Zoom Previous	Shift + Z	P
Show Cross Hairs	X	C
Zoom Actual	A	No longer used
Zoom Smart	S	Y
Zoom to Default	D	L
Cut Graph	Ctrl + X	No longer used

PSCAD

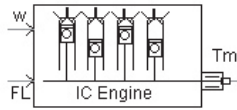
For a complete list of PSCAD keyboard shortcuts, see Keyboard Shortcuts.

New Master Library Models

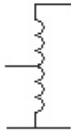
Many new models have been added to the PSCAD Master Library since the last release of PSCAD V3. The following summary provides a picture and a quick description of each new model. For more details on these components, see the PSCAD Online Help system.

Graphic	Description
 <p>The graphic shows a circular cross-section of a machine with a central shaded area. It has two electrical terminals on the right side, labeled with '+' and '-' signs. On the left side, there are two input ports labeled 'w' and 'Te'. A wavy line is drawn above the circle, representing a magnetic field or flux.</p>	Two Winding DC Machine
 <p>The graphic shows a circular cross-section of a machine with a wavy line inside, representing a permanent magnet. It has two electrical terminals on the right side. On the left side, there are two input ports labeled 'W' and 'Te'. A thick black bar is positioned above the circle, representing a permanent magnet.</p>	Permanent Magnet Synchronous Machine
 <p>The graphic shows a rectangular block representing a wind turbine. Inside the block is a stylized drawing of a three-bladed wind turbine. The text 'Wind Turbine MOD 2 Type' is written at the top of the block. On the left side, there are two input ports labeled 'Ww' and 'W'. On the right side, there are two output ports labeled 'Tm' and 'P'. At the bottom, there is an input port labeled 'Beta' with an upward-pointing arrow.</p>	Wind Turbine
 <p>The graphic shows a rectangular block representing a wind turbine governor. The text 'Wind Turbine Governor MOD 2 Type' is written inside the block. On the left side, there are two input ports labeled 'Wm' and 'Pg'. At the top, there is an input port labeled 'Beta' with a downward-pointing arrow.</p>	Wind Governor
 <p>The graphic shows a rectangular block representing a wind source. The text 'Wind Source' is written at the top of the block. Inside the block, there are four sub-labels: 'Mean', 'Gust', 'Ramp', and 'Noise'. On the left side, there is an input port labeled 'ES'. On the right side, there is an output port labeled 'Ww'. A line graph is shown inside the block, illustrating the characteristics of the wind source.</p>	Wind Source

Chapter 1: Welcome to PSCAD



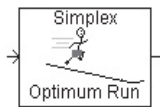
Internal Combustion Engine



1-Phase Auto Transformer



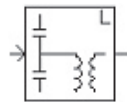
3-Phase Star-Star Auto Transformer



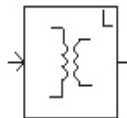
Optimum Run



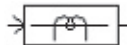
Radio Links (“Global Variables”)



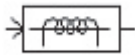
Coupled Capacitor Voltage Transformer (CCVT)



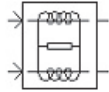
Potential Transformer (PT/VT)



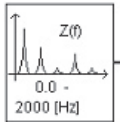
Current Transformer (CT) - Lucas Model



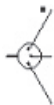
Current Transformer (CT) - JA Model



Two CT Differential Configuration - JA Model



Interface to Harmonic Impedance Solution



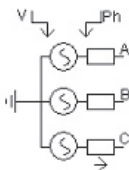
3-Phase View / Single-Line View Electrical Wire Converter (Breakout)



6-Dimension / 2 x 3-Dimension Electrical Wire Converter



Capacitive, Inductive and Resistive 3-Phase Passive Loads



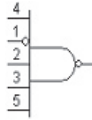
Voltage Source Model 3



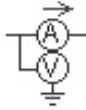
Array Product and Array Sum



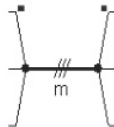
Chapter 1: Welcome to PSCAD



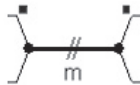
Multiple Input Logic Gate



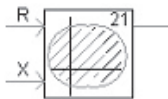
Multimeter



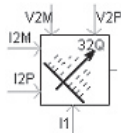
Mutually Coupled Three Wire Bus



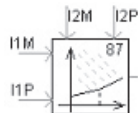
Mutually Coupled Two Wire Bus



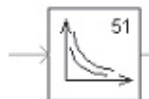
Distance Relay Element - Apple Characteristics



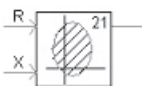
Negative Sequence Directional Element



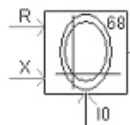
Dual Slope Current Differential Relay



Inverse Time Over Current Relay



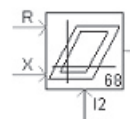
Distance Relay Element - Lens Characteristics



Out of Step Relay - Lens Characteristics



Out of Step Relay - Mho Characteristics



Out of Step Relay - Polygon Characteristics

DIMENSION LIMITS

Any limitations imposed on your simulation projects will depend mainly on which PSCAD edition you are using, and which Fortran compiler you have chosen. The following tables list the most commonly encountered dimension limitations.

PSCAD Imposed Dimensions (all Editions and Compilers)

Description	Limit
Number of Transmission Line or Cable conductors per right-of-way	20
Page Modules	1,024
Components	65,536
Output Channels (PGBs)	1,024

Chapter 1: Welcome to PSCAD

Edition Specific Dimensions

Description	Student	Educational	Professional
Electrical Subsystems	1	1	Unlimited
Electrical Nodes	15	200	Unlimited
Page Modules	5	64	1,024
Components	32,768	32,768	65,536

Compiler Specific Dimensions

Description	EGCS/GNU Fortran 77 Compiler	Fortran 90/95 Compilers
Electrical Subsystems	10	Unlimited
Electrical Nodes	200 / Subsystem	Unlimited
Electrical Branches	2,000 / Subsystem	Unlimited
Transformers	70	Unlimited
Mutually Coupled Windings	10	Unlimited
Distributed Lines and/or Cables	50	Unlimited
Frequency Dependent (Phase) Transmission Lines and Cables	20	Unlimited
Output Channels (PGBs)	1,030	Unlimited
Radio Link Transmitters	50	Unlimited
Radio Link Receivers	500	Unlimited
STOR Storage Allocations	50,000	Unlimited
STORF//L/C Storage Allocations	10,000 Each	Unlimited

PSCAD COMMUNITY FORUM

On August 18, 2004, the Manitoba HVDC Research Centre unleashed the PSCAD Community Forum on the online world. This forum is an online environment where users of PSCAD (and other related products) from around the world can meet and discuss the issues important to them. The forum includes several separate topic areas, and allows for the exchange of files through post attachments. Users may contribute to a growing list of example case projects, and show off what they have accomplished with PSCAD and/or other related software.

Please note that there are other regional users groups as well: Please contact PSCAD Technical Support (support@pscad.com) to find out if your region has one.

Membership

To become a member of the PSCAD Community Forum, you must have a passion for PSCAD/EMTDC! To add your name to the forum's worldwide member list, simply access the site (details below) and register online. There is no membership fee.

Forum Details

Users can submit their questions/comments/examples regarding PSCAD or related software by posting under the appropriate forum. Your post will immediately be viewable to all registered members. If you are responding to an existing topic, the forum will automatically send an email to the member who posted the topic.

PSCAD Community Forum: <http://bb.pscad.com>

NOTE: The PSCAD Community Forum should not be used as a support medium. If you have queries regarding problems simulating with PSCAD, bug reports, etc., please contact the PSCAD Support desk at support@pscad.com. Contacting the support desk will ensure a prompt response to your query.

CONTACTING US

The Manitoba HVDC Research Centre Inc. and its representatives are committed to providing you with the best sales and technical support available.

Contact your local PSCAD representative first for fast and efficient service. If you do not have their contact address from the time you purchased PSCAD, you can get it either from the PSCAD Web Site or by contacting the Centre directly. To make the best use of our technical support and sales facilities, you should have a maintenance contract arranged through your local PSCAD supplier.

PSCAD Support Services

For non-sales related technical support, precedence is given to commercial users with valid software maintenance contracts. We can be reached at:

E-mail:	support@pscad.com
Phone:	+1 (204) 989-1240
Fax:	+1 (204) 453-5074
Web Site:	www.pscad.com

Address:	PSCAD Support Services Manitoba HVDC Research Centre Inc. 244 Cree Crescent Winnipeg, Manitoba R3J 3W1 Canada
----------	---

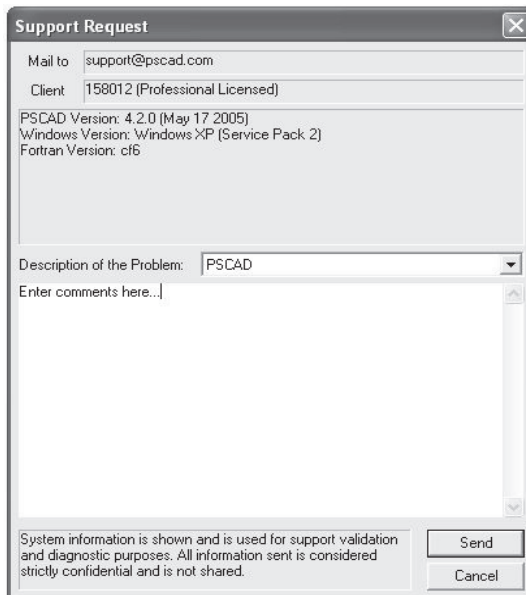
Support Petition Request

In order to provide faster, more efficient support service, we recommend that users issue a Support Petition Request directly from PSCAD, rather than e-mailing support directly. A Support Petition Request will automatically contain information regarding your PSCAD version, compiler version and license data, appended to the request. This will minimize the need for 'back-and-forth' communications, which can be cumbersome, especially for our overseas users.

To send a support query, go to the Main Menu bar and select **Help | Issue Request...**



This will bring-up the Support Request dialog window. Simply add your comments in the space provided. Please ensure that you select a **Description of the Problem** from the drop list, which most adequately describes your problem type. When finished, press the **Send** button. An email will be sent directly to the PSCAD Support Desk.



Chapter 1: Welcome to PSCAD

PSCAD Sales

For sales related inquiries (i.e. quotations, etc.), we can be reached at:

E-mail: sales@pscad.com

Phone: +1 (204) 989-1240

Fax: +1 (204) 453-5074

Web Site: www.pscad.com

Address: PSCAD Sales
Manitoba HVDC Research Centre Inc.
244 Cree Crescent
Winnipeg, Manitoba R3J 3W1
Canada

Installation and Setup

HARDWARE AND SOFTWARE REQUIREMENTS

The following are the recommended specifications.

Category	Minimum	Recommended
Processor	500 MHz (Pentium 2)	3 GHz (Pentium 4 or Athlon)
Operating System	Windows 2000 Windows XP	Windows 2000 Windows XP Pro SP2
Additional software	Digital Visual Fortran 5 Compaq Visual Fortran 6.x NOTE: A free EGCS/GNU Fortran 77 compiler is provided, which is sufficient to run most cases in PSCAD. However, certain dimensioning limits will be imposed (see Dimension Limits).	Intel Visual Fortran 9.0.x NOTE: One of the following packages must be installed for IVF 9.0 usage: • Microsoft Visual C++ .NET 2002 or 2003. • Microsoft Visual Studio.NET 2002 or 2003 (with Visual C++ installed).
Memory (RAM)	256 MB	1 GB
Hard Disk	500 MB	40 GB (Fast SCSI or IDE)
Video	SVGA (800 x 600)	XGA (1280 x 1024)
Other Peripherals and Hardware	CD-ROM Drive /w 32-bit Drivers Mouse One (1) Parallel, Serial or USB port (for hardware lock) TCP/IP Network Protocol	Same

FORTRAN COMPILERS

PSCAD requires a Fortran compiler to build and simulate projects. The following commercially available compilers are presently supported:

- **Digital Fortran 5.0**
- **Compaq Visual Fortran 6.x**
- **Intel Visual Fortran 9.0.x**

For your convenience, a free Fortran compiler, called the EGCS/ GNU Fortran 77 compiler, is provided on your PSCAD CD (or may be downloaded from our worldwide website: www.pscad.com). If you have downloaded the Student Edition from our website, the GNU compiler is available as a separate file. Please note that the GNU compiler will impose some limitations otherwise not experienced with the Digital, Compaq or Intel Fortran compilers.

If you are the owner of a professional PSCAD license, and intend to run large cases (i.e. greater than 200 nodes), it is highly recommended that you purchase the Intel Visual Fortran 9.0.x compiler. Contact us at PSCAD Sales (sales@pscad.com) for purchasing details.

LICENSING

All PSCAD software editions, except the Student Edition, must be licensed. Licensing is organized and controlled using License Manager software, which can exist either as a separate, standalone program, or can be embedded within PSCAD itself.

The type of licensing arrangement that is best for you depends on how you wish to utilize PSCAD, and on what type of license you purchased. The following sections describe each type of licensing arrangement in detail.

Multi-User Licensing

The multi-user licensing (MUL) method was first introduced with PSCAD V3, and was then the only licensing option available. With this configuration, the License Manager software is installed as a standalone program, which can be accessed by any computer (including the host), on a Local Area Network (LAN).

PSCAD

NOTE: The License Manager will only grant licenses to computers on the same subnet. Licenses requested via a router will not be accepted.

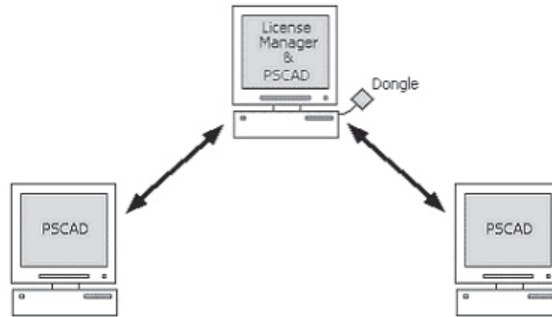
A multi-user set-up is capable of providing multiple users on multiple machines the ability to run multiple combinations of various licensable editions of PSCAD. In other words, if a user is sufficiently licensed, this method has the potential to afford a lot of freedom when operating over a LAN, and should be used if there will be several PSCAD users.

In multi-user licensing, the License Manager is used in conjunction with a hardware lock (also known as a dongle), which contains licensing information for the purpose of validation. When an instance of PSCAD is started somewhere on the LAN, a license will be requested from the License Manager. The License Manager determines whether a license is available, and checks the information on the dongle for verification.

The number of users that can access PSCAD simultaneously over the LAN depends on the number of licenses available on the dongle. For example, if there are a total of two Professional Edition licenses on a network dongle, then only two users may use the PSCAD Professional Edition at the same time. Each time a user opens or closes the PSCAD program, it will request or relinquish a license respectively. Each licensable edition of PSCAD possesses a separate product number, so that several products are possible on one dongle.

NOTE: The number of licenses owned does not limit the number of PSCAD installations on the LAN!

The License Manager and dongle needs to be installed on only one computer in a network. This computer will act as the License Manager 'server,' which will hand out licenses to other machines on the network, as requested. PSCAD may also be installed on the License Manager server machine if desired.



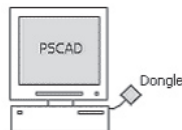
EXAMPLE: A PSCAD user owns three Professional Edition licenses programmed onto one dongle. The above diagram illustrates how the user could install the License Manager on a LAN, so that up to three users can run PSCAD simultaneously.

Multi-user licenses can be time limited or infinite (no time limit).

Single-User Licensing

Single-user licensing (SUL) still involves the use of a dongle, but a standalone License Manager program is not required. Here, an embedded License Manager, within the PSCAD program itself, manages the licenses.

A single-user license will only allow a single user on a single machine to run only one instance of a licensable edition of PSCAD. Single-user licensing is designed especially for use on standalone machines and laptops, and will NOT allow access from other machines, even if it is connected to a LAN.



EXAMPLE: A PSCAD user owns one Professional Edition license programmed onto a single dongle. The above diagram illustrates how the user would install PSCAD as a single-user configuration. Note the absence of the License Manager as a separate program.

PSCAD

Single-user licenses can be time limited or infinite (no time limit).

Trial Licensing

In order to simplify the evaluation process, and to provide access to the full PSCAD Professional Edition functions, trial licensing is now available with PSCAD V4. A trial license is a special type of single-user license, where a time limit is mandatory and the user may only own a single trial license of the PSCAD Professional Edition.

NOTE: Only the Professional Edition of PSCAD is available for trial licensing.

With trial licensing, neither the License Manager nor the dongle is required. However, the user must request a trial license after PSCAD is installed. Note that until the trial license is approved and sent by the Manitoba HVDC Research Centre Inc., the installed copy of PSCAD will function as a Student Edition (SE), with all SE limitations imposed (i.e. 15 electrical nodes maximum).

NOTE: Do not upgrade your operating system, hard drive, or network card during the duration of the trial license period. Any one of these changes will cause your trial license to be lost.

Trial licenses are locked to the hardware and registry of a particular machine and are therefore not transferable from one machine to another. The latest installed trial license will overwrite any previously existing trial licenses.

Hardware Locks (Dongles)

The License Manager software supports the following types of hardware locks:

Chapter 2: Installation and Setup



Sentinal SuperPro™ 797, Low Profile Parallel Port Hardware Lock



Sentinal SuperPro™ USB, USB Port Hardware Lock



MicroSentinal™, 25-Pin Serial Port Hardware Lock with 25 to 9-Pin Adapter

The USB and parallel port dongles are the preferred choice, due to their small size. The parallel port dongle is perfect for laptop computers, as it features a low profile and does not protrude from the back of the machine (and typically does not ever need to be removed). The serial and parallel port locks are compatible with all Windows platforms on which PSCAD is supported. However, **the USB dongle is NOT compatible** with the following operating systems:

- Windows NT 4.0
- Windows 95 OEM Service Release 2 or earlier

NOTE: Not all computers come with USB ports. If you are not sure if you have one, contact your system administrator for help.

License Key

The license key is an ASCII text file named 'license.txt,' which usually resides on a floppy disk (called 'License Manager: License.txt') required for the PSCAD Professional and Educational Editions. This file contains encrypted information designed to act as a 'Key' to adding or upgrading licenses (both single-user and standalone) to the license database file, which is used directly by the License Manager.

LICENSE MANAGER SOFTWARE

In addition to the standalone License Manager software that was first introduced with PSCAD V3, the PSCAD package now includes two additional types of License Manager. The type of License Manager you will need depends on your licensing arrangement.

There are a few important points to discuss regarding the maintenance and operation of the License Manager. These are outlined in the following sections as well.

Standalone License Manager

The Standalone License Manager (SLM) is used only with multi-user licenses. The SLM can operate on a dedicated server, or on any machine running PSCAD. Any combination of PSCAD products can be included and used interchangeably with this License Manager. Installation instructions for the SLM are given later in this chapter.

Maintenance and Support

Once the License Manager is installed and running, it is a good idea for the person operating the designated License Manager server to maintain the software and ensure proper operation.

The Standalone License Manager (SLM) usually starts automatically when the machine is booted (unless manual starting is specified during installation) and runs as a background process. All SLM events and transactions are recorded in a log file entitled 'Lmgrd-log.txt,' which is usually located in the Windows or WINNT directory (depending on your OS). When troubleshooting, this file contains important information about the cause of any problems.

If problems arise with the operation of your License Manager software, first try to address the problem using the clues found in the log file. If the solution remains elusive, send us an email at PSCAD Support Services (support@pscad.com) with a detailed description of the problem. To ensure efficient response time, please attach the log file (i.e. **Lmgrd-log.txt**). Also, go to **START | Programs | HVDC Lmgr** and run the 'Get License Info' utility and attach the generated 'Get License Info' file (**getinfo.txt**) to the email.

Local License Manager

The Local License Manager (LLM) is utilized only when running under a single-user license. Although the LLM will support multiple

Chapter 2: Installation and Setup

product licenses, it will not support multiple instances of any individual PSCAD product. Attempting to start a second instance of a certain product (say Professional Edition), will force the second instance to run as the Student Edition.

Trial License Manager

The Trial License Manager (TLM) is utilized only when running under a lockless trial license. The TLM supports only a single, time limited license of the Professional Edition (multiple trial licenses are not supported). As with the Local License Manager, the TLM will not grant licenses to other machines on the LAN, and only one instance of a trial license may be active at one time. Any additional instances of PSCAD will run as the Student Edition.

PSCAD Usage Log

As of License Manager Version 1.24, the usage of licensed PSCAD editions, such as PSCAD Professional and PSCAD Educational, is now logged. This information is saved in a comma-separated variable (*.csv) file called PscadUsage.txt, which is located in either the Windows or WinNT directory on the machine running the License Manager software. This file can be directly imported into Microsoft Excel.

The following is a sample line from a PscadUsage.txt file:

```
lha, Pro, Remote, 7327, 177, Mon Aug 22 10:45:42 2005, Mon Aug 22 10:48:39 2005  
lha, Pro, Remote, 7328, 176, Mon Aug 22 10:46:21 2005, Mon Aug 22 10:49:17 2005
```

The format of the file is as follows:

<user id>, <Pscad Edition>, <user location>, <LicenseID><#second
s of use>, <start time>, <finish time>

Description:

- <userid>: This is the login userid of the PSCAD user.
- <Pscad Edition>: Can be 'Pro' or 'Edu.' Use of PSCAD Student is not logged as it is not a licensed edition and does not require the LM.
- <user location>: 'Remote' for users using PSCAD on machines other than the LM machine. 'Local' for users using PSCAD on the same machine as the LM machine.

PSCAD

- <LicenseID>: This is the license ID of the granted license, and appears in the LM log file (lmgrd-log.txt), and in the PSCAD V4 log file (PscadLmgr.txt).
- <start time>: The time that the LM granted the PSCAD license to the user.
- <finish time>: The time that the user quit the PSCAD application.

INSTALLING V4 FOR THE FIRST TIME

This section is exclusively for those who are either installing PSCAD V4 for the first time, or installing PSCAD V4 for the first time on a particular PC.

If you currently have PSCAD V3 installed on your PC, installing PSCAD V4 will not affect it - V4 is installed in a different program directory. However, your V3 License Manager software (if present) will be upgraded during your V4 installation. This will not affect your PSCAD V3 operations, and is recommended anyway.

If you have a previous version of PSCAD V4 installed already on your machine, see 'Upgrading Your V4 Software' in this chapter.

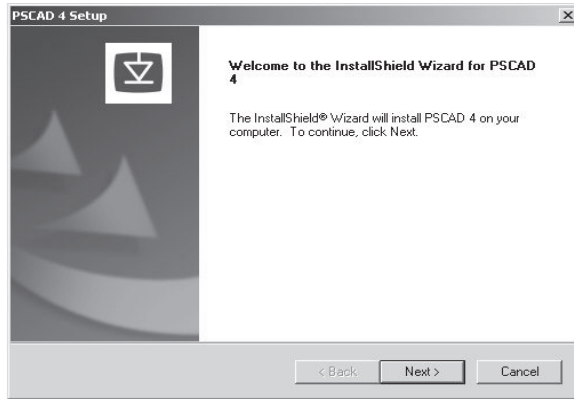
Installing PSCAD

If your operating system is Windows NT4, XP or 2000, you must have administrator or power user access to your machine, in order to install PSCAD.

Please perform the following steps:

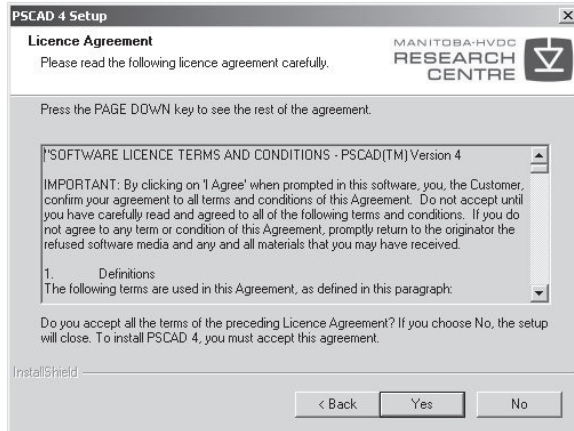
1. Insert the PSCAD CD into your CD-ROM drive.
2. The installer should start automatically and in a few moments, you should see the following screen:

Chapter 2: Installation and Setup



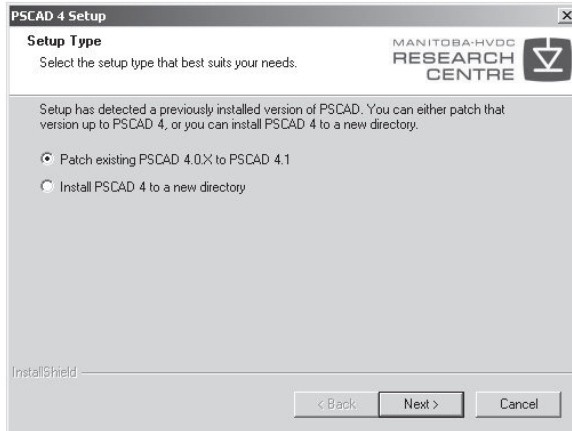
NOTE: If a different window appears asking if you want to Modify, Repair or Remove existing installed components, skip to 'Upgrading Your V4 Software' in this chapter.

3. Click **Next**. A dialog as shown below should appear outlining the PSCAD License Agreement. Once the entire agreement has been read, click **Yes** to continue.



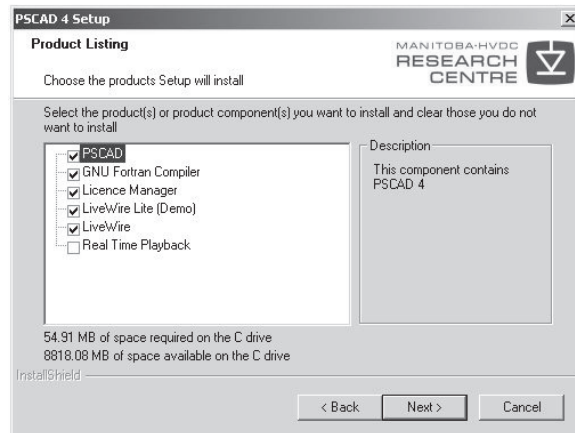
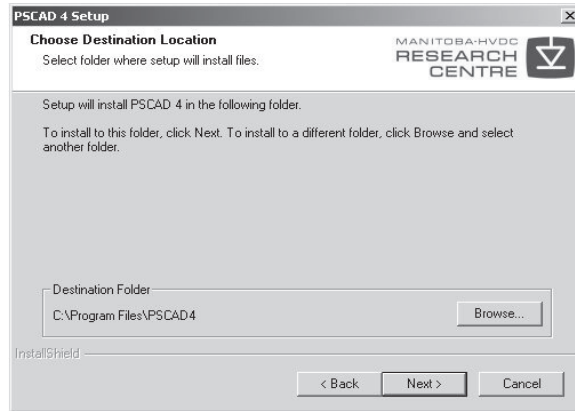
If you are installing a patch from a CD, and a previous version of PSCAD is detected, the following dialog should appear. See the notes given below.

NOTE: The PSCAD version numbers on this dialog may vary according to what version you are upgrading to/from.



- **Patch existing PSCAD 4.x.x to PSCAD 4.x.x:** Selecting this option will patch your existing installation. Files pertaining to PSCAD v4.0.x will be overwritten with the latest upgrades.
 - **Install PSCAD 4 to new directory:** Select this option to fully install PSCAD v4.1.x into a new directory. This is useful if you wish to keep the older version in order to compare results.
4. You have the option of installing PSCAD anywhere on your system. The dialog box below will allow you to make any changes you desire for installation location. When you have finished selecting a destination for PSCAD, click **Next**.

Chapter 2: Installation and Setup



5. You will be presented with the above dialog, on which options are available for your install. Explanations for each of these options are given below.
 - **PSCAD**: Select this option to install PSCAD. Although all PSCAD editions (i.e. Student, Educational and Professional) are included on the CD, you must be properly licensed to use any edition other than the Student Edition.
 - **GNU Fortran Compiler**: If you do not currently have either the Digital or the Compaq Fortran 90 compiler installed, select the GNU Fortran compiler. A Fortran compiler is **required** for any PSCAD project to simulate (run).
 - **License Manager**: Only select to install the License Manager if you intend for this machine to be the License

Manager server on a network (Standalone Licensing). If you choose to install the License Manager, proceed to Step 8.

- **LiveWire Lite (Demo):** Select this option to install the evaluation version of LiveWire. LiveWire is the primary data analysis and post-processing software tool recommended for use with PSCAD. You can get more information on this software at www.zsystems.ca/products.htm.
- **LiveWire:** Select this option to install the full version of LiveWire. Note that you must be licensed to run the full version. Please contact us at sales@pscad.com for more details on obtaining a full license.
- **Real Time Playback:** If you wish to evaluate the Real Time Playback (RTP) demo software (or install the commercial version), select this option. For more on the RTP, see our website at: www.pscad.com.

NOTE: If you change compilers at a later date, you can change the compiler options directly in PSCAD without having to re-install.

If you selected only **PSCAD** to be installed above in **Step 5**, then the installer will ask you to provide some additional information about how your current Licensing is configured:

Select PSCAD License Management Options

I do not need a standalone Licence Manager

I will only be using

- the Student Edition, or
- Lockless trial licences.

I will only be using

- Single-user/single-machine licences.

I will use a Licence Manager on this machine

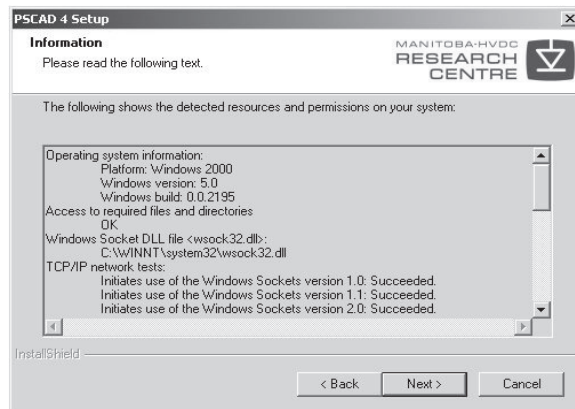
I will use a Licence Manager on another machine on my network

Machine name: (or IP address)

OK

Chapter 2: Installation and Setup

- If you are a Student Edition user, will be using a Single-User License, or intend to request a Trial License (time limited), then you do not require a Standalone License Manager. Select the appropriate choice within the **I do not need a standalone License Manager** section of the above dialog.
 - If you did not select to install the Standalone License Manager in **Step 5**, but have changed your mind, select **I will use a License Manager on this machine**.
 - If this machine will be contacting a remote Standalone License Manager, select **I will use a License Manager on another machine on my network**. If this option is selected, you must then enter the hostname of the machine, on which the License Manager resides. For example, if this machine is named 'zeus,' then you should enter this name as shown above. The Installer will also accept an IP address. The hostname CANNOT contain spaces!
6. The Installer will now perform a variety of tests to verify system settings and ensure that everything is OK to proceed. If you would like to see the results of these diagnostics, please have a look at the report dialog window (similar to that shown below). Click the **Next** button to continue.



7. You will then be asked to enter a program folder, in which to place all of the program icons. A folder called PSCAD is selected by default. Click **Next** when finished.

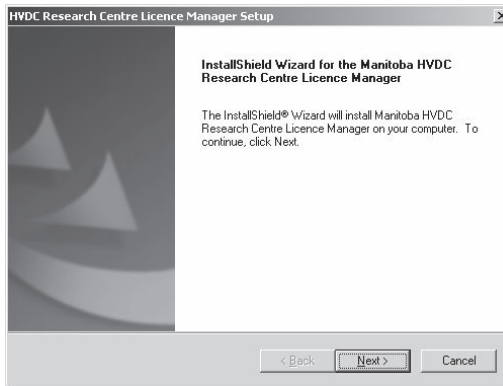
PSCAD

At this point, the Installer will install PSCAD, and then branch off into additional installations that were indicated in Step 5. If you did not select the License Manager in **Step 5**, proceed to the section entitled Installing EGCS/GNU Fortran.

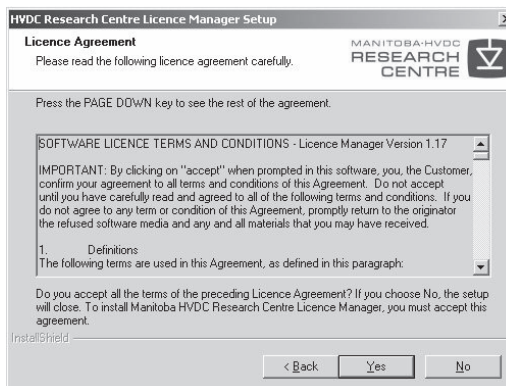
Installing the License Manager

The License Manager Installer will only run if you selected it in Step 5. If you did not, proceed to the section entitled Installing EGCS/ GNU Fortran.

8. You will be presented with the following dialog window. Click **Next**.

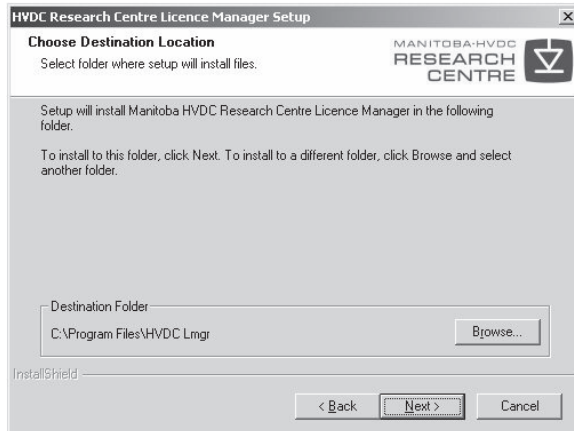
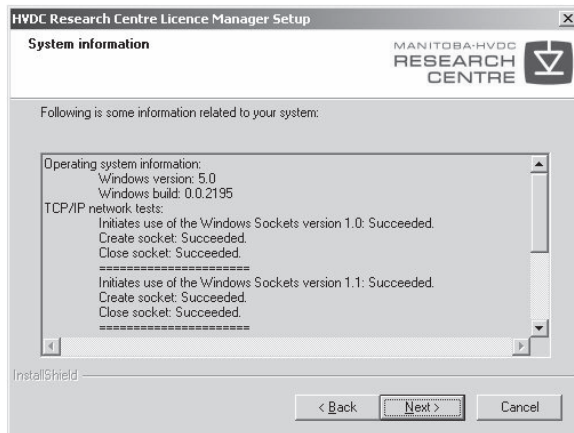


9. A dialog, as shown below, should appear outlining the License Manager License Agreement. Once the entire agreement has been read, click **Yes** to continue.



Chapter 2: Installation and Setup

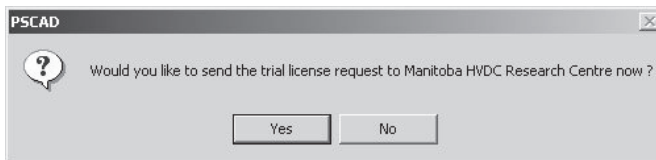
10. A window should pop-up with some pre-installation information. Please review and click **Next**.
11. The Installer will now perform a variety of tests to verify system settings and ensure that everything is OK to proceed. If you would like to see the results of these diagnostics, please have a look at System Information dialog window (similar to that shown below) once the tests are completed. Click **Next**.



12. You have the option of installing the License Manager anywhere on your system. The dialog above will allow you to make any changes that you require for installation location. When you have finished selecting a destination, click **Next**.
13. You will then be asked to enter a program folder, in which to place all of the License Manager related program icons. A folder called HVDC Lmgr is selected by default. Click **Next** when finished.

Windows 98/ME Only:

14. The installer will ask 'Would you like to start the License Manager during machine boot?'. Selecting **Yes** means that the License Manager will start automatically when the system is re-booted. If **No** is selected, the License Manager must be manually started following each machine reboot. See 'Manually Starting the License Manager' in this chapter for more details.



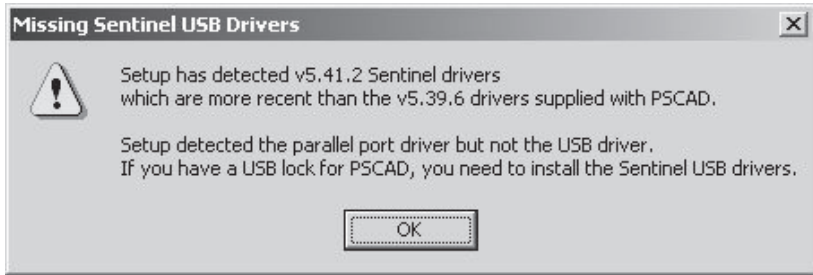
15. Another window will pop-up asking to simply verify the chosen program folder and installation location. Click **Next**.
16. At this point, the License Manager software will be installed along with the proper drivers required for the supported hardware locks. Also, a new service will be added to the Windows Services list. The Installer will run checks to ensure if the required drivers already exist, and if so, is an upgrade required.

Possible Messages

Missing USB Driver:

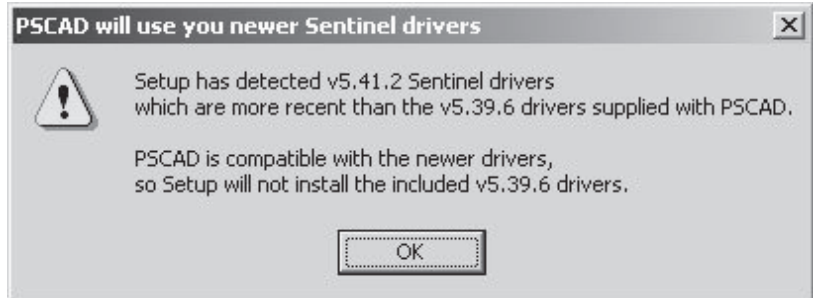
If it is found that your PC has a more recent Sentinel parallel port driver installed, but is missing a USB driver, the following message will be issued. Click **OK**.

Chapter 2: Installation and Setup

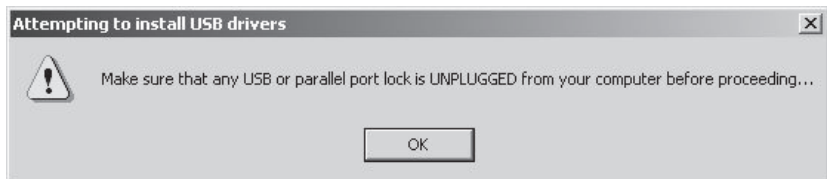


More Recent Drivers Detected:

If it is found that your PC has more recent Sentinel USB and parallel port drivers already installed than those that come with the PSCAD installation, the following message will be issued. Click **OK**.



17. Before the hardware lock drivers are installed, you should receive an important warning message similar to the following. Click **OK** once you have unplugged your dongle.



18. If installation of the hardware lock drivers is successful, the following message will appear. Take note of the specified directory path and click **OK**.



19. The Installer will now ask you to connect your dongle as shown below. If you are unsure where your USB, parallel or serial ports are located, contact your system administrator. Also, see the notes given below.



IMPORTANT:

A. Parallel Port Dongle:

- **DO NOT** insert the Parallel Port Dongle into the 25-pin serial port (if it exists).
- The dongle and the floppy disk have serial numbers (SN#) on them. They **MUST** match to be installed on the same machine.

B. USB Dongle:

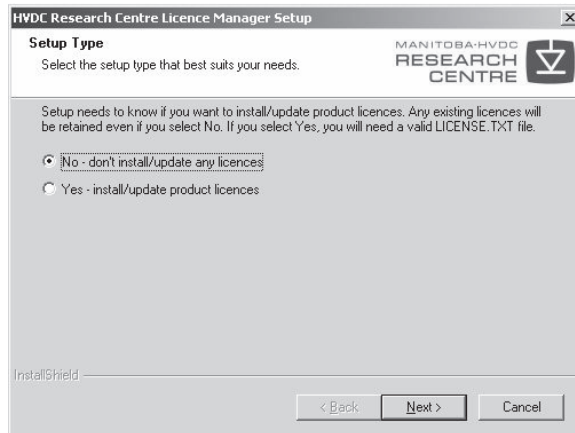
- If you are using a USB dongle, it is possible that the Windows 'Find New Hardware Wizard' may appear when you connect your dongle. If so, it will ask where the driver for the new device is. Enter the path given in the information dialog shown in Step 18. Otherwise, incorrect default drivers may be installed.
- The dongle and the floppy disk have serial numbers (SN#) on them. They **MUST** match to be installed on the same machine.

Chapter 2: Installation and Setup

C. Serial Port Dongle:

- If you are using a Serial Port dongle, connect the dongle to the 25-pin serial port. If you have specified the 25-pin serial port dongle during purchase, then a 9-pin to 25-pin converter will have been included with the PSCAD package (in case you do not have a 25-pin serial port).
- The dongle should be inserted in the direction indicated on the dongle.
- **DO NOT** insert the 25-pin serial port dongle into the parallel port.
- The dongle and the floppy disk have serial numbers (SN#) on them. They **MUST** match to be installed on the same machine.

20. You should now be presented with the following dialog window.



As this is the first time you are installing PSCAD V4 and the License Manager will require a new product license - select **Yes - install/update product licenses**. If this is not your first installation of the License Manager or you prefer not to install or update your license, select **No - don't install/update product licenses**. See the notes below if you are not sure. Click **Next**. If **No - don't install/update product licenses** is selected, proceed to Step 24.

PSCAD

NOTE:

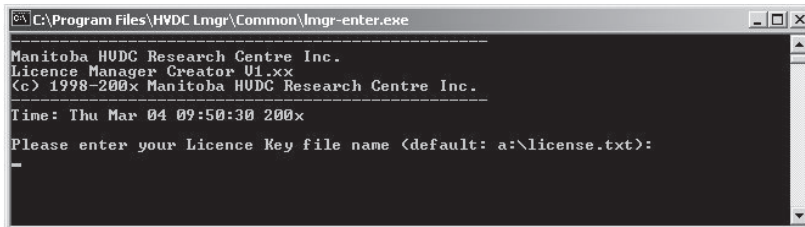
Select **Yes**:

- If this is the first time you are installing the License Manager on this machine, or
- If you are adding a new license (V4 for instance) to an existing license database file, or
- If you are updating or modifying a license, which is already in the existing license database file.

Select **No**:

- If you have no new licenses to add to the license database file, or
- If you do not wish to update or modify any licenses already in your license database file.

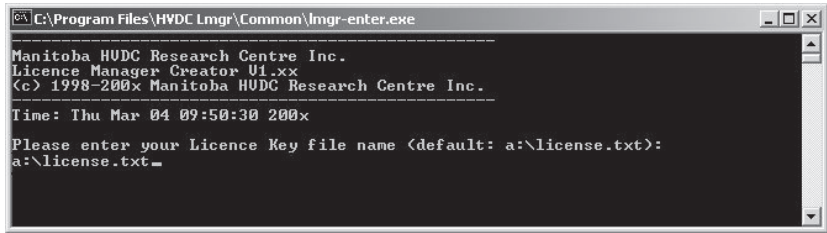
21. The Installer will then invoke the 'Enter License Key' program, and a Command Prompt window should appear asking you to enter the License Key File Name. This file is located on the floppy disk provided with PSCAD, entitled 'License Manager: License.txt'.



```
C:\Program Files\HVDC Lmgr\Common\lmgr-enter.exe
-----
Manitoba HUDC Research Centre Inc.
Licence Manager Creator U1.xx
(c) 1998-200x Manitoba HUDC Research Centre Inc.
-----
Time: Thu Mar 04 09:50:30 200x
Please enter your Licence Key file name (default: a:\license.txt):
_
```

22. Place the License Floppy Disk into your floppy drive and enter '**a:\license.txt**,' as shown below, where '**a**' represents the drive letter. If your License File is located at another location, such as on the network, enter the absolute path to the file or contact your system administrator for help.

Chapter 2: Installation and Setup



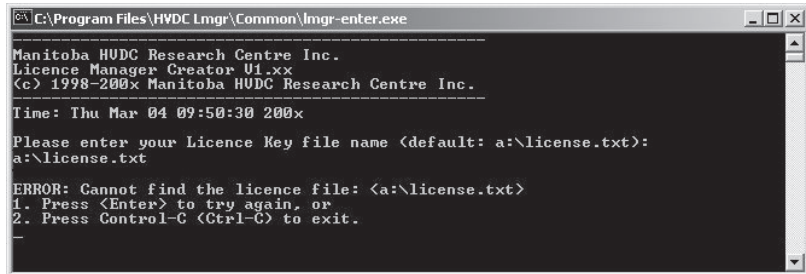
```
C:\Program Files\HVDC Lmgr\Common\lmgr-enter.exe
-----
Manitoba HVDC Research Centre Inc.
Licence Manager Creator V1.xx
(c) 1998-200x Manitoba HVDC Research Centre Inc.
-----
Time: Thu Mar 04 09:50:30 200x
Please enter your Licence Key file name <default: a:\license.txt>:
a:\license.txt_
```

Press the **Enter** key when you are finished.

Troubleshooting

License File Cannot be Found:

If the 'Enter License Key' program is unable to find the license file, you may receive the following message:



```
C:\Program Files\HVDC Lmgr\Common\lmgr-enter.exe
-----
Manitoba HVDC Research Centre Inc.
Licence Manager Creator V1.xx
(c) 1998-200x Manitoba HVDC Research Centre Inc.
-----
Time: Thu Mar 04 09:50:30 200x
Please enter your Licence Key file name <default: a:\license.txt>:
a:\license.txt
ERROR: Cannot find the licence file: <a:\license.txt>
1. Press <Enter> to try again, or
2. Press Control-C (Ctrl-C) to exit.
_
```

If so, make sure that the path to the license file is indeed correct. Also, make sure that there are no problems with the floppy disk or drive (if your License File is on a floppy disk). Press the **Enter** key.

License File and License Database Do Not Match:

This problem usually arises during a new installation when License Manager software was previously installed on this computer. The licensing information is stored on a file called 'lmgr-hvdc' (with no extension), which is usually located in your Windows or WINNT directory. This file is not deleted when the License Manager is uninstalled. If this is indeed the case, select **Option 3** as indicated below:

```
C:\Program Files\HVDC Lmgr\Common\lmgr-enter.exe
-----
Manitoba HVDC Research Centre Inc.
Licence Manager Creator U1.xx
(c) 1998-200x Manitoba HVDC Research Centre Inc.
-----
Time: Thu Mar 04 09:50:30 200x

Please enter your Licence Key file name (default: a:\license.txt):
a:\license.txt

Searching for a Licence Manager lock device...
ERROR (enter.exe): The lock and license.txt do not match the Licence Manager dat
a file.
customer ID (lock device) : 1234
customer ID (license.txt) : 1234
customer ID (lmgr-hvdc file): 158003

You have the following option(s):
Option 1: Press <Ctrl-C> to exit.
Option 2: Insert the correct USB or serial port lock 158003,
and then press <Enter> to try again.
Option 3: Press the <D> key to delete all existing licences for lock 158003
then install the new licences for lock 1234 (currently plugged in).
```

If problems persist, press **Ctrl + C** to exit the 'Enter License Key' program and continue with the installation. When the installation is completed, attempt to add your license again by following the directions given in 'Adding/Upgrading Licenses' in this chapter. If this fails, follow the directions in 'Getting License Information.' Send the 'getinfo.txt' file to PSCAD Support Services (support@pscad.com) with a description of the problem.

```
C:\Program Files\HVDC Lmgr\Common\lmgr-enter.exe
-----
Manitoba HVDC Research Centre Inc.
Licence Manager Creator U1.xx
(c) 1998-200x Manitoba HVDC Research Centre Inc.
-----
Time: Thu Mar 04 09:50:30 200x

Please enter your Licence Key file name (default: a:\license.txt):
a:\license.txt

LICENCE 1:
added new licence to licence database.
HVDC product : Professional 4.0
Quantity of licence: 1 SU
Port number : 2053
Entered on : Mon Nov 17 10:44:17 2003
Expiry date : Sat Jan 31 23:59:59 2004
WARNING: in order to use the license(s) just entered,
you must stop, then restart the Licence Manager.

Press <Enter> to exit...
```

23. The Install program will now scan both the dongle and the license file and check for validity. The output will be similar to that shown above. Press the **Enter** key to exit.
24. This completes the License Manager installation process. It is highly recommended that you view the 'Read Me' file when prompted, as it contains useful information to users.

Chapter 2: Installation and Setup

If you did not choose to install the GNU Fortran compiler in Step 5, then installation is now complete (unless you selected to install the RTP demo software). Please review Finalizing Your Installation.

Installing EGCS/GNU Fortran

If you chose to install the GNU Fortran compiler in Step 5, the Installer will now begin to install it.

25. The GNU compiler installation is quite straightforward. Press **Finish** when complete.
26. This completes the EGCS/GNU Fortran installation process. It is highly recommended that you view the 'Read Me' file when prompted, as it contains useful information to users.

Installing LiveWire/LiveWire Lite

If you chose to install LiveWire in Step 5, the Installer will now begin to install it.

27. The LiveWire installation is quite straightforward. Press **Finish** when complete.
28. This completes the LiveWire installation process. It is highly recommended that you view the 'Read Me' file when prompted, as it contains useful information to users.

UPGRADING YOUR V4 SOFTWARE

This section is exclusively for those who are upgrading a previous version of PSCAD V4. If this is the first time that PSCAD and/or the License Manager is to be installed on a particular machine, see 'Installing V4 for the First Time' in this chapter.

Upgrading a Previous Installation with a Patch

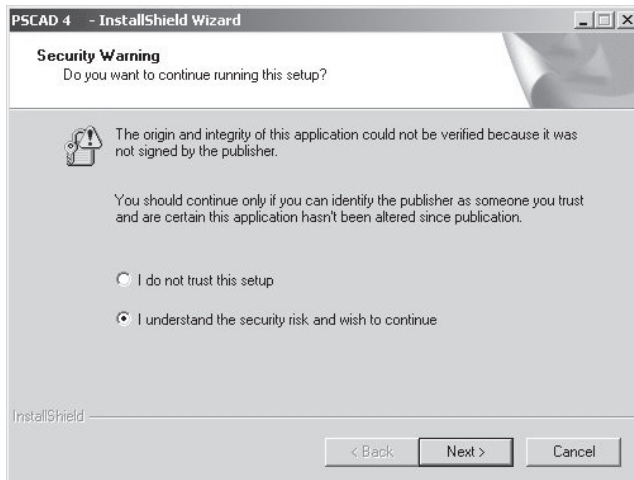
Patch upgrades are available for download on the official PSCAD website (www.pscad.com).

NOTE: Patching is not a full install and so you must have previously installed an older version of PSCAD V4.x.x before you can continue.

PSCAD

Please perform the following steps:

1. Go to the official PSCAD website (www.pscad.com) and select the Downloads section from the main menu bar. You will be asked to enter your PSCAD Member ID and Password before being allowed access to the download site. If you do not have a Member ID and Password, you can become a member by selecting the register link on the page. Contact us at support@pscad.com if you experience problems.
2. Once you are in the Downloads area, select the desired **Download, Update or Patch PSCAD V4** link under **PSCAD V4** heading. This will direct you to a new page corresponding to the PSCAD Edition type selected. Please review the information presented in this page before proceeding. Select the link entitled **Patch PSCAD™ V4 <xxx> Edition to V4.x.x**.
3. On the resulting page, click the **Install** button.
4. Set-up will initiate and you should see a dialog indicating the security risk of downloading the patch directly from our website. Please read carefully and select **I understand the security risk and wish to continue** and click **Next** IF this is indeed what you wish to do. If you wish to cancel the patch installation, click the **Cancel** button.



Chapter 2: Installation and Setup

5. The installer will then check your system for the required base installation. You must have previously fully installed a PSCAD v4.x.x version in order to proceed with the patch process. If this is indeed the case, then refer to Steps 6 and 7 in the next section.

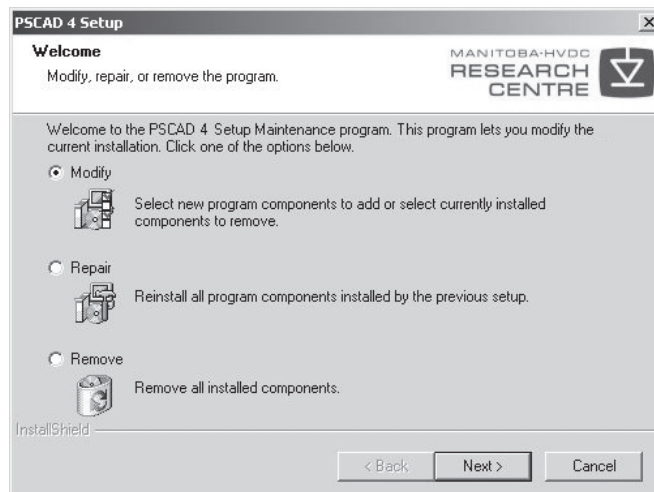
Modifying a Previous Installation

An option is provided to modify the PSCAD software components that were installed by a previous set-up. If your operating system is Windows NT4, XP or 2000, you **must** have administrator or power user access to your machine, in order to modify an installation.

NOTE: In order to install a patch, you have already installed a previous full version of PSCAD v4.x.x. *****REMOVE*****

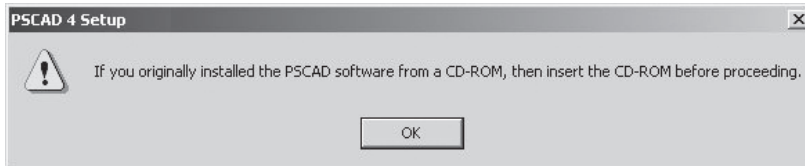
Please perform the following steps:

1. If a Standalone License Manager was previously installed on this PC, it must be stopped before you continue. See 'Manually Stopping the License Manager' in this chapter for more details.
2. Insert the PSCAD CD (from which you previously installed your software) into your CD-ROM drive.
3. The installer should start automatically and in a few moments, you should see the following screen:

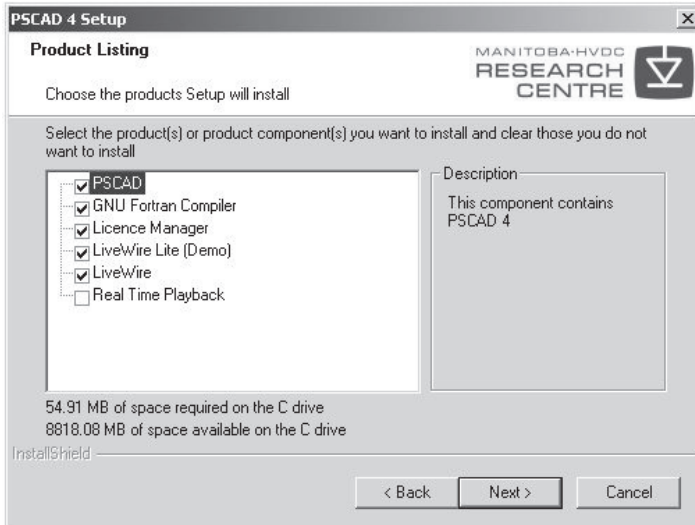


PSCAD

4. Select **Modify** and then click **Next**. Each screen that requires important user input will have a corresponding step to it in the following pages.
5. The following warning should pop-up. If this warning is not heeded, the Installer may not be able to perform the modifications. Click **OK**.



6. You will be presented with the following screen, on which options are available for your install. The options previously selected should appear. Select how you would like to modify your installation.



NOTE: All components selected will be installed or upgraded. **All components that are deselected will be uninstalled!**

Chapter 2: Installation and Setup

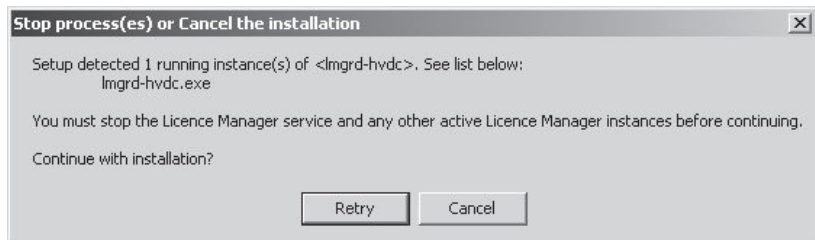
Explanations for each of the above options are given below.

- **PSCAD:** Select this option to install PSCAD. Although all PSCAD editions (i.e. Student, Educational and Professional) are included on the CD, you must be properly licensed to use any edition other than the Student Edition!
- **GNU Fortran Compiler:** If you do not currently have either the Digital or the Compaq Fortran 90 compiler installed, select the GNU Fortran compiler. A Fortran compiler is **required** for any PSCAD project to simulate (run).
- **License Manager:** Only select to install the License Manager if you intend for this machine to be the License Manager server on a network (Standalone Licensing).
- **LiveWire Lite (Demo):** Select this option to install the evaluation version of LiveWire. LiveWire is the primary data analysis and post-processing software tool recommended for use with PSCAD. You can get more information on this software at www.zsystems.ca/products.htm.
- **LiveWire:** Select this option to install the full version of LiveWire. Note that you must be licensed to run the full version. Please contact us at sales@pscad.com for more details on obtaining a full license.
- **Real Time Playback:** If you wish to evaluate the Real Time Playback (RTP) demo software (or install the commercial version), select this option. For more on the RTP, see our website at: www.pscad.com/

Possible Messages

License Manager is Currently Running:

If you selected to either install or uninstall the License Manager software and an instance of the License Manager is detected as running on this PC, you will receive the following dialog.

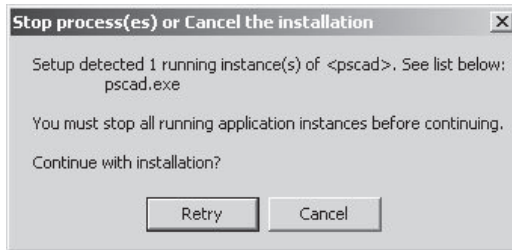


PSCAD

See Manually Stopping the License Manager for more details. Click **Retry** once the License Manager process has been stopped.

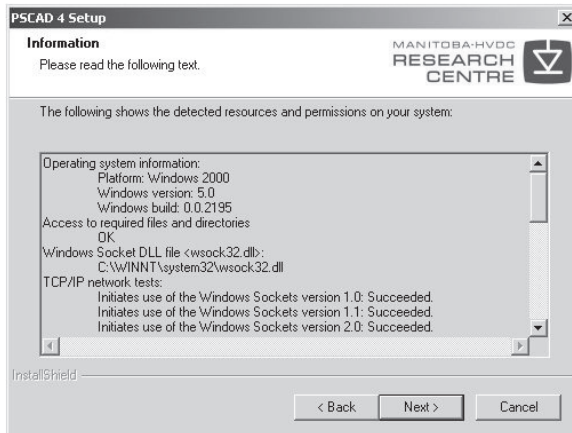
PSCAD is Currently Running:

If you selected to either install or uninstall the PSCAD and an instance of PSCAD is detected as running on this PC, you will receive the following dialog.



Close PSCAD and click **Retry**.

7. The Installer will now perform a variety of tests to verify system settings and to ensure that everything is OK to proceed. If you would like to see the results of these diagnostics, please have a look at the report dialog window (similar to that shown below):



Chapter 2: Installation and Setup

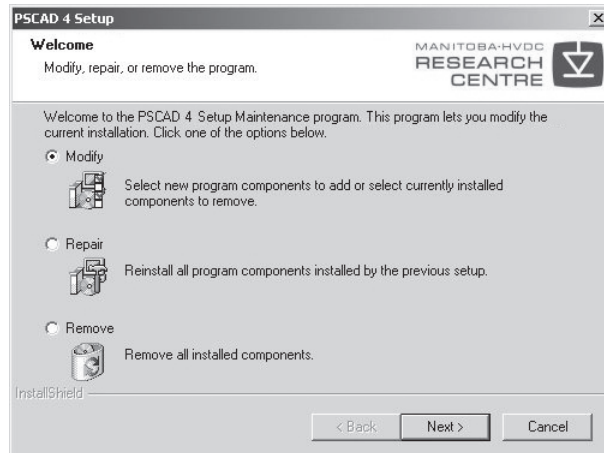
At this point, the Installer will attempt to install or uninstall components as selected in Step 6.

Repairing a Previous Installation

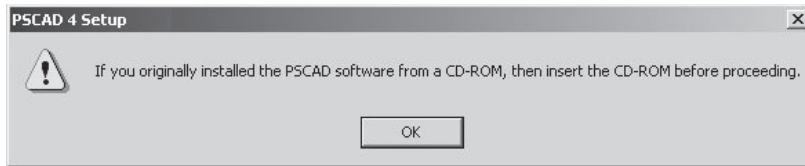
An option is provided to repair any PSCAD software components that were installed by a previous set-up. If your operating system is Windows NT4, XP or 2000, you **must** have administrator access to your machine, in order to repair an installation.

Please perform the following steps:

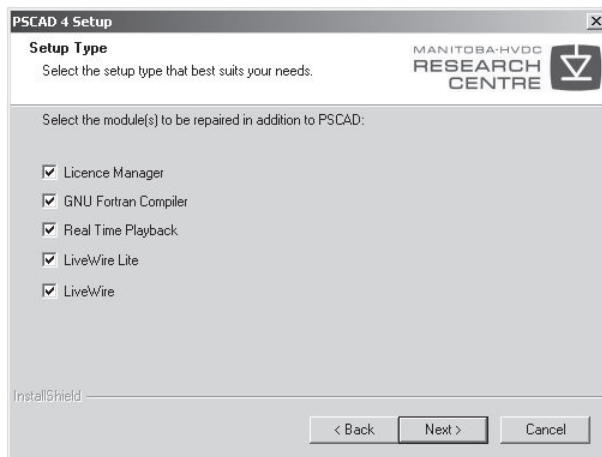
1. If a standalone License Manager was previously installed, it must be stopped before you continue. See 'Manually Stopping the License Manager' in this chapter for more details on how to stop the License Manager.
2. Insert the PSCAD CD (from which you previously installed your software) into your CD-ROM drive.
3. The installer should start automatically and in a few moments, you should see the following screen:



4. Select **Repair** and the click **Next**. Each screen that requires important user input will have a corresponding step to it in the following pages.
5. The following warning should pop-up. Click **OK**.

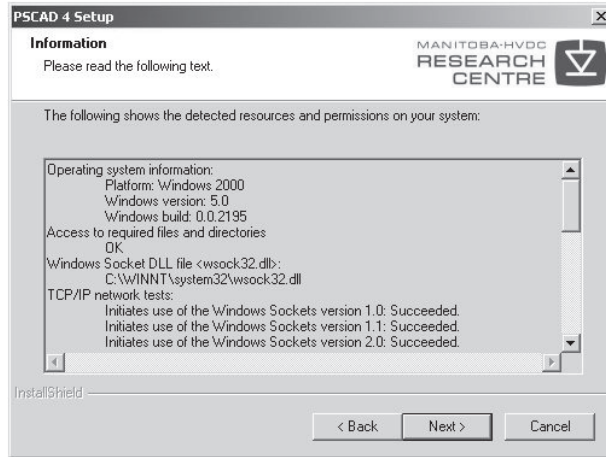


6. You will be presented with the following screen, on which software modules are available for your repair. Select which ones you would like to repair.



7. The Installer will now perform a variety of tests to verify system settings and to ensure that everything is OK to proceed. If you would like to see the results of these diagnostics, please have a look at the report dialog window (similar to that shown below):

Chapter 2: Installation and Setup

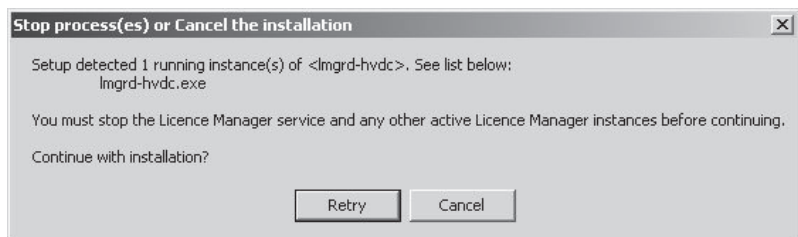


At this point, the Installer will attempt to re-install all components installed in the previous set-up.

Possible Messages

License Manager is Currently Running:

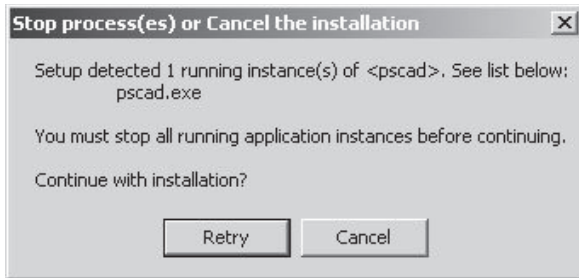
If you selected to repair the License Manager software and an instance of the License Manager is detected as running on this PC, you will receive the following dialog.



See [Manually Stopping the License Manager](#) for more details. Click **Retry** once the License Manager process has been stopped.

PSCAD is Currently Running:

If you selected to repair PSCAD and an instance of PSCAD is detected as running on this PC, you will receive the following dialog.



Close PSCAD and click **Retry**.

Uninstalling

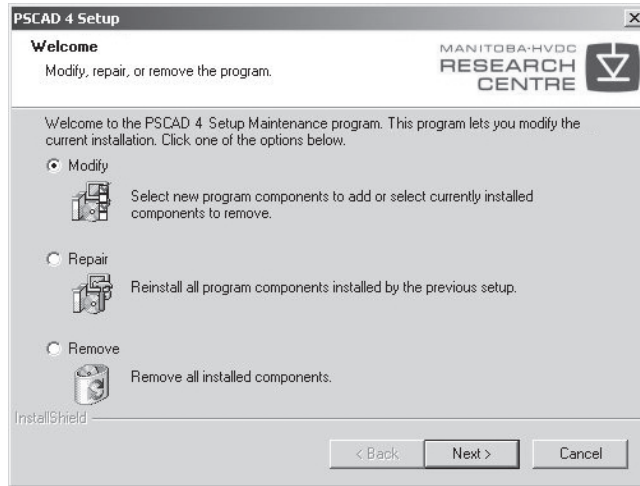
An option is provided to remove (uninstall) all PSCAD software components that were installed by the previous set-up. If your operating system is Windows NT4, XP or 2000, you must have administrator or power user access to your machine, in order to uninstall.

NOTE: The uninstall program will not remove any modified files. This includes the *.emt directories created by PSCAD at runtime. These must be removed manually.

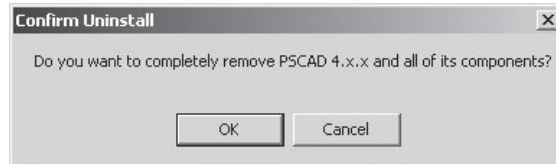
Please perform the following steps:

1. If a standalone License Manager was previously installed, it must be stopped before you continue. See 'Manually Stopping the License Manager' in this chapter for more details on how to stop the License Manager.
2. Insert the PSCAD CD, from which you previously installed your software, into your CD-ROM drive.
3. The installer should start automatically and in a few moments, you should see the following screen:

Chapter 2: Installation and Setup



4. Select **Remove** and then click **Next**.
5. The following confirmation warning should pop-up. Click **OK**.



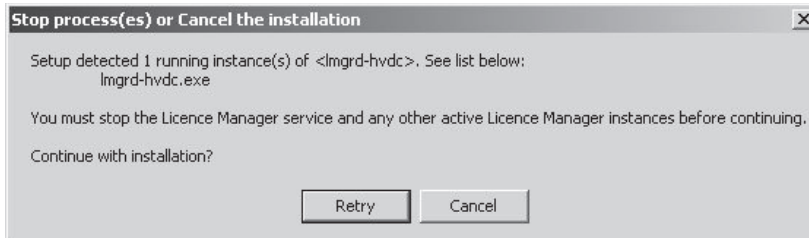
At this point, the Installer will attempt to re-install all components installed in the previous set-up.

Possible Messages

License Manager is Currently Running:

If you selected to remove the License Manager software and an instance of the License Manager is detected as running on this PC, you will receive the following dialog.

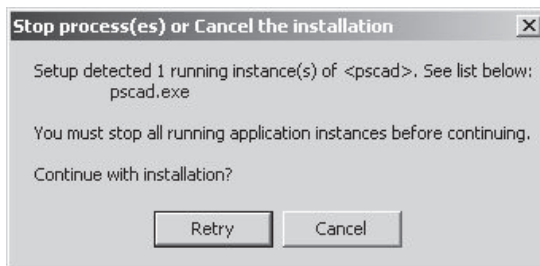
PSCAD



See [Manually Stopping the License Manager](#) for more details. Click **Retry** once the License Manager process has been stopped.

PSCAD is Currently Running:

If you selected to remove PSCAD and an instance of PSCAD is detected as running on this PC, you will receive the following dialog.



Close PSCAD and click **Retry**.

FINALIZING YOUR INSTALLATION

There are a couple of loose ends to tie-up before you begin using PSCAD V4.

Windows 98/ME Only

If you installed the EGCS/GNU Fortran 77 compiler or the License Manager, then you must reboot in order to set-up environment information.

If you chose to manually start the License Manager, then be sure to do so upon reboot, before starting PSCAD. See 'Manually Starting the License Manager' in this chapter for more details.

Chapter 2: Installation and Setup

Windows NT4/XP/2000 Only

Once installation is complete, you must manually start the License Manager service if you decide not to reboot your machine. See 'Manually Starting the License Manager' in this chapter for more details.

If you installed the EGCS/GNU Fortran 77 compiler, then you must log-out and then log-in in order to update the environment information.

Adding a Shortcut to the PSCAD Executable

All editions of PSCAD V4 now come complete as one executable file. Therefore, when creating a desktop shortcut, add one of the suffixes described below to your shortcut path.

Professional Edition:

"C:\Program Files\PSCAD4xx\bin\win\PSCAD.exe" **PRO**

Educational Edition:

"C:\Program Files\PSCAD4xx\bin\win\PSCAD.exe" **EDU**

Student Edition:

"C:\Program Files\PSCAD4xx\bin\win\PSCAD.exe"

Of course, if you have used something other than the default installation directory, the above paths will differ.

RUNNING PSCAD FOR THE FIRST TIME

To run PSCAD, simply click on START | Programs | PSCAD and select whichever PSCAD edition you are licensed for. The program should now load without any problems. You are now ready to simulate cases!

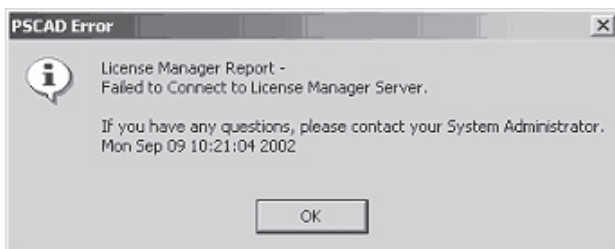
There are however, a couple of problems that may be encountered at this point. The most common are described below.

Failed to Connect to License Manager Server

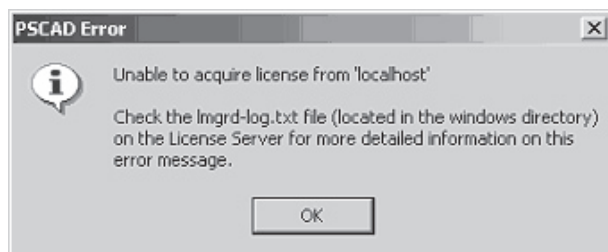
If there has been a problem with your License Manager installation, have lost contact with the License Manager, or have not started it, it will most likely be noticed when you first start PSCAD. When

PSCAD

PSCAD is invoked, it will ask the License Manager if there is a license available. If PSCAD cannot communicate with the License Manager at all, the following screen will be encountered upon attempting to start PSCAD:



Followed by:



NOTE: 'localhost' is the host name of the License Manager server. The host name indicated could also be another computer on the network.

This usually indicates one of the following possible problems:

- The License Manager is installed but either not configured properly or not running at all. This may include problems with the dongle (i.e. it is not connected or corrupt).
- The License Manager server on the network is down, or communication with the server is impossible.
- The machine name (or IP address) of the License Manager server is incorrect.
- The user failed to run the Enter License Key program to enter their license information.

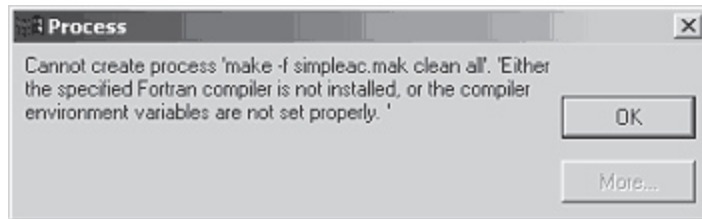
Chapter 2: Installation and Setup

Solution: First check the dongle's physical connection. If this machine is on the LAN, ask your system administrator for help. Ensure that the machine name that you entered is correct and that there are no spaces in the name. Check the License Manager Log file (i.e. Lmgr-log.txt or PscadLmgr.txt in your Windows or WINNT directory) for clues: Did you enter your License Key? If using Windows NT4, XP or 2000, did you start the License Manager service? See 'Adding / Upgrading Standalone Licenses' or 'Adding / Upgrading Single-User Licenses' in this chapter for more details on how to add a license.

If all else fails, contact us by email at PSCAD Support Services (support@pscad.com). Attach the 'lmgrd-log.txt' file and the 'PscadLmgr.txt' files from your Windows or WINNT directory for us to analyze.

Running a Simulation Generates a 'make -f' or 'nmake -f' Error

You may encounter a 'make -f' or 'nmake -f' error when attempting to run a simulation for the first time.



NOTE: The 'make -f' error means that PSCAD is configured to use the GNU Fortran compiler, while 'nmake -f' refers to one of the Fortran 90 compilers.

This error indicates that PSCAD is unable to connect with a Fortran compiler so as to build an executable for simulating. This can be caused by one of the following reasons:

- The specified Fortran compiler is not installed.
- A virus scanner is blocking TCP/IP communication.
- The compiler environment variables are not installed properly.
- Another program uses a utility program with the same name (i.e. 'make' or 'nmake') and is defined first in either the

'autoexec.bat' file, or in the system environment variables. In other words, this other utility is being called.

Solution: Ensure that the specified Fortran compiler is installed. Make sure that your virus scanner is configured to allow the required communication. Check that the compiler environment variables are set properly. Check either your 'autoexec.bat' file or your environment variables (depending on your OS) to see if there is a conflicting utility. If so, you may be able to solve this problem by placing the utility definition statement, related to the PSCAD compiler, before the other. When in doubt, ask your system administrator for help.

If all else fails, contact us by email at PSCAD Support Services (support@pscad.com) with a detailed description of the problem.

SETTING COMPILER ENVIRONMENT VARIABLES

Each compiler installer program should automatically set the environment variables for the corresponding Fortran compiler. However, there are instances where this may fail. The following are instructions on how to manually set these variables as per compiler and operating system:

EGCS/GNU Fortran 77 Compiler

Windows NT4/XP/2000:

1. Right-click on your 'My Computer' icon and select **Properties...**
2. Click the **Advanced** tab and select **Environment Variables.**
3. Ensure that the following User Variable is set:

Variable	Value
GCC_EXEC_PREFIX	C:\progra~1\egcs\lib\gcc-lib\
PATH	C:\progra~1\egcs\bin

NOTE: The environment variable paths should be in 8-character DOS format.

Windows 98/ME:

Chapter 2: Installation and Setup

The above changes need to be made by editing the 'autoexec.bat' file according to standard formats. For help on this, contact your system administrator.

Digital Fortran 5 Compiler

Windows NT4/XP/2000:

1. Right-click on your 'My Computer' icon and select **Properties...**
2. Click the **Advanced** tab and select **Environment Variables.**
3. Ensure that the following System Variables are set:

Variable	Value
FPS_LIB	C:\Program Files\DevStudio\VC\LIB
INCLUDE	C:\Program Files\DevStudio\DF\INCLUDE; C:\Program Files\DevStudio\VC\INCLUDE
LIB	C:\Program Files\DevStudio\DF\LIB; C:\Program Files\DevStudio\VC\LIB
Path	C:\Program Files\DevStudio\SharedIDE\BIN; C:\Program Files\DevStudio\DF\BIN; C:\Program Files\DevStudio\VC\BIN

Windows 98/ME:

The above changes need to be made by editing the 'autoexec.bat' file according to standard formats. For help on this, contact your system administrator.

Compaq Fortran 6 Compiler

Windows NT4/XP/2000:

1. Right-click on your 'My Computer' icon and select **Properties...**
2. Click the **Advanced** tab and select **Environment Variables.**
3. Ensure that the following System Variables are set:

PSCAD

Variable	Value
INCLUDE	C:\Program Files\Microsoft Visual Studio\DF98\INCLUDE; C:\Program Files\Microsoft Visual Studio\VC98\INCLUDE
LIB	C:\Program Files\Microsoft Visual Studio\DF98\LIB; C:\Program Files\Microsoft Visual Studio\VC98\LIB
Path	C:\Program Files\Microsoft Visual Studio\Common\Tools; C:\Program Files\Microsoft Visual Studio\Common\ Msdev98\BIN; C:\Program Files\Microsoft Visual Studio\DF98\BIN; C:\Program Files\Microsoft Visual Studio\VC98\BIN

Windows 98/ME:

The above changes need to be made by editing the 'autoexec.bat' file according to standard formats. For help on this, contact your system administrator.

Intel Fortran 9 Compiler

Windows NT4/XP/2000:

1. Right-click on your 'My Computer' icon and select **Properties...**
2. Click the **Advanced** tab and select **Environment Variables**.
3. Ensure that the following System Variables are set:

Variable	Value
INTEL_LICENSE_FILE	C:\Program Files\Common Files\Intel\Licenses
LIB	C:\Program Files\Intel\Compiler\Fortran\9.0\IA32\lib
Path	C:\Program Files\Intel\Compiler\Fortran\9.0\IA32\Bin

If Microsoft Visual Studio C++ .NET 2002 is installed, the following values are additionally required:

Variable	Value
LIB	C:\Program Files\Microsoft Visual Studio .NET\VC7\PlatformSDK\lib; C:\Program Files\Microsoft Visual Studio .NET\VC7\lib
Path	C:\Program Files\Microsoft Visual Studio .NET\VC7\bin; C:\Program Files\Microsoft Visual Studio .NET\Common7\IDE

Chapter 2: Installation and Setup

If Microsoft Visual Studio C++ .NET 2003 is installed, the following values are additionally required:

Variable	Value
LIB	C:\Program Files\Microsoft Visual Studio .NET 2003\vc7\PlatformSDK\lib;
Path	C:\Program Files\Microsoft Visual Studio .NET 2003\vc7\lib C:\Program Files\Microsoft Visual Studio .NET 2003\vc7\bin; C:\Program Files\Microsoft Visual Studio .NET 2003\Common7\IDE

Windows 98/ME:

The above changes need to be made by editing the 'autoexec.bat' file according to standard formats. For help on this, contact your system administrator.

TRIAL LICENSES

If you have already installed the PSCAD V4 Student Edition and require more than the fixed 15 electrical node limit, or if you would like to review the component design features, you may be interested in a time limited evaluation license for the PSCAD V4 Professional Edition. If interested, please follow the procedures below to receive your Trial License.

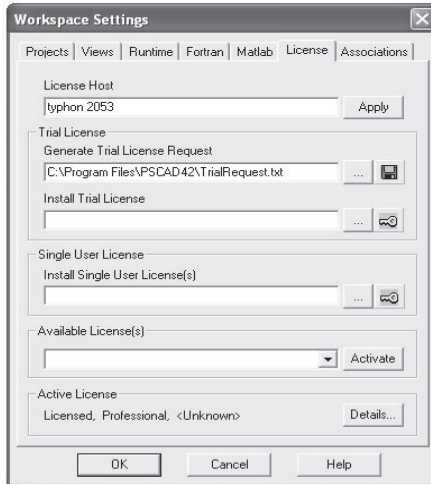
NOTE: Once you have submitted Trial License request, a new License Key will be sent pending approval. Trial licensing is only available for the PSCAD V4 Professional and Educational Editions.

Requesting a Trial License

As part of the Requesting a Trial License procedure, you must provide us with some specifics on the PC you will be using to run PSCAD. This information is supplied to us in the form of a text file, which contains important machine metrics, such as computer name and other operating system information. This data is required to create a Trial License file and is held in the strictest confidentiality.

The following procedure outlines how to request a Trial License:

1. Open the PSCAD Student Edition and go to **Edit | Workspace Settings...** in the main menu bar to open the Workspace Settings dialog.
2. Select the **License** tab near the top of the dialog.

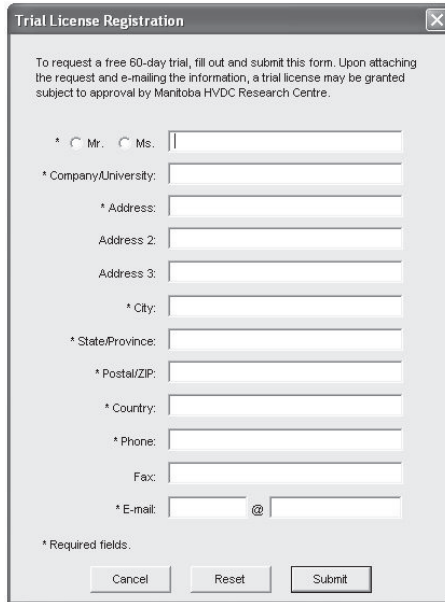


3. In the **Trial License** area of this dialog, enter a filename and path in the input field entitled **Generate Trial License Request** or use the **Browse** button to navigate to the desired directory. This will specify where PSCAD is to place the Trial License Request file once it is generated. To avoid confusion, leave this path and filename as default. Press the **'Generate'** button directly to the right of this field.



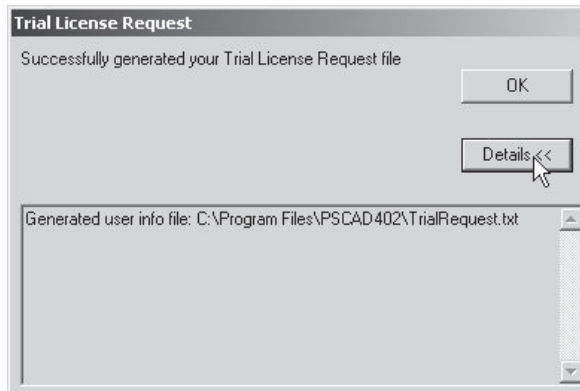
4. A Trial License Registration entry form will appear. Enter all required information into the form and then click the **Submit** button near the bottom of the form.

Chapter 2: Installation and Setup



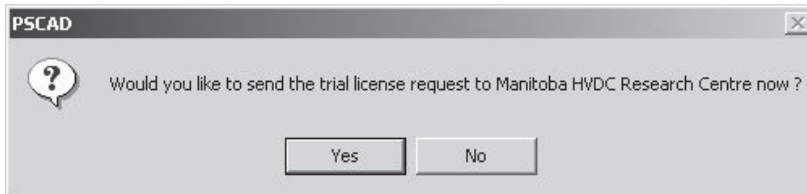
The screenshot shows a dialog box titled "Trial License Registration". At the top, there is a close button (X) and a paragraph of text: "To request a free 60-day trial, fill out and submit this form. Upon attaching the request and e-mailing the information, a trial license may be granted subject to approval by Manitoba HVDC Research Centre." Below this text are several input fields, each with an asterisk indicating it is required. The fields are: a name field with radio buttons for "Mr." and "Ms.", "Company/University", "Address", "Address 2", "Address 3", "City", "State/Province", "Postal/ZIP", "Country", "Phone", "Fax", and "E-mail" (split into two parts with an @ symbol). At the bottom, there are three buttons: "Cancel", "Reset", and "Submit".

5. If all fields in **Step 4** were entered correctly, PSCAD will generate the Trial License Request file and place it in the directory specified in **Step 3**. A Trial License Request dialog will appear confirming the placement of the file.



6. A dialog will appear asking if you would like to email the Trial License Request file now. Click **Yes** to proceed to **Step 7**. Click **No** if you wish to email the Trial License Request file at a later time. If you decide to send the file at a later

time, you can start at **Step 8** of this procedure, as the file has already been created.

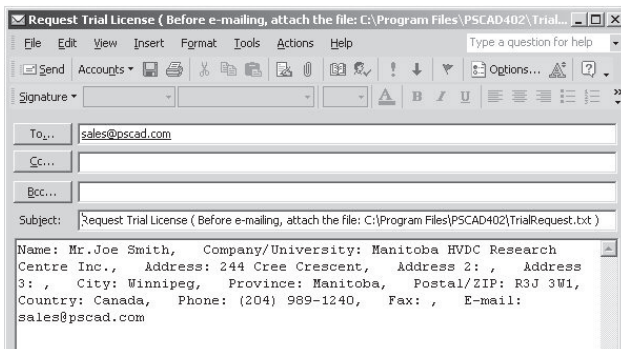


7. Another dialog will appear with a simple warning. Please read carefully:



8. PSCAD will then open a new email form from your default email tool. The body of the email will contain the information given in **Step 4** and the sending address is set to PSCAD Sales (sales@pscad.com).

NOTE: If you generated your Trial License Request file at an earlier time, you do not need to add the text to the body of the email, as this is contained within your Trial License Request file anyway.



Chapter 2: Installation and Setup

- Remember to manually attach the Trial License Request file to email in **Step 8** (the file will be located in the directory specified in **Step 3**). **Send** the email to PSCAD Sales (sales@pscad.com). You should receive a response within the next couple of days. See the next section entitled 'Installing a Trial License' upon receipt of your Trial License key.

NOTE: Do not upgrade your operating system, change your system clock, or change any hardware configuration (i.e. hard drives, motherboard, etc.) while waiting for your Trial License Key.

Installing a Trial License

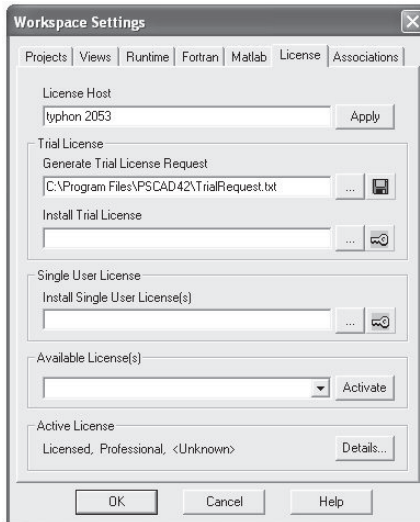
Upon approval of your Trial License request, you will receive an email from us with your Trial License key attached (i.e. 'Trial.txt').

Before proceeding with installation, please note the following important points:

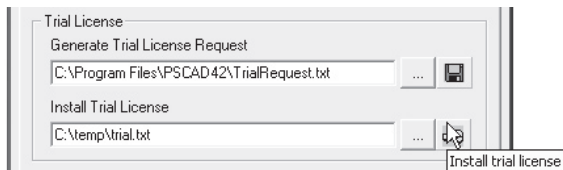
- The Trial License can only be installed on the computer from which it was requested.
- When operating under a Trial License, PSCAD will only run if it is the only PSCAD instance active on your machine. Quit all other instances of PSCAD (both V3 and V4), prior to starting with a Trial License.
- The Trial License will not operate if certain computer system configurations are changed. **Do not** upgrade your operating system, change your system clock, or change any hardware configuration (i.e. hard drives, motherboard, etc.) during the course of the Trial License period.
- If you encounter problems during this process, contact us at PSCAD Support Services (support@pscad.com).

The following procedure outlines how to install a Trial License:

- Save the Trial License key (i.e. 'Trial.txt' file) somewhere on your hard drive (say C:\temp).
- Open the PSCAD Student Edition and go to **Edit | Workspace Settings...** in the main menu bar to open the Workspace Settings dialog.
- Select the **License** tab near the top of the dialog.

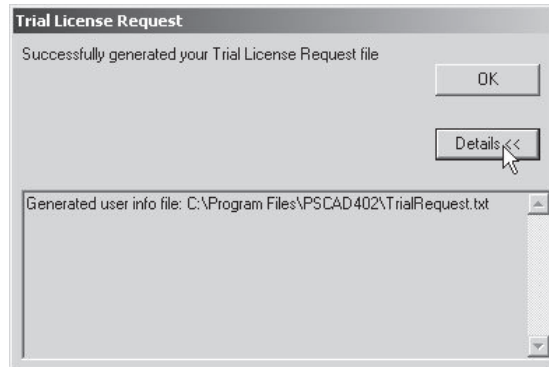


4. In the Trial License area of this dialog, enter the filename and path to your 'Trial.txt' file in the input field entitled **Install Trial License** or press the **Browse** button to navigate to the file. Once entered, press the **Install** button directly to the right of this field.



5. If Trial License is installed correctly, a Trial License Installation dialog will appear confirming this, along with the expiry date of the Trial License. Click the **OK** button.

NOTE: If the license does not install correctly, please contact us at PSCAD Support Services (support@pscad.com).



6. Close the PSCAD Student Edition and then open the PSCAD Professional Edition. See 'Viewing Active License Information' in this chapter for details on checking if your Trial License is running properly.

NOTE: Please see 'Adding a Shortcut to the PSCAD Executable' in this chapter if you plan to add a shortcut to the PSCAD Professional or Educational Edition on your desktop.

ADDING/UPGRADING STAND-ALONE LICENSE MANAGER LICENSES

You do not need to reinstall the License Manager in order to update or add a Standalone License Manager License. You will simply need to update the License Manager database file, which resides on your computer's hard drive.

To update or add a new Standalone License Manager License to your license database file, use the 'Enter License Key' program:

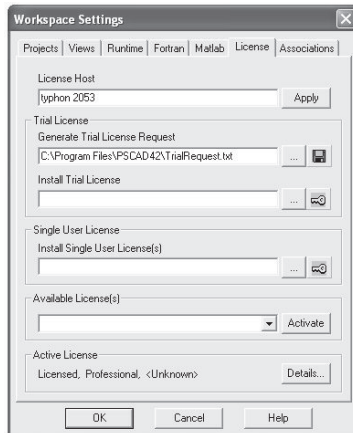
1. Go to START | Programs | HVDC Lmgr and select 'Enter License Key.'
2. Follow Steps 21 to 23 in 'Installing V4 for the First Time' in this chapter.

ADDING/UPGRADING A SINGLE-USER LICENSE

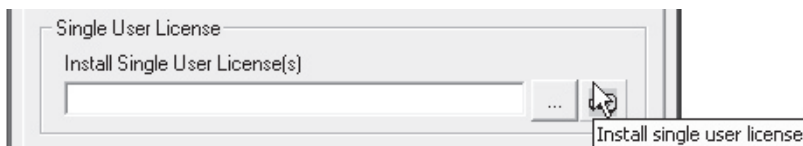
You do not need to reinstall the License Manager in order to update or add a single-user license. You will simply need to update the License Manager database file, which resides on your computer's hard drive.

To update or add a new license to your license database file:

1. Save your license file (located on the floppy disk provided with PSCAD, entitled 'License Manager: License.txt') to a convenient directory (say C:\temp).
2. Open PSCAD V4 Student Edition and go to **Edit | Workspace Settings...** in the Main Menu Bar. Click the **License** tab.



3. In the **Single User License** area of this dialog, enter the filename and path to your Single User license file in the input field entitled **Install Single User License(s)**. Press the **Install** button directly to the right of the input field and navigate to the saved 'license.txt' file. Select **OK** in the pop-up dialog window.

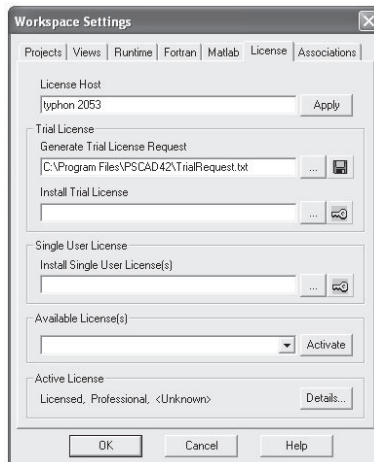


Chapter 2: Installation and Setup

4. Click **OK** on the **Workspace Settings** dialog and exit the PSCAD Student Edition.
5. You should now be able to load the PSCAD edition for which you are licensed.

CHANGING ACTIVE LICENSING SETTINGS

PSCAD allows you to alter your present (i.e. 'active') license settings without the need to close and reopen the program in a different mode. You can change the License Manager host computer, and/or the license type (i.e. Professional, Educational or Student) directly from the Workspace Settings dialog. To open this dialog, go to **Edit | Workspace Settings...** in the main menu bar and click the **License** tab.



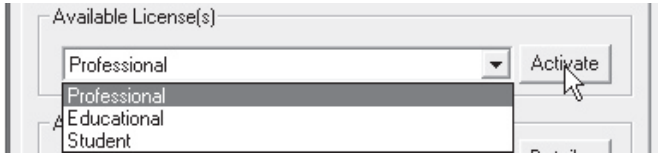
Changing License Manager Host

To change the License Manager host computer, simply enter the name of the new host (along with 2053) and click the **Apply** button.



Changing Active License Type

To change the active license type, click the down arrow in the selection box under the **Available License(s)** area and select the required license. Click the **Activate** button.

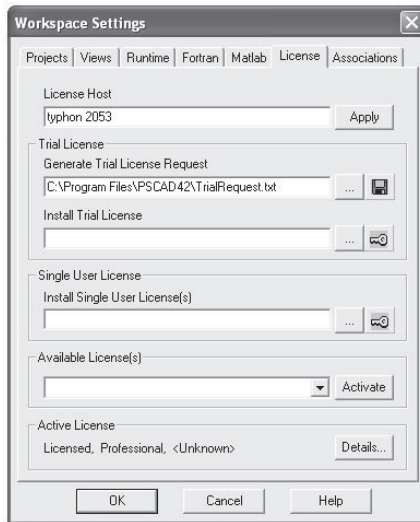


NOTE: You must be properly licensed in order to change license type!

VIEWING ACTIVE LICENSE INFORMATION

At any time, you can review the status of your active license.

1. Open PSCAD and go to **Edit | Workspace Settings...** in the Main Menu bar. Select the **License** tab.



2. Near the bottom of the dialog, there is an area entitled **Active License**. Some preliminary information is given

Chapter 2: Installation and Setup

directly within this area. For more detailed information, click the **Details...** button. A dialog should pop-up displaying all available information regarding your current licensing. If you wish you can save this data to a text file or copy it to the Windows clipboard. Select **OK** to close the dialog.



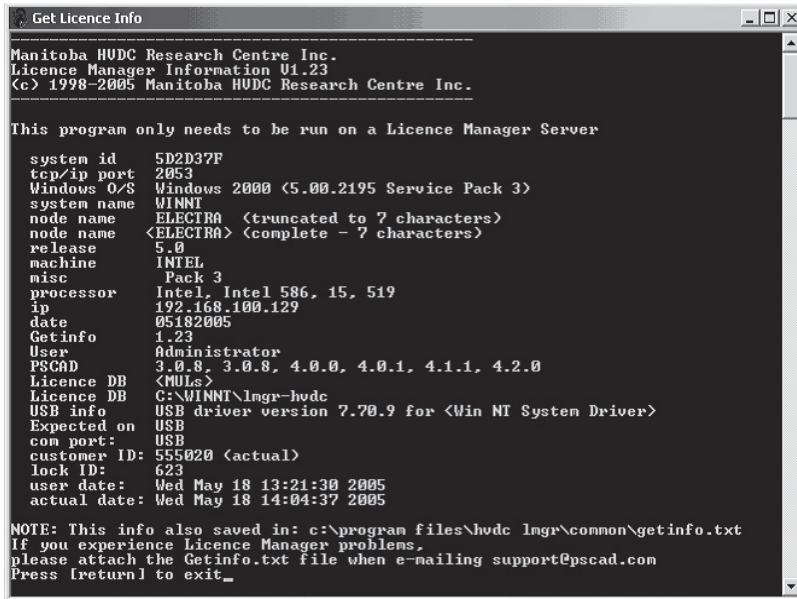
GETTING LICENSING INFORMATION

If you would like information about your licenses, you can get this directly from the dongle itself. This information is also helpful in troubleshooting if a problem arises, and is used by PSCAD Support to help solve License Manager issues.

Standalone License Manager Information

To view your Standalone License Manager information, use the 'Get License Info' program:

1. Go to **START | Programs | HVDC Lmgr** and select 'Get License Info.'



```
Get Licence Info
-----
Manitoba HVDC Research Centre Inc.
Licence Manager Information V1.23
(c) 1998-2005 Manitoba HVDC Research Centre Inc.
-----

This program only needs to be run on a Licence Manager Server

system id      5D2D37F
tcp/ip port    2053
Windows O/S    Windows 2000 (5.00.2195 Service Pack 3)
system name    WINNT
node name      ELECTRA (truncated to 7 characters)
node name      <ELECTRA> (complete - 7 characters)
release        5.0
machine        INTEL
misc           Pack 3
processor       Intel, Intel 586, 15, 519
ip             192.168.100.129
date           05182005
Getinfo        1.23
User           Administrator
PSCAD          3.0.8, 3.0.8, 4.0.0, 4.0.1, 4.1.1, 4.2.0
Licence DB     <MULs>
Licence DB     C:\WINNT\lmgrr-hvdc
USB info       USB driver version 7.70.9 for <Win NT System Driver>
Expected on    USB
com port:      USB
customer ID:   555020 (actual)
lock ID:       623
user date:     Wed May 18 13:21:30 2005
actual date:   Wed May 18 14:04:37 2005

NOTE: This info also saved in: c:\program files\hvdc lmgrr\common\getinfo.txt
If you experience Licence Manager problems,
please attach the Getinfo.txt file when e-mailing support@pscad.com
Press [return] to exit_
```

When you run this program, a DOS based window should appear similar to that shown above.

A text file will also be created at the same time called 'getinfo.txt' (as indicated above). This is an important file to include when contacting PSCAD Support Services by email (support@pscad.com).

Active Licensing Information

See 'Viewing Active License Information' in this chapter for more details.

MANUALLY STARTING THE LICENSE MANAGER

The method for starting the License Manager is different for the various Windows platforms:

Windows 98/ME

- Go to START | Programs | HVDC Lmgr. Select 'Start License Manager.'

Chapter 2: Installation and Setup

Windows NT

- Go to START | Settings | Control Panel | Services. Select the HVDC License Manager and select 'Start.'

Windows XP/2000

- Right-click on your **My Computer** desktop icon and select **Manage**.
- Double-click on the **Services and Applications** entry and then double-click **Services**.
- Highlight 'HVDC License Manager,' right-click and select 'Start.'

MANUALLY STOPPING THE LICENSE MANAGER

The method for stopping the License Manager is different for various Windows platforms:

Windows 98/ME

- Press **Ctrl + Alt + Del** to bring up the Task List and then use End Task to stop any instances of 'Imgrd-hvdc.'

Windows NT

- Go to START | Settings | Control Panel | Services. Select the HVDC License Manager and select 'Stop.'

Windows XP/2000

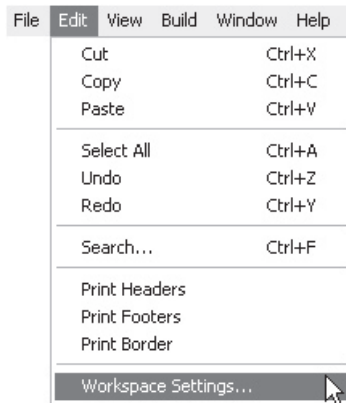
- Right-click on your **My Computer** desktop icon and select **Manage**.
- Double-click on the **Services and Applications** entry and then double-click **Services**.
- Highlight 'HVDC License Manager,' right-click and select 'Stop.'

Workspace Settings

Workspace Settings Overview

Now that PSCAD has been installed successfully, you may want to adjust some of the available work environment settings and preferences available. Workspace settings are those that affect the entire Workspace (i.e. all loaded Projects). These parameters are contained within a single dialog window called the Workspace Settings dialog.

To access this dialog, go to **Edit | Workspace Settings...** in the Main Menu bar.



The settings available in the Workspace Settings dialog control window are divided into seven specific areas:

- Projects
- Views
- Runtime
- Fortran
- Matlab
- License
- Associations

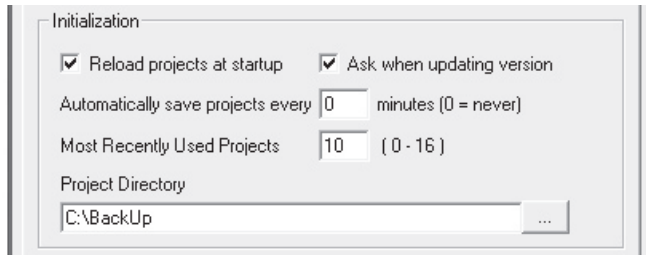
PROJECTS

The Projects section of the Workspace Settings dialog window contains a variety of user preference settings related to Projects. These are explained in detail below.



Initialization

This area provides some preferences specific to saving and initialization:



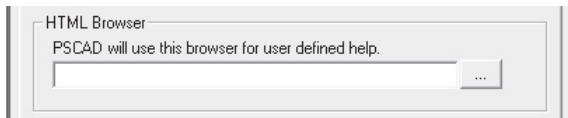
- **Reload Projects at Startup:** When selected, this setting will ensure that all Library and Case Project files, which were loaded in the Workspace when you last exited PSCAD, are reloaded again at start-up. When de-selected, only the Master Library will be loaded at start-up.
- **Ask When Updating Version:** This feature provides the user with some added protection against inadvertently changing the version of a loaded case or library project, when that project is saved. This becomes helpful for example, when the user is dealing with a client's case who uses an earlier version of PSCAD. If the case is saved as the latest version, the case may be rendered unusable by the client. When selected, PSCAD will check the original version of the case, and ask whether or not the case should be updated when saved.

PSCAD

- **Automatically Save Projects:** When selected, PSCAD will save all Library and Case Projects loaded in the Workspace at the specified intervals.
- **Most Recently Used Projects:** Enter the number of recently used Library and Case Projects that you would like to appear in the **File** menu of the Main Menu bar. You may enter a number in the range from 0 to 16 recently used Projects.
- **Project Directory:** Enter the default directory path for the Load Project dialog. You can either enter the path directly, or use the browse button to select the directory. This path may be changed at any time.

HTML Browser

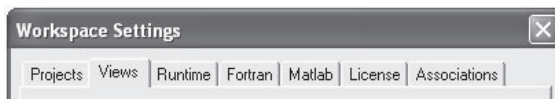
This area provides some preferences specific to HTML browser settings:



- **PSCAD will use this browser for user defined help:** Enter the path to your preferred HTML browser. You can either enter the path directly, or use the browse button to select the directory. This browser will be used for viewing your user-written component help files. See the section entitled 'Help Segment' in Chapter 9 of this manual for details on setting up component help files.

VIEWS

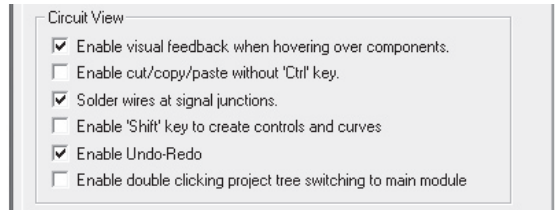
The Views section of the Workspace Settings dialog window contains a variety of mostly cosmetic preference settings and functions. These are explained in detail below.



Chapter 3: Workspace Settings

Circuit View

This area provides some preferences specific to the Circuit view:



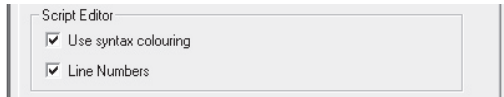
- **Enable Visual Feedback when Hovering Over Components:** Selecting this option will enable selection boxes to appear around Components and Modules when the mouse pointer is over top of them.
- **Enable Cut/Copy/Paste without 'Ctrl' Key:** Select this option to revert to the PSCAD V3 style hot keys for cut/copy/paste (i.e. **x**, **c** and **v**)
- **Solder Wires at Signal Junctions:** Select this option to show solder connection symbols at all overlapping signal junctions. This feature is used mostly to detect for overlapping Wires and possible short circuits. Once debugging is complete, this option should be turned off on larger Projects, as it can add significant build time to the Project.
- **Enable 'Shift' key to create controls and curves:** This option is directly related to Drag and Drop. By default, all Drag and Drop functions are invoked using the 'Ctrl' key in combination with the left mouse button. This option should be enabled if the user would prefer to designate the 'Shift' key (in place of 'Ctrl') to place Curves to Graphs, or Control Interfaces/Meters to Control Panels.
- **Enable Undo-Redo:** Select this option to enable the Undo and Redo feature.
- **Enable Double Clicking Project Tree Switching to Main Module:** When this option is enabled, a left double-click on the case name in the Projects section of the Workspace

PSCAD

window, will focus the Circuit canvas on the project main page.

Script Editor

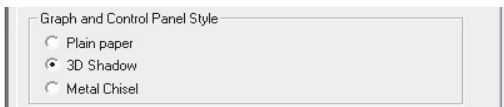
This area provides some preferences specific to the Script section of the Design Editor:



- **Use Syntax Colouring:** Select this option to turn on code syntax colouring in the Script section. When de-selected, all Script code will appear in black.
- **Line Numbers:** Select this option to enable display of line numbers when viewing code in the Script section or when viewing any text file in PSCAD.

Graph and Control Panel Style

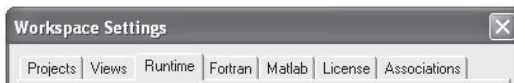
Select a preferred panel style for Control Panels and Graph Frames.



The default setting is 3D Shadow, but you may also choose either Plain Paper or Metal Chisel.

RUNTIME

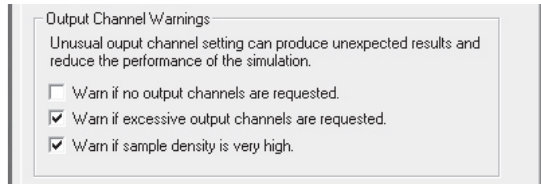
The Runtime section of the Workspace Settings dialog window is used to set-up output and memory related settings. These are explained in detail below.



Chapter 3: Workspace Settings

Output Channel Warnings

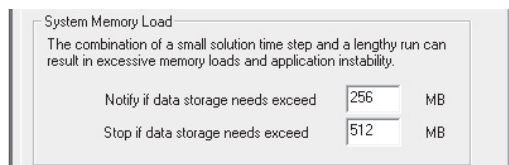
This section contains Output Channel settings:



- **Warn if no output channels are requested:** If this option is selected, PSCAD will issue a warning message upon Project Run indicating that there are no Output Channels (and hence no graphical feedback).
- **Warn if excessive output channels are requested:** Performs a sanity check to ensure that the user has not requested an excessive number of output channels. If the number of output channels is very large, the user should send channels on demand rather than all of them (otherwise, the simulation may perform very slowly). See the section entitled 'Send Only the Output Channels that are In Use' in Chapter 7 for more details.
- **Warn if sample density is very high:** Performs a sanity check to ensure the user has not selected a sample count that is too big for the system display and storage. This can occur when the time step is very small and the run duration is long. For example, a one second run plotted at every microsecond will have 1 million data points per trace. These are upper bound to ensure that the plot and graphs continue to function with reasonable performance. See the section entitled 'Time Settings' in Chapter 7 for more on setting simulation time step and plot step.

System Memory Load

This section contains Output Channel settings:

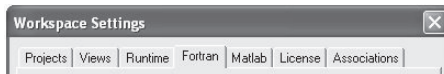


PSCAD

- **Notify if storage needs exceed:** If this option is selected, and the storage required for a simulation exceeds the entered value, a warning will be issued upon Project Run.
- **Stop if data storage needs exceed:** If this option is selected, and the storage required for a simulation exceeds the entered value, the Project Run will not proceed.

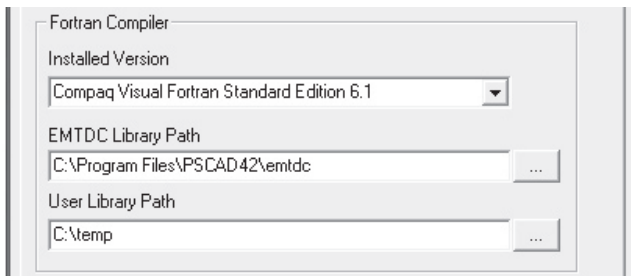
FORTRAN

The Fortran section of the Workspace Settings dialog window is used to set-up Fortran related settings. These are explained in detail below.



Fortran Compiler

This section contains Fortran compiler settings:



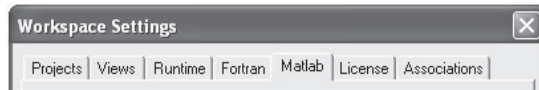
- **Installed Version:** Select the Fortran compiler to be used for compilation of PSCAD projects. If you have more than one compiler installed, you may freely switch between compilers without having to restart PSCAD or reload the project.
- **EMTDC Library Path:** Enter the path to the EMTDC library directories. You can either enter the path directly, or use the browse button to select the directory. This path is set to default to the installation directory and should be left as such.

Chapter 3: Workspace Settings

- **User Library Path:** The path entered here is automatically appended to files entered into the Additional Library (*.lib) and Object (*.obj) Files input field in the Project Settings dialog.

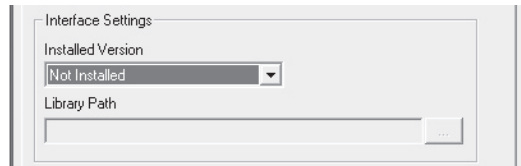
MATLAB®

The MATLAB® section of the Workspace Settings dialog window is used to set-up MATLAB®/Simulink® Interface related settings. These are explained in detail below.



Interface Settings

The Interface Settings area is used to set-up a directory path to the MATLAB® installation libraries.



- **Installed Version:** If you intend to make use of the MATLAB®/Simulink® Interface in PSCAD, then choose which version of MATLAB® you will be using in this field. PSCAD V4 currently supports MATLAB® Versions 5.0 and 6.
- **Library Path:** If you are using MATLAB® Version 5.0, you must specify a path to the MATLAB® installation library directory. This path will be used for all PSCAD projects using the MATLAB®/Simulink® Interface. If MATLAB® version 6 is selected, setting a path to the MATLAB® installation library directory is not required. You can either enter the path directly, or use the browse button to select the directory.

NOTE: Enabling the MATLAB®/ Simulink® interface is a Project specific option. See the section entitled 'Link Options' in Chapter 7 for more details.

LICENSE

The License section of the Workspace Settings dialog window is devoted to License Manager and other licensing preferences and features. These are explained in detail below.



License Host

The hostname of the License Manager server machine on your network.



This field is automatically configured during installation and should not need to be adjusted unless your License Manager server host-name is changed. The additional '2053' number is always required (specifies the port number). This field is not used if you are using a trial license. For more details on how to change the License Manager host directly from this dialog, see the section entitled 'Changing Active Licensing Settings' in Chapter 2.

Trial License

The Trial License area is used only when requesting and installing a trial license.

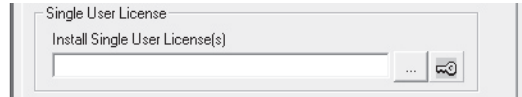


Chapter 3: Workspace Settings

See the section entitled 'Trial Licenses' in Chapter 2 for more details.

Single User License

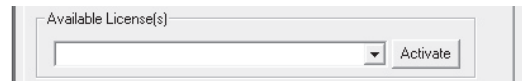
The single user license area is used only when installing and activating single user licenses.



See the sections entitled 'Adding/Upgrading a Single-User License' and 'Getting Licensing Information' in Chapter 2 for more details.

Available License(s)

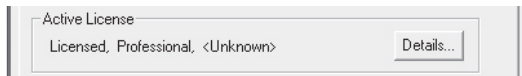
The Available License(s) area is used for switching between available licenses.



See the section entitled 'Changing Active Licensing Settings' in Chapter 2 for more details.

Active License

The Active License area is used only when requesting and installing a trial license.



See the section entitled 'Trial Licenses' in Chapter 2 for more details.

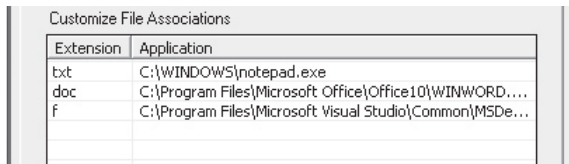
ASSOCIATIONS

The Associations section of the Workspace Settings dialog window is used specifically to set-up file association settings. All file associations specified here, will enable the user to start-up external applications from within PSCAD, by using the File Reference component.



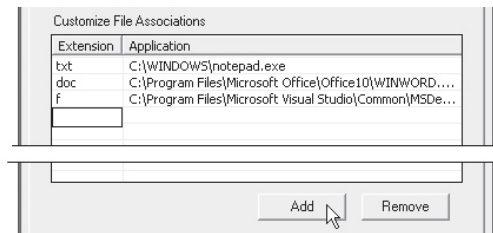
Customize File Associations

File associations are entered in two parts: The file extension (with no period), and the path to the application executable file.

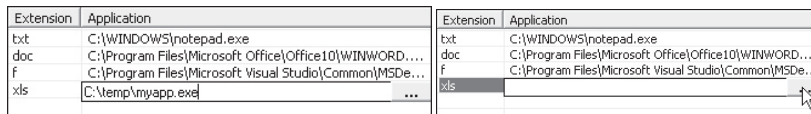


Creating a New Association

To enter a new file association, press the **Add** button. The next available input box in the Extension column will appear as selected.



Enter the file extension (without the period) and press **Enter**. Either **double left-click** the box directly to the right of the new extension, or press the Browse (...) button to enter a path to the corresponding application executable file. The Browse button will bring up a dialog window by which you can navigate to the file.



Chapter 3: Workspace Settings

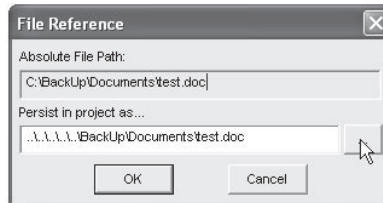
Removing an Association

Simply select the association to be removed and then press the **Remove** button.

Invoking an Associated External Application

To make use of the file associations entered here, the File Reference component must be linked to a file with a specified extension, as follows:

1. Add a File Reference component by right-clicking over a blank part of the Circuit canvas and selecting **Add Component | File Reference**.
2. Either left double-click on the component or right-click on it and select **Properties...** to bring up the component properties dialog.
3. Enter the path (relative to the Project file) to the file either directly or by pressing the Browse button.



That's all there is to it! The next time the File Reference component is double-clicked, it will start the associated application and open the specified file.

The PSCAD Environment

The term ‘PSCAD Environment’ refers not only to how the PSCAD graphical users interface is organized visually, but also to naming conventions, utilities and other features that facilitate creativity and design. As you may notice, a lot of effort has gone into enhancing the work environment in PSCAD V4. We have however, tried to maintain the look and feel of the previous versions, preserving the most popular functions and enhancing others.

This chapter discusses a wide range of introductory topics ranging from simple terms and definitions, to a general overview of most important functions in PSCAD. The tutorial section My First Simulation is a must read for new PSCAD V4 users.

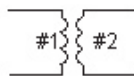
For a more detailed description of basic functions and operations, see Chapter 5 - Features and Operations.

TERMINOLOGY AND DEFINITIONS

In order to fully appreciate the PSCAD User’s Guide and Online Help System, a quick review of the following common terms is suggested. Due to the environmental changes implemented since PSCAD V3, there have been several changes in the terminology used to describe some new and old features as well.

Components

A component (sometimes referred to simply as a ‘block’) is essentially a graphical representation of a device model, and is the basic building block of circuits created in PSCAD. Components are usually designed to perform a specific function, and can exist as either electrical, control, documentary or simply decorative in type.



Single-Phase Transformer component in PSCAD

Chapter 4: The PSCAD Environment

Components usually possess inputs and outputs and can be pieced together with other components to form larger systems. Model parameters, such as input variables and constants, can also be manually entered through the component.

Definitions

A definition is essentially the underlying 'blueprint' of a component. Accessed through the Design Editor, the component definition is where all aspects of the component design are defined. This can include graphical appearance, connection nodes, input dialogs and model code.

Component definitions are not graphical entities, and are usually stored in library projects. Definitions stored in a library project can be used to create component instances in any case project, including that library project itself. However, definitions stored in a case project are local to that case and cannot be used in other projects.

Instances

A component instance is a graphical 'copy' of the component definition, and is normally what is seen and used in a project. An instance is not exactly a copy, in that many component instances, based on the same definition can exist simultaneously. Each instance is its own entity, and can have different parameter settings, or even appear graphically different from other instances.

Since all instances are based on a single definition, any design changes to a component definition will affect all instances.

Modules

Modules are a special type of component, where the basic function of the component model is described using a combination of other basic components, rather than code. Modules can contain other modules within them and thus provide a hierarchical modeling capability. Also referred to as 'sub-pages' or 'page components,' modules are also defined by a definition, but presently may only possess one instance per definition. A module behaves very much like an ordinary component except that it does not allow input parameter entry.

Projects

PSCAD allows the user to store everything involved in a particular simulation (except output files) into one file called a project. Projects can contain stored component definitions, on-line plots and controls,

PSCAD

and of course the graphically constructed system itself. There are two types of projects in PSCAD: Library and Case projects.

NOTE: Component definitions stored in case projects are local to that case and cannot be used in other projects. Library projects should not contain components defined in other cases or libraries.

Case

Case projects (or simply 'cases') are where most work is performed in PSCAD. In addition to performing the functions of a library project, cases may be compiled, built and run. Simulated results can be viewed directly within the project through on-line meters and/or plots. Case projects are saved with the file extension '*.psc.'

Library

Library projects are used primarily to store component definitions and viewable component instances. Instances of component definitions stored in a library, can be used in any case project, provided that the library is loaded first. Library projects are saved with the file extension '*.psl.'

TUTORIAL: MY FIRST SIMULATION

This tutorial is designed to give new users a 'jump start' in learning how to use PSCAD, as well as to give you a chance to try out a PSCAD simulation before proceeding further in this chapter.

In this tutorial you will learn:

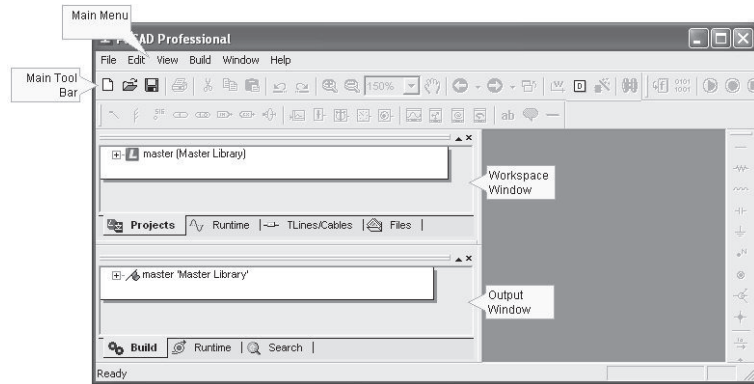
- How to Load a Case Project
- How to Run a Simulation
- How to Print

If you can successfully run this example Project at the end of this chapter, it will also mean that your installation was successful. So let's begin!

Starting PSCAD

To start PSCAD, go to **Start | Programs | PSCAD** in the Windows Start Menu, and select whichever PSCAD Edition you are licensed for. This will open the main PSCAD environment, as shown below.

Chapter 4: The PSCAD Environment



You should see a list of items across the top of the environment (**File**, **Edit**, etc.). These items are part of the Main Menu. The buttons directly below the main menu are part of the Main Toolbar.

Title, Menu and Main Tool Bar

The image below shows the PSCAD V4 Title and Menu Bars.



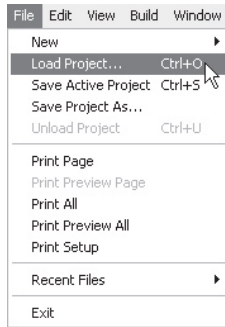
Title Bar and Active Project

The top most part of the window that displays **PSCAD [Edition]** is called the Title Bar. The Title Bar will also include the Project page that is currently being viewed in the Design Editor, once a Project is opened.

Menu Bar and Menu Items

The area under the Title Bar, consisting of menu items and menu buttons, is called the Main Menu. All main menu items are drop down menus, so when you left-click with your mouse pointer over top of one of these menus, you will see a list appearing below it.

To select an item from this list, first move the pointer onto that item (the item becomes raised) and then click the left mouse button. The following menu shows how to load a project using the **File** menu from Main Menu bar.



Toolbar Buttons

The row of buttons directly below the Main Menu bar comprises the Main Toolbar.

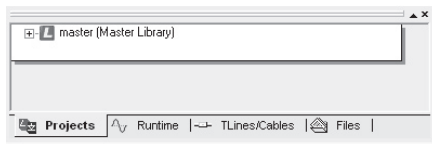


While menu items are a two-stage process, a toolbar offers a single-click alternative. The Main Toolbar buttons initiate actions as soon as you click on them and hence are easier to use. For this reason, most frequently used operations have toolbar button equivalents.

After you become more familiar with the program, you may begin to use the many keyboard shortcuts available. See the section entitled 'Keyboard Shortcuts' in this chapter for more details.

Workspace and Output Windows

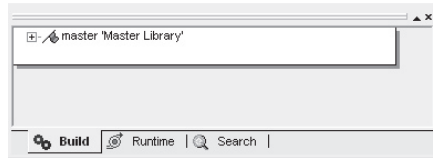
If you take a look at the top left-hand corner of the PSCAD Environment, you should see a smaller window referred to as the Workspace window. If it is not visible, go to the main menu bar and click on **View | Workspace**.



The Workspace gives you an overall view of any library and/or case projects loaded. You can use it to select components and perform many other activities.

Chapter 4: The PSCAD Environment

Directly below the Workspace window, you should see another window referred to as the Output window. If the Output window is not visible, click on **View | Output**.



All the status, warning and error messages involved in both Build and Runtime procedures are logged in this window - so it is a good idea to keep this window open and visible at all times.

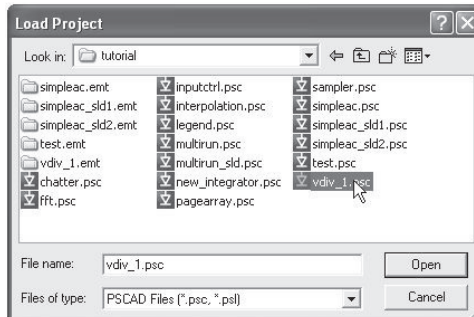
Both the Workspace and Output windows are based on a tree-type structure, where branches can be expanded and collapsed at will.

Loading a Case Project

We will start with the most simple of example cases for this tutorial. This exercise will help us to ensure that PSCAD and any Fortran compilers being used are installed correctly. We will learn to create a case from scratch in 'Creating a New Project' tutorial of Chapter 5.

To load an existing Case Project, click on **File** from the Main Menu bar and select **Load Project...** You can also either press **Ctrl + O** or click the **Load** button in the Main Toolbar.

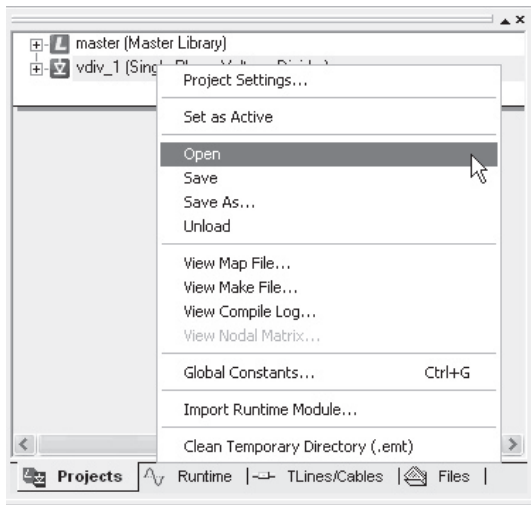
You should see the Load Project dialog pop up on your screen. By default, the selected file type is 'PSCAD Files (*.psc, *.psl)' at the bottom of the dialog. With this type selected, you will see only PSCAD Project files that have '*.psc' or '*.psl' extension and directories. The file extensions stand for **pscad case** or **pscad library**.



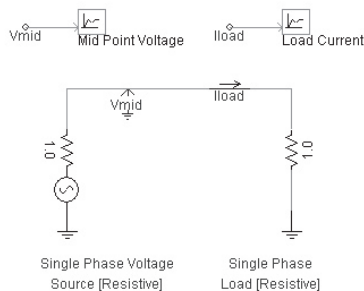
PSCAD

Navigate to the 'tutorial' directory inside your PSCAD installation directory (i.e. C:\Program Files\PSCAD4xx\examples\tutorial). Click on the 'vdiv_1.psc' file and then click on the **Open** button to load this Case Project into PSCAD.

The Workspace window will now list a second project entitled 'vdiv_1 (Single Phase Voltage Divider)' directly under the Master Library Listing. Double-click on the Project title (or right-click and select **Open**) in the Workspace window to open and view the main page of the Project in the Circuit window of the Design Editor.



You should see the assembled voltage divider circuit as shown below, which is located at the top left corner of the main page of the project that you just opened. The plots are situated directly to the right of the circuit.



Chapter 4: The PSCAD Environment

The circuit consists of a single-phase resistive voltage source connected to a resistive load. Since the magnitude of the source resistance (1Ω) and the load resistance are the same, the voltage at the load terminal is half that of the voltage behind the source resistance. This voltage is measured using a voltmeter called 'Vmid' connected to the node between the source and the load. The current in the circuit should be:

$$I_{\text{Load}} = \frac{E}{R_s + R_L}$$

The plot and graphs will contain the values of the voltage at the mid-point of the circuit, and the current flowing through the circuit when the project is run.

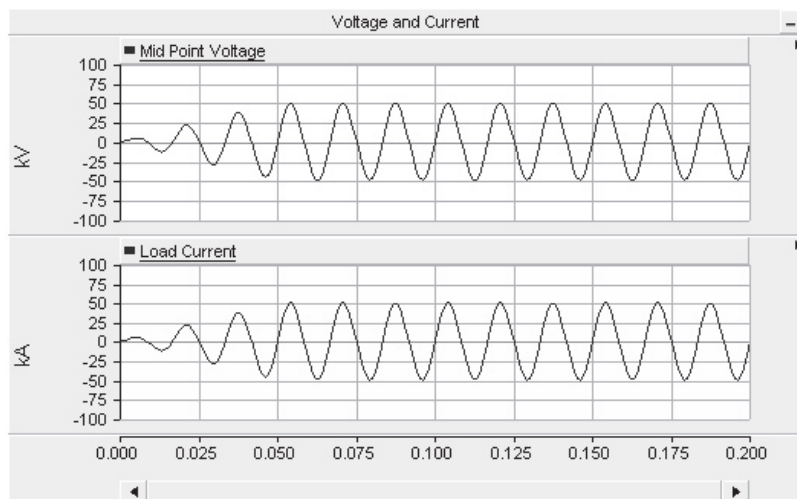
Running a Simulation

Before we run the simulation we will do a simple calculation to find out what load current and mid-point voltage we should be expecting. Double-click on the source component to open and view its properties - note that the source voltage magnitude is 70.71 kV RMS (or 100 kV peak). Close this dialog by clicking on the **Cancel** button at the bottom of the dialog and left-click anywhere in an empty space on the page, to de-select the selected source component (this will stop the component from blinking). For a 100 kV source voltage, we know that the mid-point voltage should then be 50 kV peak, and the load current should be 50 kA peak. Now let us run the simulation and actually verify the current and voltage waveforms.

To run a case, simply click on the **Run** button in the Main Toolbar. When this button is pressed, PSCAD will go through several stages of processing the circuit before starting the EMTDC simulation. You should see a message in the status bar at the bottom of the PSCAD window, related to various stages of the process. Depending on how fast your computer is, you may not be able to read these.

Watch the graphs as the simulation progresses. If you look near the bottom-right corner of the PSCAD Environment, you will see a message 'xx% complete' where 'xx' represents the percentage of the total simulation length. To the right of it you will also see the current simulation time, which changes with the simulation. Once again, depending on the speed of your computer, the simulation may finish almost instantaneously.

This tutorial case is set up to run for 0.2 seconds. At the end of the run you will see the message 'EMTDC run completed' in the status bar. Your plots should look similar to the following - depending on your plot settings:



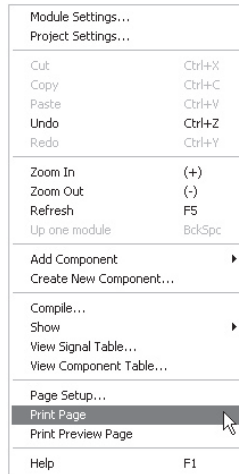
Make sure that your simulation produces the same result as shown here. This is one step towards ensuring that your PSCAD is installed correctly.

Click again on the **Run** button to see the run once again. PSCAD will go through all three stages (i.e. compile, build and run), however, you may not be able to detect the first two stages, as they pass by very quickly. This is because PSCAD performs them only if changes have been made to the circuit.

Printing the Circuit

To print the circuit along with the graph you just simulated, click the right mouse button on the background of the main circuit page and either select **Print Page** or **Print Preview Page** items.

Chapter 4: The PSCAD Environment







This should bring up either the Print dialog or the Print Preview Viewer. The contents of the Print dialog depends on what you are printing - click the **OK** button to proceed. The Print Preview Page allows you to view your page before printing.

THE WORKSPACE

Formally known as the 'Project Tree' in PSCAD V3, the Workspace has matured into a central project database for PSCAD. The Workspace not only provides an overview of all projects currently loaded, but also organizes data files, signals, controls, transmission line and cable objects, display devices, etc. within an easily navigable environment. Features such Drag and Drop may also be utilized from the Workspace.

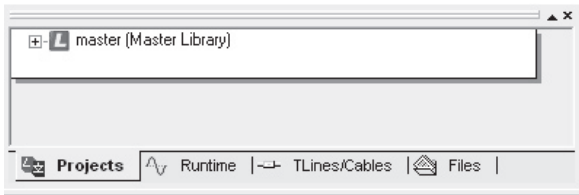
NOTE: The PSCAD Master Library is always the first project loaded into the Workspace. The Master Library cannot be unloaded.

The Workspace window is divided into four tabular sections:

-  Projects
-  Runtime
-  TLines/Cables
-  Files

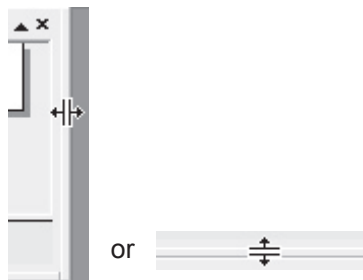
Moving and Resizing the Workspace Window

The Workspace window may be moved and dropped anywhere in the PSCAD environment. To do this, click and hold the left mouse button with the mouse pointer directly over the top window bar.



Drag the window to where you want it placed (you should see a boxed outline of the window) and let go of the mouse button. Depending on where the window outline is when the mouse button is released, the window will either dock itself into position or appear as a floating window.

The Workspace window can be resized by moving the mouse pointer over one of the four window edges until the pointer changes to that shown below:



Click and hold the left mouse button and drag the mouse pointer to resize in either the horizontal or vertical direction as shown above.

The Projects Section




When a Case or Library Project is loaded into PSCAD, the Project filename and description will appear in the Projects section of the Workspace window. It is possible to load multiple Projects – and if this happens – the Projects will be listed in the order in which they were loaded.

The Projects section is used mainly for navigating either between Projects or within Projects; including direct access to Modules and

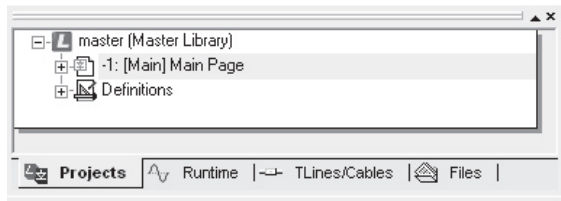
Chapter 4: The PSCAD Environment

Definitions. A simple double-click on a listed Module, for instance, will bring you directly to that page in Circuit view.

Icons are included for easy visual differentiation between Case and Library Projects. These are listed below:

-  Library Project
-  Inactive Case Project
-  Active Case Project

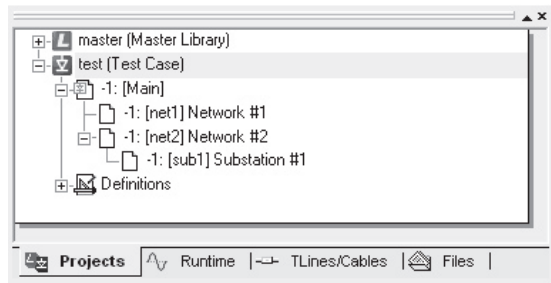
As mentioned above, each project listed in the Projects section contains information regarding any stored Definitions, as well as Module hierarchy, organized into a standard tree-type structure. This structure is referred to as the Project Tree.



You can expand and detract each branch in the project tree by clicking on either the [+] or the [-] symbols respectively.

Main Page Branch

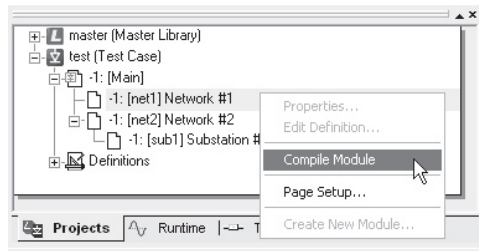
This branch lists the various Module instances in the project. Starting with the main page, any existing Modules are organized as they exist in the Project, not only making navigation simple, but providing an overview of the Project structure.





PSCAD

For example, the Case Project above (called 'test') contains a main page with two Modules called 'net1' and 'net2.' The Module 'net1' contains another Module called 'sub1.'

A simple double-click on a particular Module will bring you directly to that page in the Circuit window of the Design Editor. Right-clicking on a Module will bring up a menu where you can either compile that particular Module, or change the Module page set-up. See 'Compiling an Individual Module' or 'Editing the Page Setup' in Chapter 5 for more details on these menu functions.



Icons are included in this branch for visual representation and are listed below:

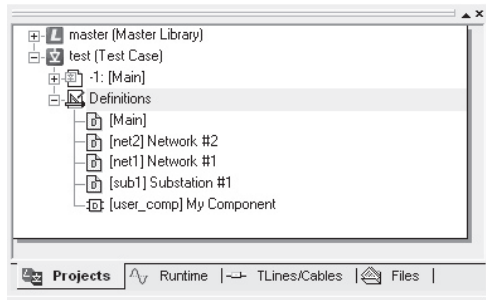
-  Main Page
-  Module Instance

Definitions Branch

The Definitions branch contains a list of all definitions that are stored within that particular Project. Instances of component Definitions that are defined in other Projects (such as Master Library components) do not appear here.

The image below shows the list of component Definitions for the same Case Project discussed in the previous section. In addition to the definitions of each Module, the existence of a component Definition (called 'user_comp') is also indicated.

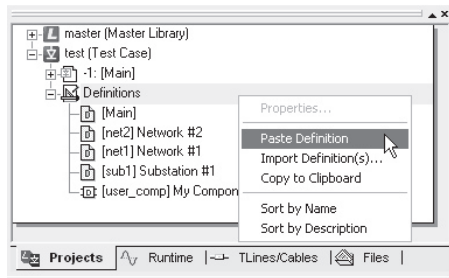
Chapter 4: The PSCAD Environment



NOTE: This example is for illustrative purposes only. It is not usually a good idea to define component Definitions within a Case Project, as they can then only be used in that Project. All component Definitions should exist exclusively in Library Projects.

A simple double-click on a Module Definition will bring you directly to that page in the Circuit window of the Design Editor, whereas double-clicking on a component Definition will bring you directly to the Graphic view.

There are a couple of right-click menu functions involved with the Definitions branch, as well (some of these functions are not available anywhere else). Right-clicking directly over the Definitions branch will bring up a menu as shown below:



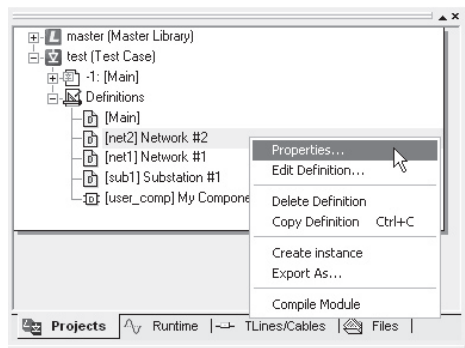
The following list describes the functions of this pop-up menu:

- **Paste Definition:** This function can be used to paste a Definition that has been copied from this Project or from

another Project currently loaded in the Workspace (see the Copy Definition menu function described below).

- **Import Definition(s)...**: This function is used to import stored Definition Files (*.cmp). See 'Importing/Exporting Definitions' in Chapter 5 for more details.
- **Copy to Clipboard**: This option allows the user to copy the entire Definitions list (i.e. names only) to the Windows clipboard as ASCII text.
- **Sort By Name**: Select this option to sort the Definitions list by name. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.
- **Sort By Description**: Select this option to sort the Definitions list by description. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.

If you right-click with the mouse pointer over a specific component Definition in the list, the following menu will appear:



The following list describes the functions of this pop-up menu:

- **Properties...**: This brings up the Definition Properties dialog window for editing the definition Name and Description. See 'Editing Definition Properties' in Chapter 5 for more details.
- **Edit Definition**: Select this option to edit the Definition of the component (opens the Graphic section of the Design Editor). See 'Editing a Component or Module Definition' in Chapter 9 for more details.
- **Copy Definition**: Select this option to copy the Definition to the Windows Clipboard.
- **Delete Definition**: Select this option to delete the Definition (and all existing Instances of course!).

Chapter 4: The PSCAD Environment

- **Create Instance:** Select this option to create an Instance of the component for graphical display in the Circuit window of the Design Editor. Note that once this option has been selected, go to a blank part of the Circuit canvas, right-click and select **Paste**. This will paste the copied Instance directly onto the page. Once the first instance has been created, subsequent copies may be made by directly copying the original Instance. Drag and Drop may also be used to perform this feature.
- **Export As...:** This option brings up the 'Export As' dialog so you can save the Definition to a Definition File (*.cmp). See 'Importing/Exporting Definitions' in Chapter 5 for more details.







Modules Only:

- **Compile Module:** If the selected Definition is a Module, then selecting this option will compile that Module individually.

Components Only:

- **Help:** If the Definition is a component, selecting this option will open the associated help file.



Icons are included for visual differentiation between component definitions and module definitions. These are listed below:

-  Component Definition (no Instances exist in Project)
-  Component Definition (one or more Instances exist in Project)
-  Module Definition (no Instances exist in Project)
-  Module Definition (an Instance exists in Project)
-  Component Definition has changed since last save
-  Module Definition has changed since last save

Active Project

As mentioned earlier, the Projects section of the Workspace will allow you to have multiple Projects loaded simultaneously. If a Case Project is to be compiled and run, PSCAD needs to know which one – this is accomplished by selecting an 'Active' Case Project (right-click on the project and select **Set as Active**). The Projects, Runtime and Files sections of the Workspace operate based on the Active Project.

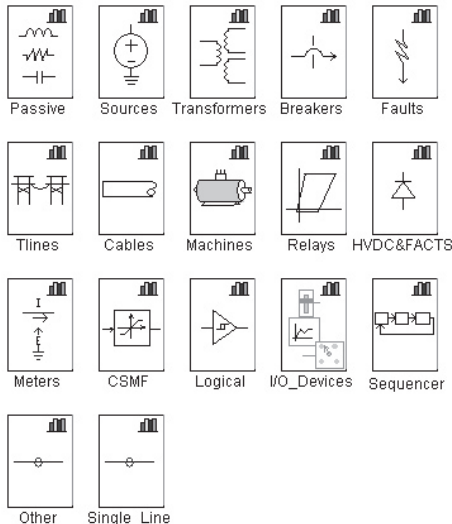
Active case projects are indicated as shown below:

-  Inactive Case Project
-  Active Case Project

See 'Setting the Active Project' in Chapter 5 for more details.

The Master Library

The Master Library is always the first Project listed in Projects section of the Workspace whenever PSCAD is started. It contains most of the components required to build almost any circuit. To open the Master Library, simply left double-click the title in the Workspace or right-click on the title and select **Open** from the pull down menu. The Master Library page will open (Circuit view in the Design Editor window), giving access to all of its components.



Chapter 4: The PSCAD Environment

Components stored in the Master Library are categorized into several Modules (located in the top-left corner of the main page), according to the functionality of the component. For example, all transformer components are stored in the Transformers Module. In addition to these categories, most Master Library components are also located on the main page. Some users may find this format allows for easier navigation to the correct component.

NOTE: Component definitions stored in the Master Library can be viewed, but not directly modified. If you wish to modify one of these definitions, copy the definition into another Library Project first.

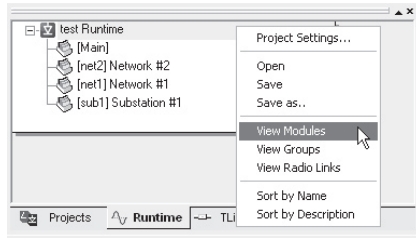
If you want to add any of these components to your own case, simply copy the Instance directly from the Master Library and paste into your own Project. You can also use the Library Pop-up Menu system, or the Draw toolbars.

The Runtime Section

The Runtime section is provided specifically for the organization of Runtime related objects, such as Output Channels, Controls, Graphs, etc. into a convenient tree-based environment. The main trunk of the Runtime tree is by default arranged according to existing Modules. However, it is also possible to view Runtime objects according to Runtime Group, or to view a list of existing Radio Links components. Regardless of the view settings, any Runtime object can be navigated to directly with a simple double-click.

NOTE: The Runtime Section is available only for the currently Active Project. If you would like to see the Runtime details of a Project that is not active, simply set the desired project as active first before clicking the Runtime tab. See 'Setting the Active Project' in Chapter 5 for more details.

To switch between each view type, right-click on the Project in the Runtime section to view the pop-up menu.



Select one of the following:

- **View Modules:** Select this option to view all Runtime objects according to Module hierarchy.
- **View Groups:** Select this option to view all Runtime objects according to Groups. See 'Groups View' below and 'Grouping of Runtime Objects' in Chapter 6 for more details.
- **View Radio Links:** Select this option to view a list of existing Radio Links components.

The other functions in this menu are as described below:

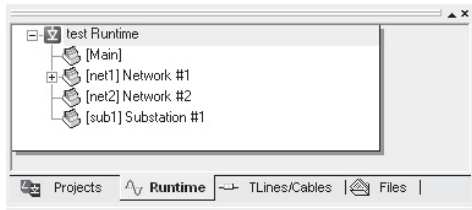
- **Project Settings...:** Select this option to access the settings dialog for the Project. See 'Editing Project Settings' in Chapter 5 for more details.
- **Open:** Select this option to open the Project main page in Circuit view. See 'Opening the Project Main Page' in Chapter 5 for more details.
- **Save:** Select this option to save the Project. See 'Saving Project Changes' in Chapter 5 for more details.
- **Save as...:** Select this option to save the Project to another filename. See 'Saving Project Changes' in Chapter 5 for more details.
- **Sort By Name:** Select this option to sort all Modules displayed by name. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.
- **Sort By Description:** Select this option to sort all Modules displayed by description. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.

Modules View

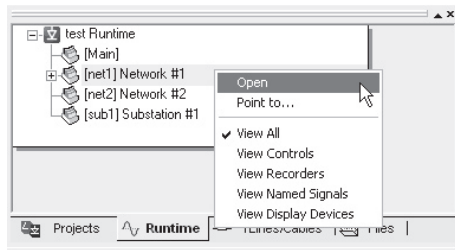
In Modules View, each Module Instance existing in a Project is represented in the Runtime section as a branch. If the Module contains any Runtime objects, such as Output Channels, Controls, Graphs, etc. within it, a [+] box will appear beside its branch. In the

Chapter 4: The PSCAD Environment

Case Project shown below, the only Module containing any Runtime objects is 'net1.'



A simple double-click on a particular Module branch will bring you directly to that page in the Circuit window of the Design Editor. Right-clicking on a Module branch will bring up the corresponding pop-up menu.



The following list describes the functions available in this pop-up menu:

- **Open:** Opens the Module in Circuit view.
- **Point to...:** PSCAD will point to the location of the Module Instance in the Project.
- **View All:** Allows viewing of all existing Runtime object sub-branches.
- **View Controls:** Allows viewing of Control objects only.
- **View Recorders:** Allows viewing of Recorder objects only.
- **View Named Signals:** Allows viewing of Named Signal objects only.
- **View Display Devices:** Allows viewing of Display Devices only.

NOTE: The following section entitled Runtime Objects describes these options in more detail.

PSCAD

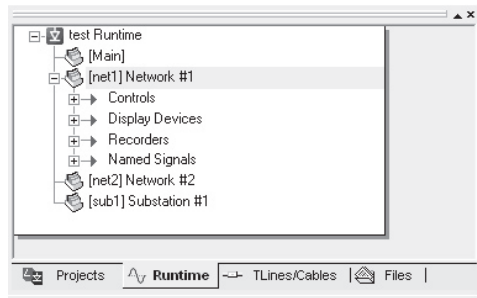
Runtime Objects

Each Module branch will contain a list of all Runtime objects that exist in the corresponding Module (provided that **View All** is selected as described above). Whenever a Runtime object is added to the Project, a Record of it is immediately added under the appropriate Module branch. The Record will remain until the object is removed from the project.

Runtime objects are classified as special components that are involved exclusively with the PSCAD Runtime interface and are organized into four main groups:

- Controls
- Recorders
- Display Devices
- Named Signals

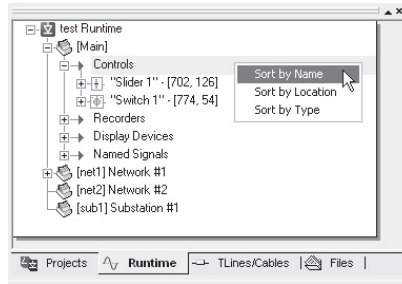
If a module contains any Runtime objects, these will appear as sub-branches under the corresponding Module branch, as shown below.



NOTE: If the Project does not contain any devices in a certain category, then the respective category branch will not appear here.

You can expand and detract each Module or sub-branch by clicking on either the [+] or the [-] symbols respectively. Right-clicking on a sub-branch branch will bring up the corresponding pop-up menu.

Chapter 4: The PSCAD Environment



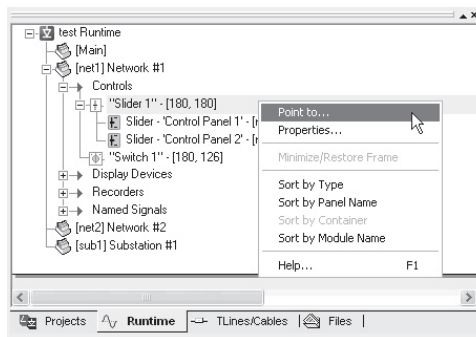
The following list describes the functions of this pop-up menu:

- **Sort By Name:** Select this option to sort all Records contained within the sub-branch by Record name. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.
- **Sort By Location:** Select this option to sort all Records contained within the sub-branch by Record location (i.e. coordinates). Continually selecting this option will toggle the alphabetical order from 1 to 9 or 9 to 1.
- **Sort By Type:** Select this option to sort all Records contained within the sub-branch by Record type.

A simple double-click on any Runtime object will point you directly to that object in the Circuit window. For example, double-clicking on the 'Slider 1' object under 'Controls' in the above diagram will result in the following in Circuit view:



Right-clicking on a Runtime object will invoke the pop-up menu:



The following list describes the functions of this menu:

- **Point to...:** Select this option to point directly to the associated Runtime object in Circuit view (same as double-click).
- **Properties...:** The properties dialog associated with this Instance of the Runtime object can be accessed directly using this option.
- **Help...:** Invokes the help page related to the Runtime object.

Display Devices Only:

- **Minimize/Restore Frame:** Select this option to minimize/restore any display devices. See 'Frame/Panel Minimization' in Chapter 6 for more details.

Controls and Records Only:

- **Sort By Type:** Select this option to sort all Observers (if they exist) associated with the Runtime object by Type. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.
- **Sort By Module Name:** Select this option to sort all Observers (if they exist) associated with the Runtime object by Module name. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.

Controls Only:

- **Sort By Panel Name:** Select this option to sort all Observers (if they exist) associated with the Runtime object by Control Panel name. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.



Records Only:

- **Sort By Container:** Select this option to sort all Observers (if they exist) associated with the Runtime object by Container (i.e. Graph Frame or Control Panel). Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.





NOTE: You may use the Drag and Drop feature to send Controls and Records directly to a Display Device.

Chapter 4: The PSCAD Environment



Icons are included in each Runtime object section for easy visual identification and are listed below:

-  Module
-  Runtime sub-branch






Controls:

-  Slider Component Instance
-  Two State Switch Component Instance
-  Dial Component Instance
-  Push Button Component Instance




Recorders:

-  Output Channel Instance
-  RTP/COMTRADE Recorder Component Instance

Display Devices:

-  Control Panel Instance
-  Plot Frame Instance
-  XY Plot Instance
-  PolyMeter Instance
-  PhasorMeter Instance

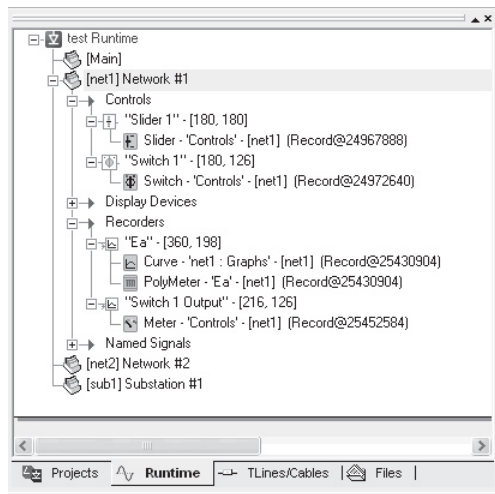
Named Signals:

-  REAL Data Signal defined with a Data Label
-  INTEGER Data Signal defined with a Data Label
-  LOGICAL Data Signal defined with a Data Label

Observers

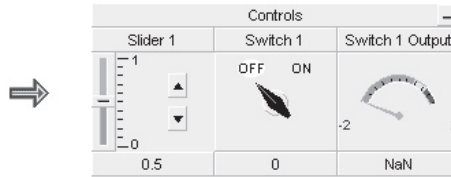
Output Channels and Control objects can exist in a Project without being associated with a Curve, Meter or Control Interface (i.e. the data is not plotted or monitored). If any of these objects is added as a Curve to a Graph, Meter to a Control Panel, or added directly to a PolyMeter or PhasorMeter, an entry referred to as an 'Observer' will be added as a sub-branch to the corresponding Recorder or Control in the Runtime section. If the object is displayed in more than one Display Device, an Observer will be added for each occurrence.

The figure below shows the existing Controls and Recorders in the 'Network #1' Module for our test Project. The Project contains a Slider called 'Slider 1,' a Two State Switch called 'Switch 1,' as well as two Output Channels entitled 'Ea' and 'Switch 1 Output.' Each object possesses an Observer to indicate that the output of the objects is being sent to a Display Device.



A simple double-click on any Observer will point you directly to that occurrence of Runtime Object within the Display Device. For example, double-clicking on the 'Slider 1' Observer under 'Controls' in the above diagram will result in the following in Circuit view:

Chapter 4: The PSCAD Environment



Icons are included for all Observer types. These are listed below:

Recorders:

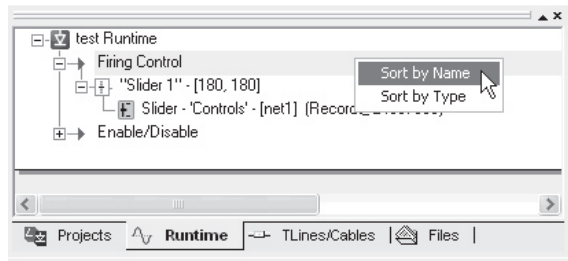
- Curve
- Meter
- PolyMeter
- PhasorMeter

Controls:

- Slider
- Dial
- Two State Switch
- Push Button

Groups View

In Groups View, Runtime objects are listed according to Runtime Group. If a particular Group contains any Runtime objects within it, a [+] box will appear beside its branch. Right-clicking on a Group branch will invoke a pop-up menu:



NOTE: To switch to Groups view, right-click on the Project name and select **View Groups**. See the beginning of this topic for details.

The following list describes the functions available in this pop-up menu:

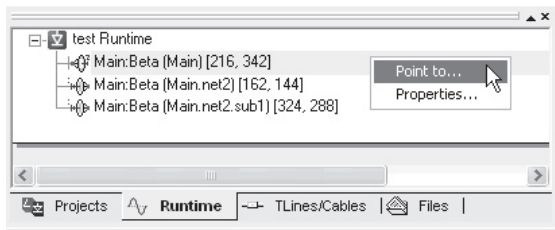
- **Sort By Name:** Select this option to sort all Runtime objects within a Group by name. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.
- **Sort By Type:** Select this option to sort all Runtime objects within a Group by type. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.

In this view, all Controls or Output Channel objects (and corresponding Observers) are organized according to a specified Runtime Group name. Each group can contain any Controls or Output Channel object that is described in the Runtime Objects and Observers sections above.

For more information on creating Runtime Groups, see 'Grouping of Runtime Objects' in Chapter 6.

Radio Links View

In Radio Links View, all Radio Links components existing in the Project are listed in alphabetical order. Right-clicking on a Group branch will bring up a pop-up menu:





NOTE: To switch to Radio Links view, right-click on the Project name and select **View Radio Links**. See the beginning of this topic for details.

Chapter 4: The PSCAD Environment

The following list describes the functions available in this pop-up menu:

- **Point to...:** PSCAD will point to the location of the corresponding Radio Link component within the Project.
- **Properties...:** Select this option to edit the properties of the corresponding Radio Link component.

Icons are included for both Radio Link types. These are listed below:



-  Transmitter
-  Receiver

TLines/Cables

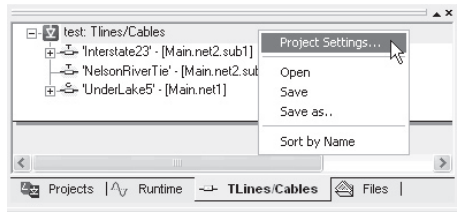
The TLines/Cables section is provided specifically for the organization of Overhead Transmission Line and Underground Cable related objects, specifically Transmission Line/Cable Configuration, Overhead Line Interface, and Cable Interface components. The main trunk of the TLines/Cables tree is a simple list of all such objects, where the objects may be sorted by name.

NOTE: The TLines/Cables Section is available only for the current Active Project. If you would like to see the TLines/Cables details of a Project that is not active, simply set the desired project as active first before clicking the TLines/Cables tab. See 'Setting the Active Project' in Chapter 5 for more details.

Icons are included for easy visual differentiation between TLines and Cables. These are listed below:

-  Overhead Transmission Line
-  Underground Cable

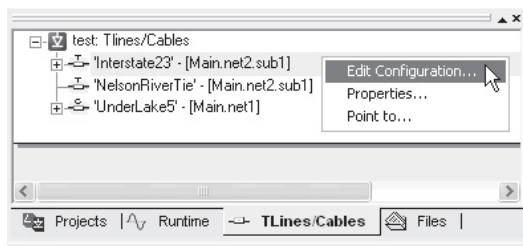
Right-clicking on the Project name will invoke a pop-up menu:



The functions in this menu are as described below:

- **Project Settings...:** Select this option to access the Project Settings dialog for the Project. See ‘Editing Project Settings’ in Chapter 5 for more details.
- **Open:** Select this option to switch to Circuit view.
- **Save:** Select this option to save the Project. See ‘Saving Project Changes’ in Chapter 5 for more details.
- **Save as...:** Select this option to save the Project to another filename. See ‘Saving Project Changes’ in Chapter 5 for more details.
- **Sort By Name:** Select this option to sort all Modules displayed by name. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.

Each branch in the TLines/Cables list represents the Transmission Line/Cable Configuration component for either an overhead line or underground cable transmission system. Right-clicking on a TLines/Cables branch will invoke the pop-up menu:



The following list describes the functions of this menu:

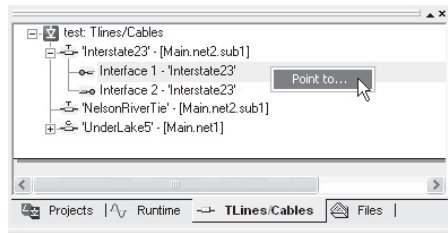
- **Edit Configuration...:** Select this option to access the The T-Line/Cable Properties Editor directly for this Transmission Line/Cable Configuration component.
- **Properties...:** Select this option to access the Transmission Line/Cable Configuration component parameters dialog.

Chapter 4: The PSCAD Environment

- **Point to...:** Select this option to point directly to the associated Transmission Line/Cable Configuration component in Circuit view (same as double-click).

If the line is an overhead line with Termination Style in **Remote Ends** mode, then a '+' box will appear beside the branch name. Expanding this branch (press the '+' symbol) will show the two Overhead Line Interface components representing both ends of the transmission line. Note that cables will always have remote ends, as Direct Connection cables are not supported.

Right-clicking on an Interface object will invoke the pop-up menu:



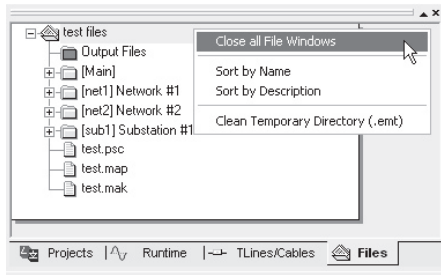
The following list describes the functions of this menu:

- **Point to...:** Select this option to point directly to the associated TLines/Cables object interface in Circuit view (same as double-click).

The Files Section

The Files section is provided specifically for the organization of all files found in the Project Temporary Directory, such as Fortran, Data, Output, etc. files into a convenient tree-based environment. The main trunk of the Files tree is arranged according to existing Modules, with an additional 'Output Files' branch specifically to harbour miscellaneous Output Files unrelated to specific Modules.

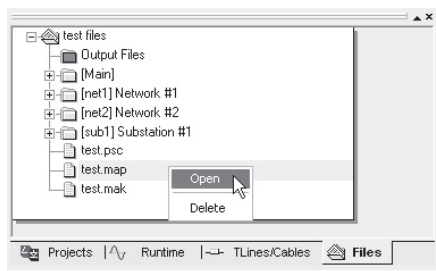
Right-clicking on the Project name will invoke a pop-up menu:



The functions in this menu are as described below:

- **Close all File Windows:** Selecting this option will close all text file windows currently open for viewing. This option will only close windows related to external files, it will not close any open Circuit views.
- **Sort By Name:** Select this option to sort all Modules displayed by name. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.
- **Sort By Description:** Select this option to sort all Modules displayed by description. Continually selecting this option will toggle the alphabetical order from A to Z or Z to A.
- **Clean Temporary Directory (.emt):** Select this option to delete all temporary Project files (i.e. compilation, make, data, etc. files) from the Temporary Directory. See 'PSCAD Temporary Directories' in this chapter for more details.

The files shown on the main trunk are the Active Case Project files themselves (*.psc), along with the corresponding Map (*.map) and Make (*.mak) files. A simple double-click on any file will open that file in the main viewing area of the PSCAD Environment. Right-clicking on the File will invoke a pop-up menu:






The functions in this menu are as described below:

Chapter 4: The PSCAD Environment

- **Open:** Open the selected file for viewing.
- **Delete:** Delete the selected file from the PSCAD Temporary Directory.

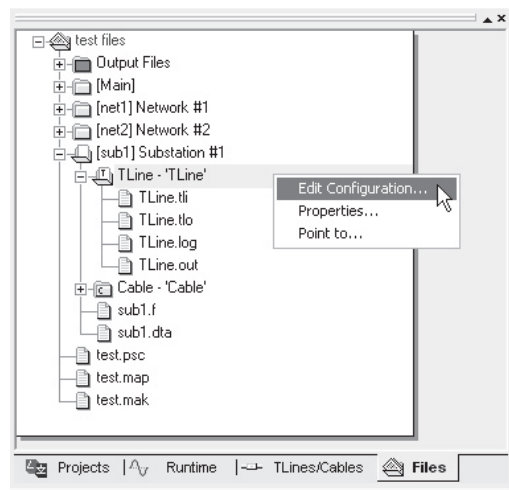
The icons used here are described as follows:

-  Module directory
-  Output files directory
-  File

Transmission Lines and Cable Files

For the most part, Project related files can be viewed according to their parent Module. Once the Project has been compiled, each Module directory will contain a corresponding Fortran (*.f) and a Data (*.dta) file.

If there are any existing transmission lines or cables within the Module, all related line constants files will appear within a separate directory, organized by the transmission system name. In the example below, the Module 'sub1' contains both a overhead transmission line and an underground cable system, in addition to an Fortran and Data file:

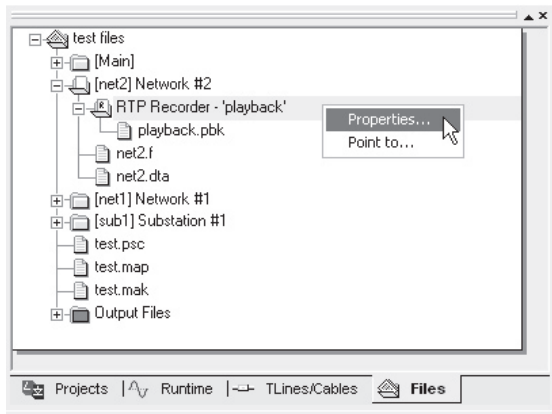


Right-clicking on a transmission line or cable sub-directory will invoke a pop-up menu as shown above. The options in this menu are as described below:

- **Edit Configuration:** Selecting this option will open the The T-Line/Cable Configuration Editor for the corresponding transmission system, directly from the Files section.
- **Properties...:** Select this option to access the input dialog for the corresponding Transmission Line Configuration or Cable Configuration.
- **Point to...:** PSCAD will point to the location of the transmission system in the Project.

RTP/COMTRADE Recorder Files

If there are any existing RTP/COMTRADE Recorder components within the Module, all related files will appear within a separate directory, organized by the RTP/COMTRADE Recorder output filename. In the example below, the Module 'net2' contains an RTP/COMTRADE Recorder component with an output filename labelled as 'test,' in addition to a Fortran and Data file:






Right-clicking on a RTP/COMTRADE Recorder sub-directory will invoke a pop-up menu as shown above. The options in this menu are as described below:

- **Properties...:** Select this option to access the input dialog for the corresponding RTP/COMTRADE Recorder.
- **Point to...:** PSCAD will point to the location of the RTP/COMTRADE Recorder component in the Project.

NOTE: Depending on the settings of the RTP/COMTRADE Recorder component, there could be a number of possible file types appearing in this directory. See the RTP/COMTRADE Recorder component for more details.

Additional icons are provided as follows:

-  Transmission line directory
-  Cable directory
-  RTP/COMTRADE Recorder component directory

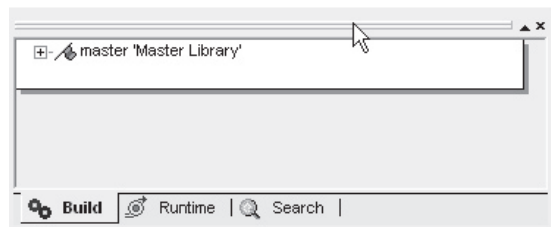
THE OUTPUT WINDOW

Formally known as the ‘Message Tree’ in previous PSCAD versions, the Output window provides an easily accessible interface for viewing feedback and troubleshooting your simulation. All error and warning messages, either issued by a component, PSCAD or EMTDC, can be viewed here - subdivided into either Build or Runtime messages.

The Output window will also provide the results of a project search, using the Find feature, under the Search section tab.

Moving and Resizing the Output Window

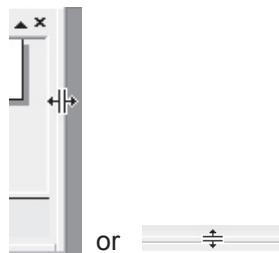
The Output window may be moved and dropped anywhere in the PSCAD environment. To do this, click and hold the left mouse button with the mouse pointer directly over the top window bar.



Drag the window to where you want it placed (you should see a boxed outline of the window) and let go of the mouse button. Depending on where the window outline is when the mouse button is

released, the window will either dock itself into position or appear as a floating window.

The Output window can be resized by moving the mouse pointer over one of the four window edges until the pointer changes that shown below:



Click and hold the left mouse button, and drag the mouse pointer to resize in either the horizontal or vertical direction.

Errors and Warnings

A distinction can be made between error and warning messages simply by the colour of the symbol preceding the message. The colour code is as follows:

-  OK
-  Warning
-  Error

Warning messages are not considered detrimental to the simulation run, and PSCAD will continue to build and run the project regardless of any warnings. However, warnings can indicate areas of the system that, although not technically illegal, will still affect the simulation results (i.e. a mistakenly disconnected node, for instance). It is therefore important to study the Output window every time the project is built and run.

If an error is reported, the simulation build or run will be halted immediately. The user must then study any error messages reported and attempt to determine the problem source. See 'Locating the Problem Source' in this chapter for more details.

Chapter 4: The PSCAD Environment

Build

The Build section provides mainly component and PSCAD based error and warning messages related to the compilation and building of Fortran, data and map files for the project. PSCAD has the capability of detecting a number of different types of system inconsistencies related to this. See 'Common Output Window Messages' in Chapter 11 for more information.

Any warning or error messages defined in the Checks segment of any component definition will be displayed in Build.

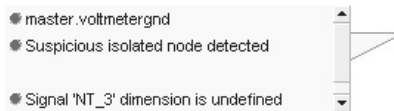
Runtime

The Runtime section provides error and warning messages related to the simulation run - that is, messages from EMTDC. Runtime messages are usually more serious in nature and can involve numerical instabilities and other problems of this type.

It is important to study the messages thoroughly. In some cases, PSCAD will direct you towards the subsystem and node number in the electrical system where the problem is occurring. The Find feature in PSCAD can help point you toward the problem area. Also, see 'Common Output Window Messages' in Chapter 11 for more information.

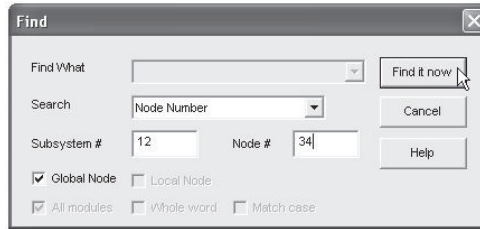
Locating the Problem Source

The Output window provides an easy method of locating the source of any displayed message: Either right-click the message itself and select **Point to Message Source**, or simply double-click the message. PSCAD will then automatically open the source page in Circuit view and point directly at the problem with a message box:



Search

If a message indicates that the problem is arising at a particular subsystem and node, then you can utilize the Find feature in PSCAD to search for this exact location. Simply Invoke the Find dialog and in the **Search** input field, select **Node Number**. Enter the **Subsystem #** and **Node #** indicated in the Output Window and select the **Find it now** button.



The feedback from this search will be displayed in the Search section of the Output window. Simply double-click the search feedback messages and PSCAD will automatically open the source page in Circuit view and point directly at the problem (with the same arrow as described above).

THE DESIGN EDITOR

The Design Editor window is probably the most important part of the PSCAD environment, and is where most (if not all) project design work is performed. The Design Editor is used mostly for the graphical construction of circuits (Circuit view), and also includes an embedded component definition editor.

The Design Editor window is invoked (in Circuit view) by either a left double click on a loaded project title in the Workspace, or by right clicking on the title and selecting **Open**.

Viewing Windows

The Design Editor is divided into six sub-windows. Each of the sub-windows can be accessed by simply clicking on a particular tab on the tab bar at the bottom of the Design Editor. The tab bar is shown below for reference:



As you can see, some of the tabs are initially disabled (greyed-out). This depends on what you are trying to view, as well as whether or not the project has been compiled.

The **Graphic**, **Parameters** and **Script** tabs are used exclusively for component design, and will not be enabled unless editing a component definition. The Graphic window will also be enabled while viewing a module page. You can edit a component definition by a

Chapter 4: The PSCAD Environment

Ctrl + left double click on the component, or right clicking on the component and selecting **Edit Definition...**

Circuit

The Circuit window is always the default view when a project is first opened. This is where the majority of design work is done while working with PSCAD, and is where all control and electrical circuits are constructed.

While the Circuit window is open, the Control Palette and Electrical Palette toolbars are enabled.

Graphic

The Graphic window is used for editing the graphics of a component definition or module.

Parameters

The Parameters window is used for editing the parameters of a component definition.

Script

The Script window is used for editing component definition code.

Fortran

The Fortran window is a simple text viewer that allows easy access to the EMTDC Fortran file corresponding to the module currently being viewed in the Circuit window. For example, if you are viewing the main page of a project in Circuit view, the Fortran window (click the **Fortran** tab) will show the EMTDC Fortran file corresponding to the main page.

Data

The Data window is a simple text viewer that allows easy access to the EMTDC input data for any existing electric network, corresponding to the module being viewed in the Circuit window. For example, if you are viewing the main page of a project in Circuit view, the Data window (click the **Data** tab) will show the EMTDC input data for the main page.

OTHER IMPORTANT FEATURES

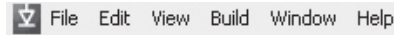
Menus

There are a couple of different types of menus available in PSCAD. The following sections provide a description of each.

PSCAD

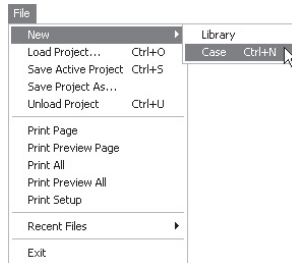
Main Menu Bar

The Main Menu bar contains an assortment of the most common PSCAD commands. It is located near the top of the PSCAD Environment, appearing as shown below:



This bar may appear differently depending on what section of the PSCAD Environment you are currently in.

To select a menu item, move the mouse pointer onto that item (the item becomes raised) and then click the left mouse button. The following menu shows how to create a new project using the **File** menu.



You will notice that some menu items, **New** in the **File** menu for instance, contain an arrow symbol pointing to the right. This indicates that there are sub-menu listings related to that menu item, as shown above.

You may also notice that some menu items are followed by a triple-period, for instance the **Load Project...** menu item in the **File** menu. This signifies that a dialog box will open if this command is invoked.

NOTE: The Main Menu bar contains only some of the standard PSCAD functions. There are many others available using right-click pop-up menus.

Right-Click Pop-Up Menus

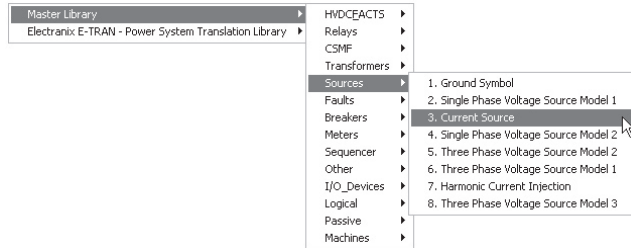
PSCAD V4 contains an extensive set of right-click pop-up menus for easy support of location specific commands and tasks. Depending on the location of your mouse pointer, you can invoke pop-up menus almost anywhere in the environment by simply pressing your right

Chapter 4: The PSCAD Environment

mouse button. The menus that pop-up will always contain commands, which are specific to the area you are currently working in.

Library Pop-Up Menus

Library pop-up menus are designed to provide an organized method of accessing desired Component Instances located in the Master Library, or any other user-defined Library Project. This menu system is invoked by pressing **Ctrl + right mouse button**, while the mouse pointer is over a blank section of any project page in Circuit view.



NOTE: It is a good idea to always provide an adequate description in user-defined components, so as to avoid confusion while using these menus. If a description is not provided, the Library Pop-Up Menu will display the actual filename of the component.

Library pop-up menus display whatever is entered into the description field of the component definition. In the above image, the user has two Library Projects currently loaded in the Workspace: The Master Library and another library with 'Electranix E-TRAN...' entered into the Project description field. The mouse pointer is pointing towards the Current Source component in the 'Sources' module, in the Master Library. A left-click at this point will add the current source component to whichever project page is open in Circuit view.

In general, library pop-up menus construct a system of menus according to the hierarchal structure of any loaded library projects. As a default, the Master Library will always be displayed first, and any user-defined library projects will then be placed in the order as they appear in the Workspace. If components are located within a module in a specific library project, the library pop-up menu will automatically nest these components into another sub-menu. This sub-menu will be named according to the contents of the corresponding module description field.

PSCAD

Here are some key points to remember when using library pop-up menus:

- Component instances, located on the main page of the Master Library, will not be included in the pop-up menu. However, such instances will be included for user-defined library projects.
- Duplicate component instances in a module will be ignored.
- Duplicate component instances in multiple modules will be added to the menu multiple times.
- Modifications to a library project will not appear in the library pop-up menu until the library project has been reloaded.








Toolbars

Toolbars are useful in any program to provide a quick and easy method to access menu functions. PSCAD V4 now includes multiple toolbars, which contain the most commonly used menu tasks.





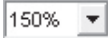







The new PSCAD drawing related toolbars allow you to add the most common drawing objects and components by simply using the left mouse button: A single click on the toolbar to select an object, and a single click to add the object to the drawing canvas.

Main Toolbar

The Main Toolbar resides near the top of the PSCAD environment. This bar primarily contains tasks found in the Main Menu bar, but also includes others, as well. The individual toolbar buttons are listed below with a short description:

Button	Description
	Creates a new Case Project (*.psc file)
	Loads a case project (*.psc file)
	Saves changes to the Active Project (*.psc file)
	Print
	Cuts the current selection to the Windows clipboard
	Copies the current selection to the Windows clipboard
	Pastes from the Windows clipboard

Chapter 4: The PSCAD Environment





	Undo
	Redo
	Zoom in one step
	Zoom out one step
	Zoom control list box
	Navigate back to last view (select the down arrow for history)
	Navigate forward to last view (select the down arrow for history)
	Up on module level
	Invoke Wire Mode
	Create a new default Module
	Create new component
	Launches Find feature

Status Bar

The Status bar is located at the very bottom of the PSCAD Environment, and is not actually a toolbar, but used for the purpose of display only (does not contain any command buttons). Most of the time, the Status bar is used to monitor the status of a simulation build, link and run, and will display the percentage complete and the simulation time during a run.

Rotation Bar











The individual Rotation bar buttons are listed below with a short description:

Button	Description
	Rotate selection counter-clockwise
	Rotate selection clockwise
	Mirror selection
	Flip Selection

PSCAD






Runtime Bar

The individual Settings bar buttons are listed below with a short description:












Button	Description
	Compile (build) all modules (active project only)
	Make project (active project only)
	Run simulation (active project only)
	Stop simulation (active project only)
	Pause simulation (active project only)
	Advances run by one time step (while pause is invoked)
	Take a snapshot
	Change PSCAD Plot Step
	Control settings menu button
	Control settings template list

Electrical Palette

The Electrical palette contains common electrical components used when building circuits in Circuit view. Most of these button functions can also be found under **Add New Component** in the Circuit view right-click pop-up menu. The individual toolbar buttons are listed below with a short description:








Button	Description
	Add Wire
	Add Resistor
	Add Inductor
	Add Capacitor
	Add Ground












Chapter 4: The PSCAD Environment

	Add Node Label
	Add External Node (Xnode)
	Add Breakout
	Add Pin
	Add Current Meter
	Add Voltmeter
	Add Voltmeter with ground
	Add a T-Line Configuration
	Add a T-Line interface
	Add a Cable Configuration
	Add a Cable interface

Control Palette








The Control palette contains common control components used when building circuits in Circuit view. Most of these button functions can also be found under **Add New Component** in the Circuit view right-click pop-up menu. The individual toolbar buttons are listed below with a short description:

Button	Description
	Add Data Tap
	Add Data Merge
	Add Data Label
	Add Integer Constant
	Add Real Constant
	Add Import
	Add Export






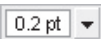


	Add Radio Link
	Add Output Channel
	Add Slider
	Add Switch
	Add Dial
	Add Pushbutton
	Add Graph Frame
	Add Phasor Plot
	Add XY Plot
	Add Control Panel
ab	Add Annotation Box
	Add Sticky Note

Graphic Palette

The Graphic palette contains functions used only when editing component definitions in the Graphic section of the Design Editor. The individual toolbar buttons are listed below with a short description:








Button	Description
	Reload changes from last save
	Return to Circuit view
	Set layers
	Show/Hide all layers
	Add Line
	Add Rectangle
	Add Ellipse

Chapter 4: The PSCAD Environment

	Add Arc (90°)
	Add Arc (180°)
	Add Text Label
	Add Connection
	Default line colour
	Default line weight
	Default line style
	Default fill style

Graphic Filter Bar

The Graphic Filter toolbar is used specifically within the Graphics section of the Design Editor. These buttons provide a quick means to show or hide certain types of Graphic Objects, so as to create a less cluttered environment in which to work.

















Button	Description
	Show/hide ellipses
	Show/hide arcs
	Show/hide lines
	Show/hide rectangles
	Show/hide text
	Show/hide connections
	Show/hide all graphics

Parameters Bar

The Parameters bar contains functions used only when editing component definitions in the Parameters section of the Design

PSCAD






Editor. The individual toolbar buttons are listed below with a short description:

Button	Description
	Reload changes from last save
	Return to Circuit view
	Set layers
	Search for symbol names
	Space Fields evenly (See Field Alignment)
	Make Fields same width (See Field Alignment)
ab	Add new Text Field
re	Add new Input Field
	Add new radio button Choice Box
	Add new drop list choice box
Configuration ▾	Navigate Category pages
	Add new Category
	Duplicate current Category
	View Category properties
	Delete Category
	Move Category up one level
	Move Category down one level
	Move Category to top
	Move Category to bottom

Chapter 4: The PSCAD Environment

Script Bar

The Script bar contains functions used only when editing component definitions in the Script section of the Design Editor. The individual toolbar buttons are listed below with a short description:

Button	Description
	Reload changes from last save
	Return to Circuit view
	Search for text
	Search for and replace text
<input type="text" value="Computations"/>	Navigate segments drop list
	Segment Manager

Keyboard Shortcuts

There are many keyboard shortcuts (or hotkeys) available in order to make the design process more efficient. Hotkeys help to reduce the 'amount of clicks' required for performing a specific task.

The following tables list the available keyboard shortcuts in PSCAD V4. See 'What's New in PSCAD V4?' in Chapter 1 for changes to keyboard shortcuts since PSCAD V3.

General

Note that 'Ctrl' may be dropped where indicated (i.e. [Ctrl +]) if **Enable Cut/Copy/Paste without 'Ctrl' Key** is selected. See the section entitled 'Views' in Chapter 3 for more.

Shortcut	Description
[Ctrl +] X	Cut selection
[Ctrl +] C	Copy selection
[Ctrl +] V	Paste selection
A	Select all

PSCAD

R	Rotate selection right
L	Rotate selection left
M	Mirror selection
F	Flip selection
S	Set component sequence
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + F	Find function (search)
Ctrl + O	Load project
Ctrl + N	New project
Ctrl + S	Save active project
Ctrl + G	Global Constants dialog
Ctrl + U	Unload selected project
Ctrl + W	Invoke/Cancel Wire Mode
Esc	Cancel action
+	Zoom in
-	Zoom out
Ctrl + Shift + Left Mouse Hold	Pan (dynamic scroll)
Ctrl + Left Double Click	Edit definition of selected component
Ctrl + Right Click	Invoke right-click menu system
Left Double Click	Edit properties of selected component
Backspace	Navigate up one module level

Chapter 4: The PSCAD Environment

F5	Refresh canvas
F1	Help
← → ↑ ↓	Scroll Circuit page horizontally and vertically

Wires

To apply any of the following shortcuts, simply move your mouse pointer over a Wire.

Shortcut	Description
V	Insert a Wire Vertex
I	Reverse Wire Vertexes
D	Decompose Wire

Plotting

To apply any of the following shortcuts, simply move your mouse pointer over a plot area. Note that in some instances, the graph must be selected.

Shortcut	Description
Insert	Insert an Overlay Graph
+	Zoom in to graphs
-	Zoom out of graphs
P	Zoom previous
N	Zoom next
X	Zoom x-axis extents
E	Zoom x-axis limits
Y	Zoom y-axis extents
U	Zoom y-axis limits
R	Reset all extents
B	Reset all limits

PSCAD

Ctrl + Left Mouse Hold	Zoom horizontal aperture
Shift + Left Mouse Hold	Zoom vertical aperture
Left Mouse Hold	Zoom to box (simultaneous horizontal and vertical)
G	Toggle grid lines
I	Toggle tick marks
K	Toggle curve Glyphs
M	Show Markers
X	Set X marker
O	Set O marker
L	Toggle marker lock-step
F	Toggle frequency/delta view
Q	Show x-intercept
W	Show y-intercept
C	Show cross-hairs (follows curve traces)
Space Bar	Switch curves while in cross-hair mode
← →	Graph frame dynamic aperture adjustment

Window Settings

PSCAD V3 users will notice a difference in the window settings for V4. That is, all environment windows are now docked (i.e. embedded) rather than floating by default. Docked windows, although embedded, may still be manipulated in the standard ways. They may be moved, stretched, maximized, minimized and turned off quite easily.

Tab Navigation

Another 'dimension' was added to the navigational abilities of PSCAD in V4: Tabs. Tabs allow you to view several aspects of a project, by simply clicking tab buttons (usually located at the bottom of the window). For example, once a project is compiled, you can quickly look at project related files (i.e. Circuit, Fortran, Data, etc.) with a single click.



Chapter 4: The PSCAD Environment

Tab navigation is also available in the Output window, the Design Editor, as well as other areas.

PSCAD TEMPORARY DIRECTORIES

When a Case Project is compiled, several files are created and are placed into a temporary directory located in the same directory as the project file (*.psc) itself. The directory is named by appending an '.emt' extension to the project filename. For example, if a case project has a filename 'test.psc,' a temporary directory called 'test.emt' will be created when the project is compiled. The temporary directory will contain all files created by both the Fortran compiler and PSCAD.

All files located in the temporary directory can be viewed in an organized manner in the Files section of the Workspace.

You may also clean out the entire *.emt directory at any time (see 'Cleaning the Temporary Directory' in Chapter 5). Removing compiled files from the temporary directory forces PSCAD to recompile everything from scratch.

PSCAD ON-LINE HELP SYSTEM

PSCAD V4 boasts an entirely new and improved on-line help system. Based on the content of both the PSCAD V2 manuals and the PSCAD V3 on-line help, the new system has been brought up-to-date and includes new information previously unavailable.

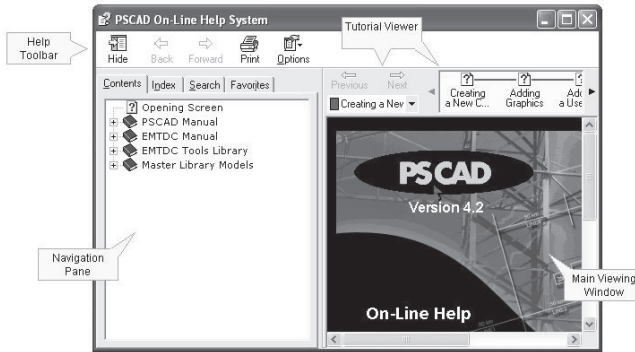
The new on-line help is no longer a collection of HTML documents as it was in V3, but a single compiled file in Microsoft HTML Compiled Help format. This single file includes it's own browser, and as such, a standalone web browser is no longer required for viewing. In addition, features previously unavailable, such as a **full-text search engine** and comprehensive **index** are now at the user's disposal.

In addition to the complete PSCAD V4 manual set, the on-line help file includes component specific documentation (not included in the manuals). See 'Accessing the Online Help System' in Chapter 5 for details on accessing help.

PSCAD

Look and Feel






When the help system is invoked (by pressing the F1 key), a window similar to that shown below should appear.



There are three main sections of the help system. Across the top is the Help Toolbar, and below this are the Navigation Pane on the left, and the Main Viewing Window on the right.

The Help Toolbar

The Help toolbar contains the most commonly used functions. These are summarized below:

Button	Description
 Hide	Hides navigation pane
 Back	Moves backward in the viewed topic list
 Forward	Moves forward in the viewed topic list
 Print	Prints currently viewed topic
 Options	Provides some options to allow the user to customize the online help viewer

The Navigation Pane

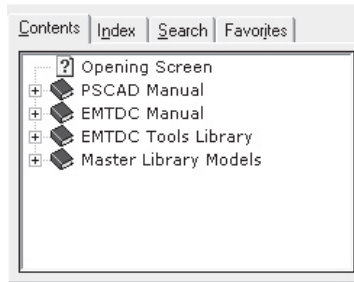
The Navigation Pane is separated into four sections: The Table of Contents (TOC), the Index, the Search engine and the Favorites

Chapter 4: The PSCAD Environment



section. To access any section, click the corresponding tab at the top of the pane.

Contents

The Contents tab contains the help system Table of Contents. The TOC is organized using a standard tree-type structure. To expand or collapse any branch of the tree, left-click on the corresponding [+] or [-] icons, directly to the left of each branch, respectively.



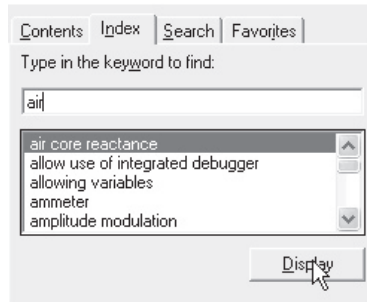
There are two icons associated with the TOC: Books and Topics. A Book simply contains a selection of Topics:

-  Book
-  Topic

Index

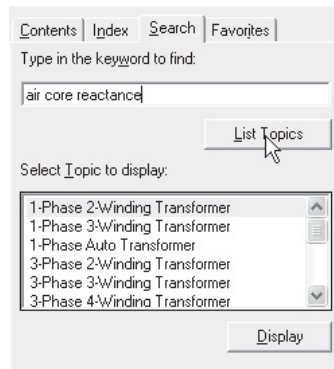
The index contains a large list of keywords, listed in alphabetical order, that have been associated with various topics within the help system. You have the choice of either scrolling through this list directly, or you may type in a word into the input field near the top of the index pane, in order to directly look up a word.

In the figure below, for example, a user is looking up 'air core reactance.' Entering 'air' will bring the list window to the nearest matching word.



Search

The Search section allows you to enter in a single word or a text string, and then list the topics that contain the word or string. Simply enter the text string and click the **List Topics** button. Any topics containing the string will be displayed in the output field, as shown below.



NOTE: The Help System search engine will search all text in each topic. Text that exists within embedded images will not be considered.

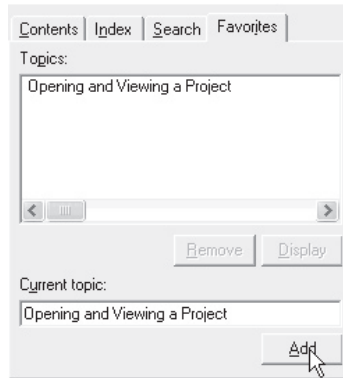
To bring up a listed topic into the main viewer, simply left double-click, or select it and click the **Display** button.

Favourites

The Favourites section allows users to bookmark their favourite help pages. To add a favourite topic to the favourites list, simply left click the **Add** button. To remove a topic from Favourites, select it and left

Chapter 4: The PSCAD Environment

click the **Remove** button. To display a topic from the favourites list, select it and left click the **Display** button.



The Main Viewer

The contents of the main viewing window are interactive in the same way as in a normal HTML web page. Depending on the page viewed, there may be hyperlinks and other types of functionality, normally seen while navigating the web. Usually, text with associated hyperlinks and other functions are shown in blue color and may or may not be underlined.

TUTORIAL PROJECTS

The PSCAD software package is installed complete with a directory of tutorial projects, which contains a variety of simple cases to illustrate various features of PSCAD. This directory is located within the PSCAD installation directory under ...examples\tutorial.

The tutorial projects described in this section are mainly meant to illustrate the use of PSCAD and hence, they are simple cases from an electrical engineering point of view. If you are a first time user, go through all the tutorial cases in the order they are listed below.

Once you are familiar with these projects, have a look at the more detailed projects contained within the main examples directory. These include a variety of practical examples ranging in topics from machines to FACTS devices.

PSCAD

Voltage Divider

vdiv_1.psc A simple voltage divider circuit with a resistor and a resistive source. Demonstrates how to assemble a circuit, monitor voltage and current, and run the simulation.

Fast Fourier Analysis

fft.psc Shows the use of Fast Fourier analysis component to perform online fft on signals.

Simple AC System with a Transmission Line

simpleac.psc A simple AC system with transmission lines. Introduces transformers, transmission lines, and the concept of subsystems in PSCAD.
simpleac_sld1.psc
psc
simpleac_sld2.psc
psc

Use of Control Arrays

pagearray.psc Demonstrates the use of control arrays and how to export electrical nodes to other pages so that a circuit can be modeled on multiple pages - even if you do not have transmission lines.

Use of Slider, Switch, Button, and Dial

inputctrl.psc Shows the use of dynamic input devices: Slider, Switch, Push Button and the Dial.

Interpolation

interpolation.psc A simple case illustrating the use of interpolation.

Generating a Legend and using PSCAD Macros

legend.psc Shows how to generate a legend. Also illustrates the use of PSCAD macros in a legend.

Chatter Elimination

chatter.psc Defines chatter and gives examples on chatter elimination techniques.

Chapter 4: The PSCAD Environment

Multiple Run

multirun.psc

multirun_sld.psc

A simple example to demonstrate the use of multiple run features to find the maximum overvoltage due to a fault in a 3-phase power system. The point on wave and type of fault is varied to determine the worst-case overvoltage.

Features and Operations

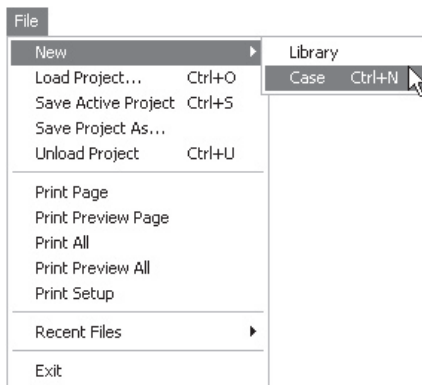
This chapter provides a resource for basic operations (or ‘clickology’) and features in PSCAD. Whether you are a new user, or a seasoned PSCAD veteran, the following topics can be very helpful in getting you familiar with PSCAD V4.

This chapter includes a tutorial entitled Creating a New Project, which is designed to jump start the learning process and to get you using PSCAD. If you have not done so already, try the tutorial entitled My First Simulation, as well. This chapter is divided into three main sections: Projects, Components and Drawing.

PROJECTS

Creating a New Project

Click the left mouse button on the menu item **File** in the Main Menu bar - a drop down menu should appear. With your mouse pointer, select **New**. A secondary menu should open listing a choice of either a **Library** project or a **Case** project.



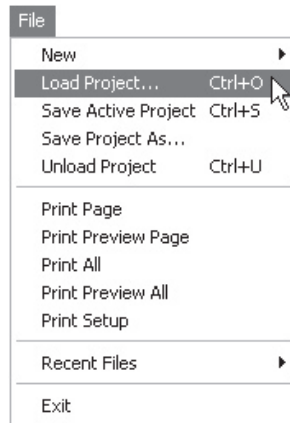
Select the desired project type. A new project entitled ‘noname’ should then appear in the Workspace window, indicating that a new project was created.

Chapter 5: Features and Operations

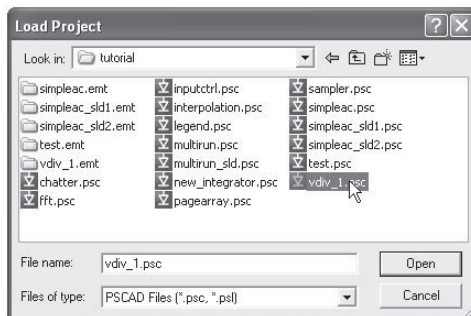
You can also create a new case project by simply pressing the **New Project** button in the Main Toolbar, or by pressing **Ctrl + N** on your keyboard.

Loading a Project

Click the left mouse button on the menu item **File** in the Main Menu bar - a drop down menu should appear. With your mouse pointer, select **Load Project....**



A dialog window should open with a default **Files of type** as **PSCAD Files (*.psc, *.psl)** for PSCAD case project and library project respectively. Navigate to the desired project and select it so that its name appears in the **File name** field, as shown below.



Click the **Open** button to open the project. The project name should then appear in the Workspace window, indicating that the project was loaded.

PSCAD

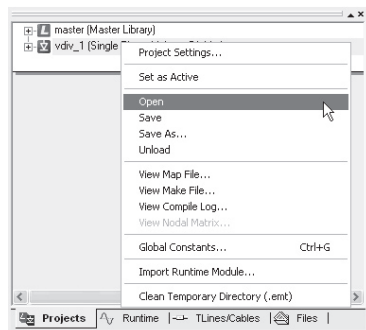
You can also bring up the Load Project dialog by simply pressing the **Load Project** button in the Main Toolbar, or by pressing **Ctrl + O** on your keyboard.

Recent Files

You can also load previously loaded projects by using **Recent Files**. Simply hold the mouse pointer over the **Recent Files** menu item and a list of previously loaded and saved projects should appear. Select the desired project from the list. The number of projects listed can be adjusted from 0 to 16 in the Projects section of the Workspace Settings dialog.

Opening and Viewing a Project

Right-click on the desired project in the Workspace window and select **Open**. You can also open a project for viewing, with a left double-click on the project title in the Workspace window.



Once opened, the main page of the project should appear in the Circuit window of the Design Editor.

Navigating Through an Open Project

Once a project has been opened in Circuit view, there are many navigational features available to help you efficiently navigate about the project.

Scroll Bars

Standard vertical and horizontal scroll bars are available in all Design Editor windows. These are located at the right-most and bottom-most edges of the open window respectively.



Chapter 5: Features and Operations

Arrow Keys

You can use the arrow buttons on your keyboard to scroll both horizontally and vertically while in Circuit view.

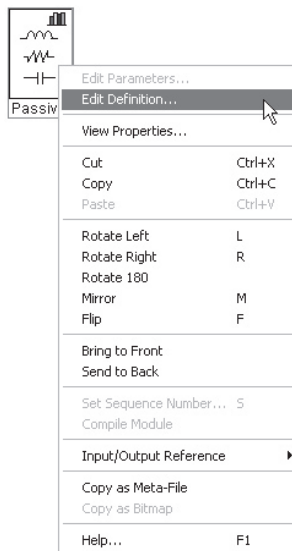
Panning (Dynamic Scroll) Mode

The panning or dynamic scroll feature allows you to scroll through the Circuit or Graphic views in a fluid motion. You can invoke panning mode by one of the following methods:

- On a blank portion of the page, press the Ctrl and Shift keys at the same time, then click and hold the left mouse button (**Ctrl + Shift + left mouse hold**). Moving the mouse will then allow panning through the page.
- Press the **Pan** button in the Main Toolbar to invoke panning mode. To indicate that you are in panning mode, the mouse pointer will change into a hand shape, as shown to the right. Press **Esc** to cancel Pan Mode.

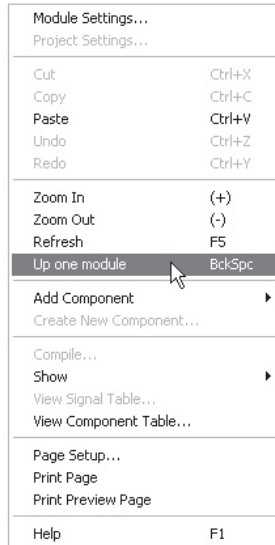
Moving In and Out of Modules

To move into or open a module in Circuit view, either left double-click directly over the module, or right-click on the module and select **Edit Definition...**



To move out of the current module (i.e. move back one level), press either the **Backspace** key on your keyboard, press the **Up one mod-**

ule button on the Main Toolbar, or right-click on a blank part of the module page and select **Up one module**.

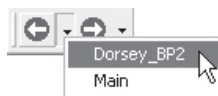


Forward/Back Buttons

PSCAD maintains a navigational history while moving into and out of modules in a project. This history is presented in the form of **Forward** and **Back** buttons on the Navigation Bar, which should be located near the top of the PSCAD Environment. If you cannot see this bar, select **View | Navigation Bar** in the PSCAD Main Menu.



Once a navigational history is started, you may then use the **Forward** and **Back** buttons to navigate into and out of modules, or to jump directly to previously viewed modules. To access the navigational history, press the down arrow on either the **Forward** or **Back** button as shown below:



Tabs

The new tab features in PSCAD V4 add an extra dimension to navigation. That is, the tabs allow you to jump from one viewing area

Chapter 5: Features and Operations

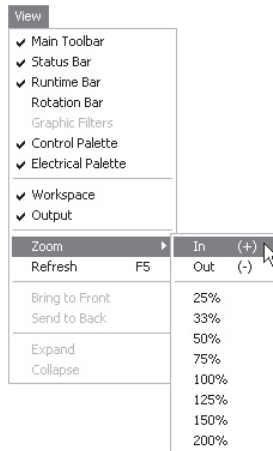
to another by simply selecting a specific tab. There is a tab bar included at the bottom of the Design Editor window (as shown below), as well as on the Output window and others.



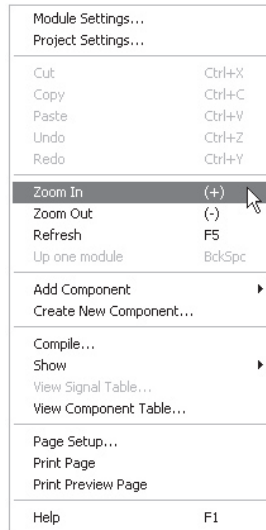
Zooming

Zoom features are available when working in either the Circuit or the Graphic sections of the Design Editor. There are a few different methods for zooming available:

- From the main menu bar, select View | Zoom. You then have a choice to select either In, Out or a specified zoom percentage.



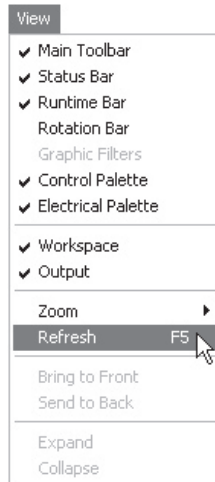
- From the Main Toolbar, select either the **Zoom In** or **Zoom Out** buttons, or select a percentage zoom directly from the **Zoom In/Out** drop down list.
- Press the **+** or **-** keys on your keyboard.
- Right-click on a blank part of the page and select **Zoom In** or **Zoom Out** from the pop-up menu.



Refresh

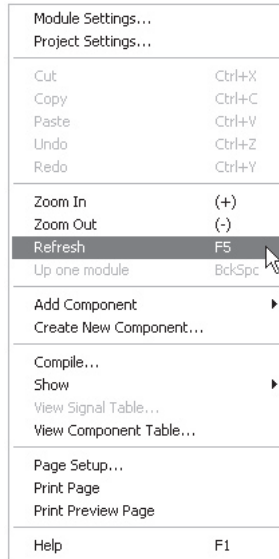
You can refresh the page view in either the Circuit or Graphic windows by one of the following methods:

- From the Main Menu bar, select **View | Refresh**.



- Press the **F5** key on your keyboard.
- Right-click on a blank part of the page and select **Refresh**.

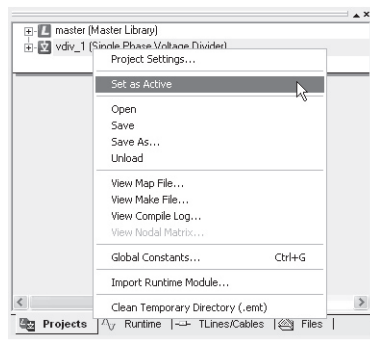
Chapter 5: Features and Operations



Setting the Active Project

When there is more than one case project loaded simultaneously in the Workspace window, PSCAD needs to know which project to run when a simulation is requested. This is accomplished by setting the 'Active Project.'

In the Workspace window, right-click on the title of an inactive project and select **Set as Active**.



Compiling and Building a Project

Before compiling and building a Project, ensure that the desired project is the Active Project in the Workspace window.

NOTE: Running a Project without compiling is perfectly fine. If a Run command is issued, PSCAD will perform the **Compile Modified Modules** and then the Make procedures before the run starts. See 'Running a Simulation' in this chapter for more.

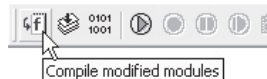
Compile

Compiling refers to the recreation of Fortran, Data, and Map files for a the Project. You may choose to compile the entire Project, or only Modules that have been modified since the last compile. The latter choice comes in handy for very large Projects, which may take minutes to compile all.

To compile the entire Project from scratch, select the Compile All Modules button in the Main Toolbar:

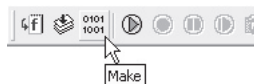


To compile the only Modules that have been modified since the last compile, select the Compile Modified Modules button in the Main Toolbar:



Make

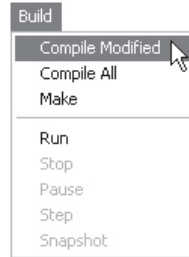
Making the Project refers to the assembly of the associated executable file (*.exe) to be run by EMTDC. To make your Project, press the Make button.



NOTE: If there are any problems with compilation or making your project, these errors will appear in the Output Window. You can locate the source of any of the messages through a simple left double-click on the message itself. PSCAD will point to the source with an arrow icon.

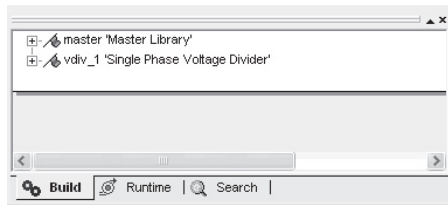
Chapter 5: Features and Operations

You may also perform these functions by selecting Build | Compile Modified, Compile All or Make in the Main Menu.



Viewing Errors and Warnings

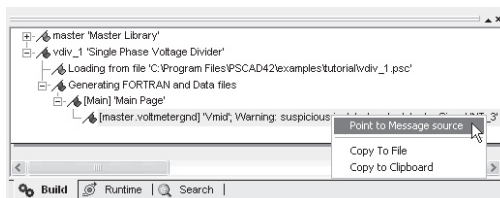
If a problem is detected when a simulation is run, all error and warning messages will appear in the Output window.



The Output window is divided into three main areas: Build, Runtime and Search. Compilation messages will appear under Build, while EMTDC runtime messages will appear in Runtime. The Search area is used to view results from the Find Feature.

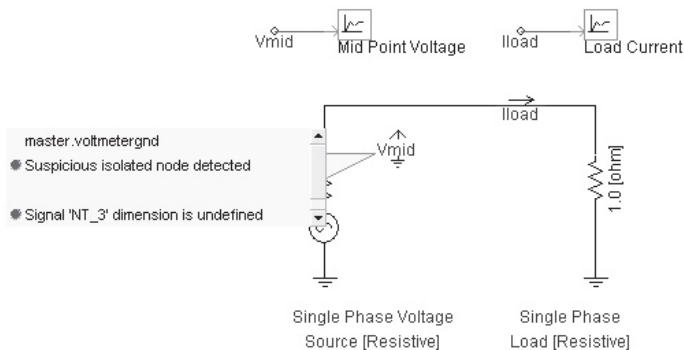
Finding the Message Source

If an error or warning message is received, PSCAD can automatically point to the source of the problem. In the Output window, navigate to the error or warning message, hold the mouse pointer over the message itself and either left double-click or right-click and select **Point to Message Source**. PSCAD will point to the problem source in the Circuit window of the Design Editor.



PSCAD

In the example image below, it can be seen that the voltmeter 'Vmid' has been shifted and is not measuring a specific node voltage.



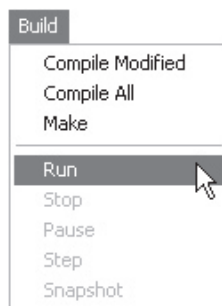
Running a Simulation

Before running a simulation, ensure that the desired project is the Active Project in the Workspace window. Use either of the following methods:

- Press the Run button in the Main Toolbar.



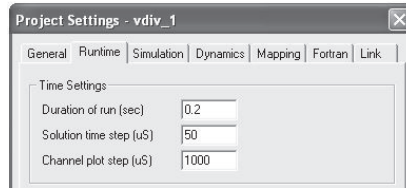
- In the Main Menu, select Build | Run.



Chapter 5: Features and Operations

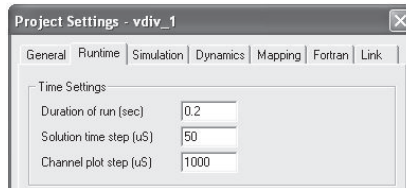
Changing the Run Duration

Open the Project Settings dialog window as described in Editing Project Settings. Click the Runtime tab and, in the Time Settings area, enter a new Duration of Run time in seconds.



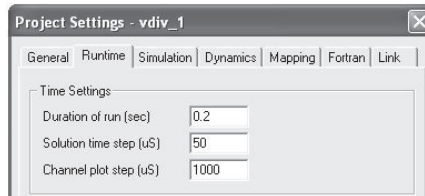
Changing the Simulation Time Step

Open the Project Settings dialog window as described in Editing Project Settings. Click the Runtime tab and, in the Time Settings area, enter a new Solution Time Step time in microseconds (μs).



Changing the Simulation Plot Step

Open the Project Settings dialog window as described in Editing Project Settings. Click the Runtime tab and, in the Time Settings area, enter a new Channel Plot Step time in microseconds (μs).

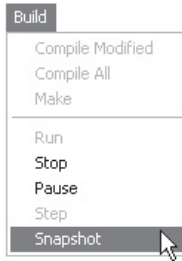


Taking a Snapshot

Taking a snapshot of a simulation run is presently the only way to start PSCAD from an initialized condition. This can be accomplished by one of several methods. See 'Initialization and Initial Conditions' in Chapter 2 of the EMTDC Manual for more details on Snapshot Files.

Manual

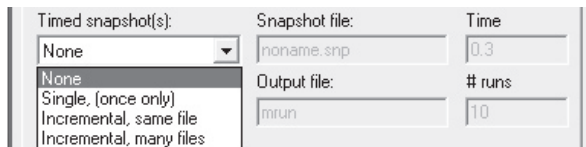
You can manually take a snapshot by either pressing the Snapshot button on the Main Toolbar, or selecting Build | Snapshot from the Main Menu bar, while the project is running.



Pre-Defined

Pre-defined snapshots are set before the run is started. There are a couple of different pre-defined snapshot types available.

Open the Project Settings dialog window as described in Editing Project Settings. In the Runtime section of the Project Settings dialog, choose the snapshot type from the Timed Snapshot(s) drop list.



Enter a name for the snapshot file in the Snapshot File field, and the exact time at which to take the snapshot during the next run in the Time field.

Click the OK button to exit the Project Settings dialog and save changes. Run the project to completion.

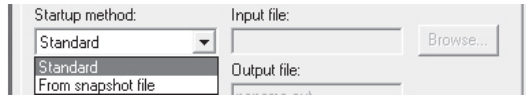
NOTE: Make sure that the run duration is greater than the snapshot time, or no snapshot file will be created!

Starting from a Snapshot

Open the Project Settings dialog window as described in Editing Project Settings. In the Runtime section of the Project Settings dialog, choose From snapshot file in the Startup method drop list. Enter

Chapter 5: Features and Operations

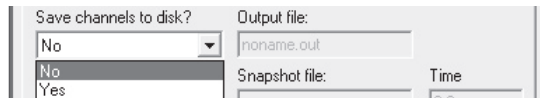
the name for the snapshot file in the Input File field or use the Browse... button to select the file.



Click the **OK** button to exit the Project Settings dialog and save changes. Run the project - it should start initializing from the snapshot time. See 'Initialization and Initial Conditions' in Chapter 2 of the EMTDC Manual for more details on Snapshot Files.

Saving Output to File

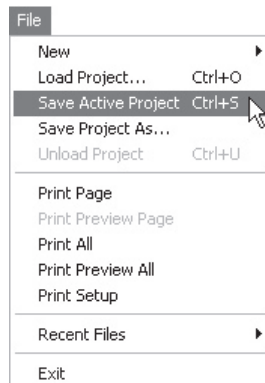
Open the Project Settings dialog window as described in Editing Project Settings. In the **Runtime** section of the Project Settings dialog, choose **Yes** from the **Save channels to disk?** drop list. Enter the name for the output file in the **Output File** field.



Click the **OK** button to exit the Project Settings dialog and save changes. Run the case to create the output file. See 'EMTDC Output Files' in this chapter for more details on Output File format.

Saving Project Changes

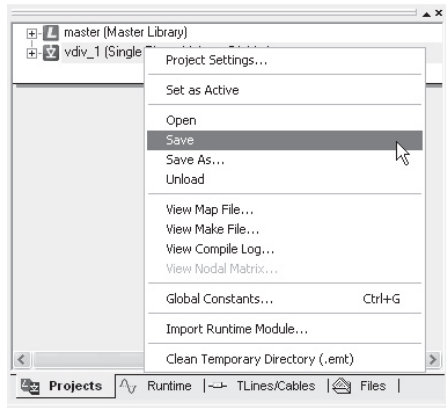
To save the Active Project, click the left mouse button on the menu item **File** in the Main Menu bar - a drop down menu should appear. With your mouse pointer, select **Save Active Project....**



PSCAD

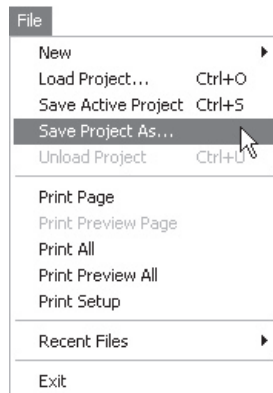
You can also save the Active Project by pressing the **Save Active Project** button in the Main Toolbar, or by pressing **Ctrl + S** on your keyboard.

To save a selected project (either active or inactive), right click on the project title in the Workspace window and select **Save** from the pop-up menu.



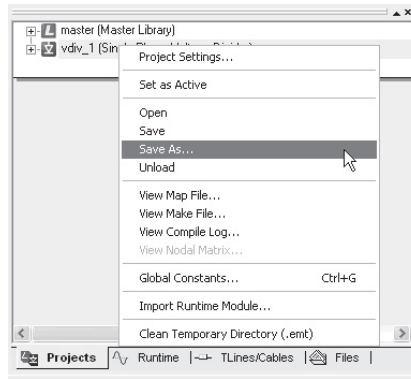
Save Project As...

To save the Active Project as another file, click the left mouse button on the menu item **File** in the Main Menu bar - a drop down menu should appear. With your mouse pointer, select **Save Project As...**

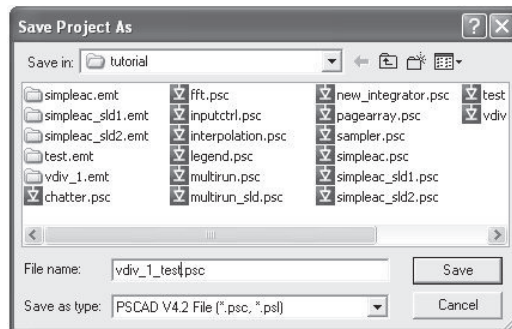


To save a selected project (either active or inactive) as another file, right click on the Project title in the Workspace window and select **Save As...** from the pop-up menu.

Chapter 5: Features and Operations



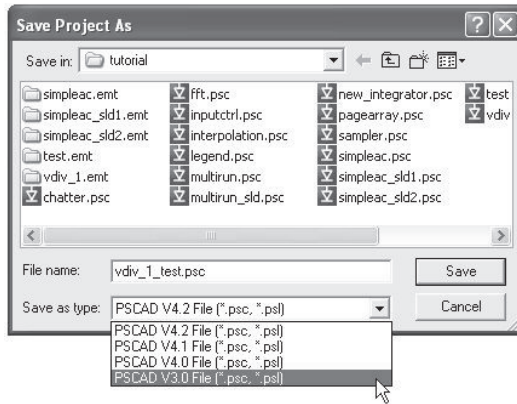
In either case, a dialog window should open with a default **Save as type** as **PSCAD V4 Files (*.psc, *.psl)** for PSCAD V4 case project and library project respectively. Change the file name of the project in the **File name** field, as shown below.



Click the **Save** button to save the project under a different file name. The project name should then change in the Workspace window, indicating that the project was renamed.

Saving as an Older PSCAD Format

Follow the same procedure outlined in the Save Project As... section above. When the Save As dialog window appears, go to the **Save as type** drop list and select **PSCAD Vx.x File (*.psc, *.psl)**.

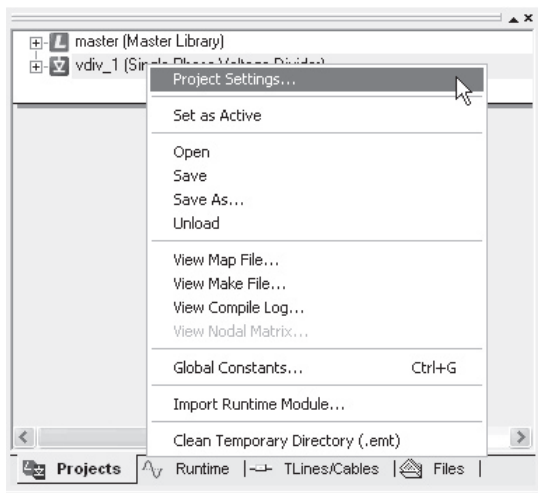


NOTE: Some components available in the PSCAD V4 Master Library are not available in PSCAD V3 and will be lost upon conversion. Also, single-line diagrams are not supported in V3.

Click the Save button to save the project in PSCAD V3 format.

Editing Project Settings

Right-click on the desired project in the Workspace window and select **Project Settings...**



Chapter 5: Features and Operations

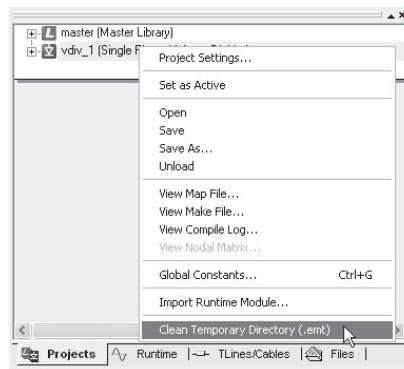
Also, you can right-click over a blank area of any project page in Circuit view and select **Project Settings...** from the pop-up menu.



In either case, a dialog window entitled Project Settings will open. The Project Settings dialog allows for many project-specific parameter settings to be adjusted. For detailed information on this dialog, see 'Project Settings Overview' in Chapter 7.

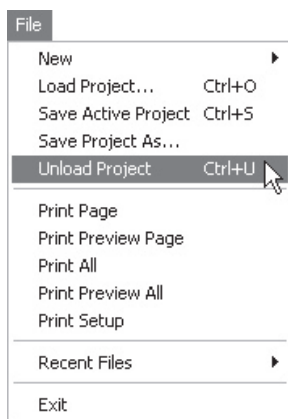
Cleaning The Temporary Directory

To delete all temporary Project files (i.e. compilation, make, data, etc. files), right-click on the Project title in the Workspace window, and select **Clean Temporary Directory (.emt)** from the right-click pop-up menu. See 'PSCAD Temporary Directories' in Chapter 4 for more details.

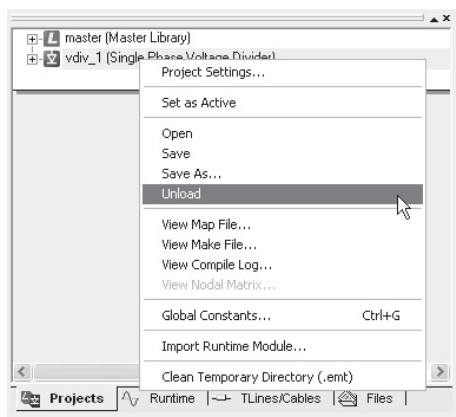


Unloading A Project

In the Workspace window, select the project that is to be unloaded. Click the left mouse button on the menu item **File** in the Main Menu bar - a drop down menu should appear. With your mouse pointer, select **Unload Project**.



You can also unload a project by either right clicking on the project title in the Workspace window and selecting **Unload** from the pop-up menu, or by selecting the project in the Workspace window and pressing **Ctrl + U** on your keyboard.

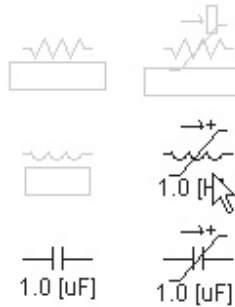


COMPONENTS AND MODULES

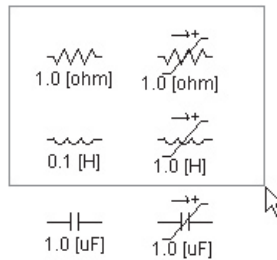
Selecting Objects

Select an individual object with a simple left-click on the object itself. Select a group of objects by one of the following methods:

- Press and hold the **Ctrl** key and then separately select (**left-click**) all objects to be grouped into one selection.



- Press and hold the left mouse button and then drag the mouse pointer so that a box outline appears. Encompass all desired objects to be selected and then release the left mouse button.

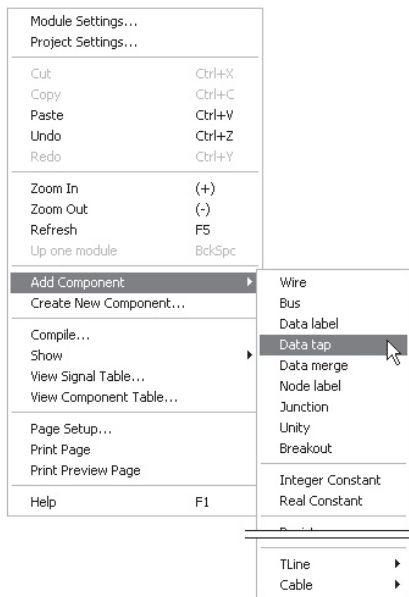


To ungroup any objects, hold the **Ctrl** key and then select (left-click) all objects to be ungrouped.

Adding Components To A Project

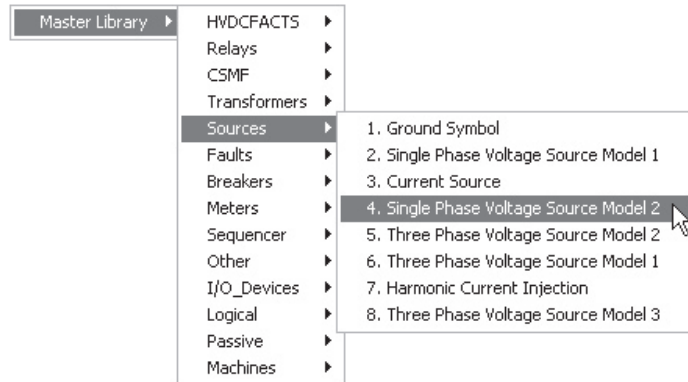
There are a variety of ways to insert components into a Project page. Before proceeding, ensure that your project has been opened and you are viewing the desired Project page in the Circuit window.

- **Manual Copy/Paste:** Open the Master Library and navigate to the area containing the desired component. Right-click on the component and select **Copy**, or select the component and press **Ctrl + C**. Open the project page where you wish to add the component, right-click over a blank area and select **Paste** (or press **Ctrl + V**).
- **Right-Click Menu:** Right-click over a blank area of the page and select **Add Component**. A sub-menu will appear containing the most commonly used components from the Master Library. Select a component and it will be automatically added.



- **Library Pop-Up Menu:** Press **Ctrl + right mouse button** over a blank area of the page to invoke the library pop-up menu system. Select a component and it will be automatically added.

Chapter 5: Features and Operations



- **Control and Electrical Palettes:** Left-click on any of the palette buttons and then drag your mouse pointer over the Circuit window - you should see the object attached to your pointer. Continue to move the object to where you want it placed, and then left-click again.

NOTE: The Control and Electrical Palettes are usually located to the right of the Design Editor window. If you cannot see them, select **View | Control Palette** and/or **View | Electrical Palette** from the Main Menu bar.

Adding Multiple Instances of a Component

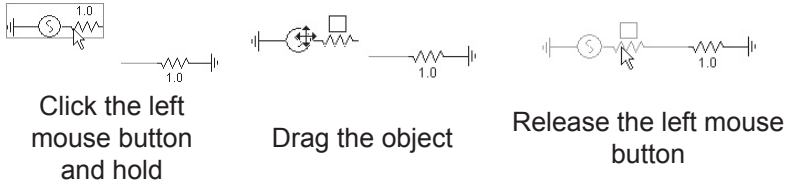
Multiple Instances of the same component can be added when using the Control and Electrical Palettes:

1. Hold down the **Ctrl** key.
2. Left-click on the desired component from either the Control Palette or the Electrical Palette (continue holding **Ctrl**).
3. Move the mouse pointer over the Circuit canvas and left-click again to paste the first Instance of the component (continue holding **Ctrl**).
4. Move the mouse pointer to a new position and left-click again to paste a second Instance of the component.

You may continue this process and add as many Instances as desired, provided the **Ctrl** key remains depressed. To escape from this function, simply release the **Ctrl** key at any time.

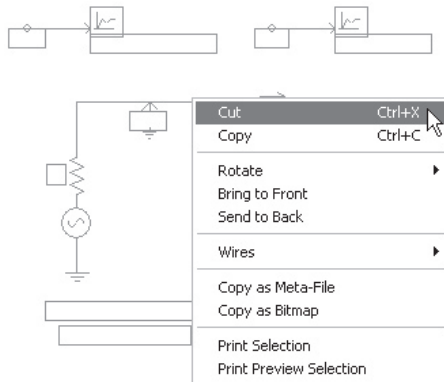
Moving Or Dragging An Object

To move an object, place the mouse pointer over the object icon. Press and hold the left mouse button. Now drag the mouse to move the component. When you move a component, it will always snap to the nearest drawing grid, even if the grid dots are not visible.



Cut/copy/paste

Objects can be cut, copied and pasted as many times as desired within the Circuit window: Right-click on the object and select either **Cut** or **Copy** (or select the object and press **Ctrl + X** or **Ctrl + C** respectively). Multiple objects can be cut, copied or pasted simultaneously.



Once cut or copied, you can paste the object by right clicking over a blank area of the Project page and selecting **Paste** (or press **Ctrl + V**).

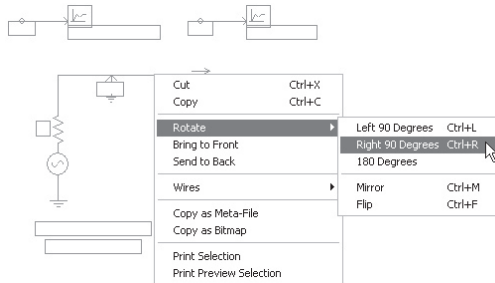
Rotate/Mirror/Flip

Once added, single or multiple objects can be rotated, flipped or mirrored:

- **Hotkeys:** Select the object and press the **R**, **F** or **M** to rotate, flip or mirror respectively.

Chapter 5: Features and Operations

- **Right-Click Menu:** Right-click over the object and select **Rotate** or **Flip**. Select the desired option from the resulting sub-menu.



- **Rotation Bar:** Select the object or group of objects and then press one of the four **Rotation Bar** buttons.

NOTE: If you cannot see the Rotation Bar, select **View | Rotation Bar** from the main menu bar.

Deleting Objects

Select the object (or objects) and press the **Delete** key.

Undo And Redo

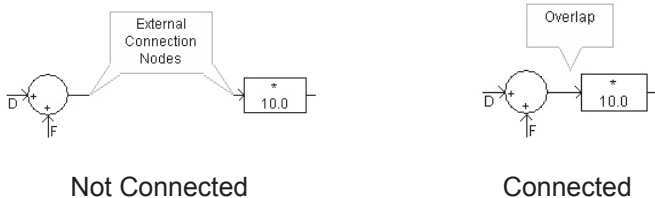
To undo or redo any object manipulations, such as moves, cuts, pastes, deletions, etc., select the **Undo** or **Redo** button in the Main Toolbar, or press **Ctrl + Z** or **Ctrl + Y**. The undo and redo features will store most manipulations and changes, however some limitations will apply.

If this feature does not appear to be working, check the Views section of the Workspace Settings dialog to see if **Undo-Redo** is enabled in the **Circuit View** area.

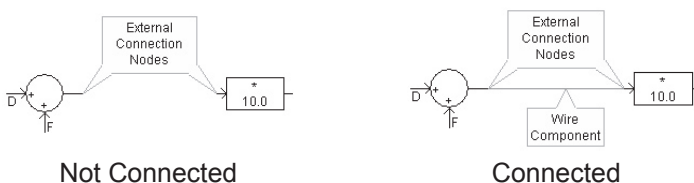
Connecting Components Together

A connection between components can be made in one of the following ways:

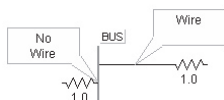
- Overlapping one component external Connection overtop another.



- The endpoint of a Wire component can be used to make contact with an external Connection, or the endpoint another Wire:

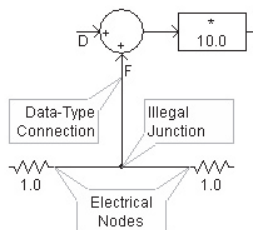


- Any point along a Bus component is considered a valid connection point, to which Wire endpoints and component external Connections may be connected.



Invalid Component Connections

It is important to note the difference between an electrical and a data signal when connecting components together. Wires may be used interchangeably as data signal paths, or for connecting electrical nodes. However, it is **illegal** to connect data signals to/from electrical nodes. For example, the following connection is invalid:



Chapter 5: Features and Operations

Wires are universal in that they assume the signal type of the nodes they are connected to. If a wire is connected to an electrical node, it will become an electrical wire, for example.

Transferring Data Into And Out Of Modules

At some point in building a project, you may want to incorporate a common group of components into a Module (sometimes referred to as a 'sub-page'). Modules can be useful in that they simplify the appearance of the overall circuit design, along with providing a 'container' in which to keep all components involved in a specific task.

There are two types of signal that may be transferred between Modules: These are data and electrical types. Transferring signals between Modules is handled a bit differently depending on the signal type.

Data Signals

When a project is compiled, PSCAD creates a variety of files describing the case so that EMTDC can understand it and run the simulation. One or more of these files are Fortran files (*.f), in which are declared the dynamics (i.e. data signals) in the case. One Fortran file will be created for each Module existing in the case (including the main page – see 'System Dynamics' in Chapter 2 of the EMTDC Manual). Each file is an individual subroutine describing the contents of the corresponding module. Each subroutine is called using a standard Fortran CALL statement from its parent Module Fortran file.

If control signals are passed between Modules, then these signals will either appear as subroutine arguments (if they are hard wired), or be transferred to and from global storage if using Radio Links.

Hard Wiring

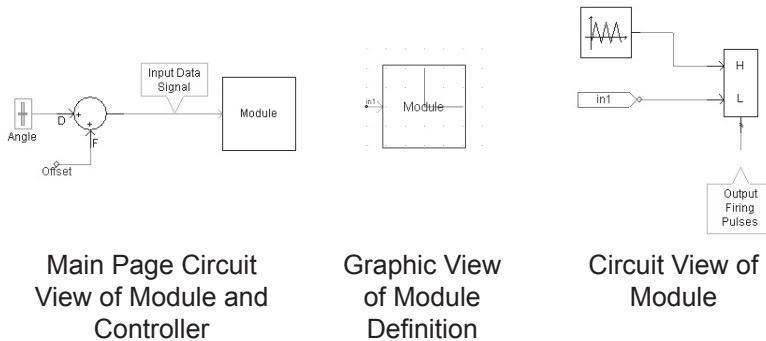
Hard wiring data signals is the original method introduced in PSCAD V3 for transferring data signals between modules. This method involves the following steps:

1. Create a new Module or edit the definition of a Module to which you want to transfer data. See 'Creating a New Component or Module' or 'Editing a Component or Module Definition' in this chapter for more details.
2. Define an input or output Connection in the Graphic section (Graphic tab) of the Module. See 'Connections' in Chapter 9 for more details.

3. Return to Circuit View. See 'Returning to Circuit View' in Chapter 9.
4. Add an Import or Export tag component within the Module Circuit view. Each Import or Export must be named the same as its corresponding Connection.

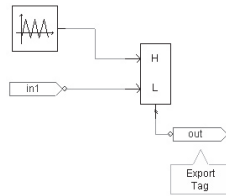
EXAMPLE 5-1:

Consider a case Project that consists of a simple phase angle offset control being input into a Module. The Module definition (i.e. Graphic view) consists of a single external Connection named 'in1,' which is hard wired to the incoming data signal from the control system on the main page. The input signal 'in1' is used as a reference angle to a firing pulse generator inside the Module (i.e. Circuit view).

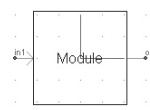


A user wants to bring the firing pulse output signal back out to the main page, and wants hard wire it using an output Connection. The first step is to define an Export tag and connect it to the firing pulse output signal. The user names the export signal 'out'. The final step is to define an output Connection in the Module definition (i.e. Graphic view). See 'Connections' in Chapter 9 for details on adding a new output Connection.

Chapter 5: Features and Operations

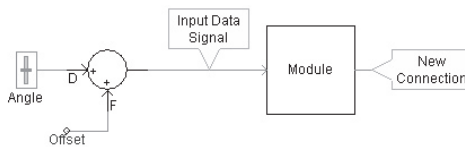


Added Export tag in Circuit View of Module



Defined Output Connection in Graphic View of Module Definition

The user may now input this new output signal to any other component or Wire on the main page.



Main Page Circuit View of Module and Controller with New Connection

Here are a few important factors to remember when transferring control signals into and out of Modules:

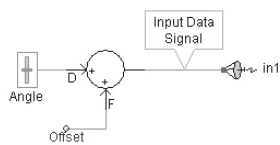
- Import and Export tags cannot be placed on the project main page.
- Only one instance of an Import or Export tag may exist within the Module for a single Connection. If the signal is required at more than one location within the page, you can use the Data Label component.

Wireless Connections

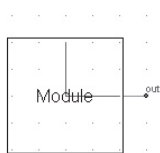
The above control signal transfers can also be accomplished using Radio Link components. Radio Links offer a 'wireless' method of transferring the signals, without the need for cumbersome imports and exports on each Module. In fact, Radio Links allow the user to transfer signals through multiple Module layers.

EXAMPLE 5-2:

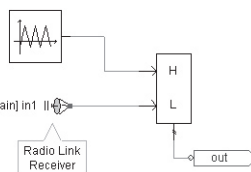
Consider the Project in Example 5-2. A user wants to modify the system so that the input signal 'in1' is transferred into the Module with a wireless connection. The input data signal is first disconnected from the Module and then reconnected to a Radio Link transmitter named 'in1.' Secondly, the input Connection is removed from the Module definition (i.e. Graphic view), as it is no longer required. Lastly, the Import tag is removed and replaced with a Radio Link receiver.



Main Page Circuit View of Controller with Radio Link Transmitter



Graphic View of Module Definition with Input Connection Removed



Circuit View of Module with New Radio Link Receiver

Electrical Nodes

When a project is compiled, PSCAD creates a variety of files describing the case so that EMTDC can understand it and run the simulation. One or more of these files are Data Files (*.dta), which describe the construction of electrical network in the case. One Data file is created for each Module existing in the case (including the main page – see 'Network Solution' in Chapter 2 of the EMTDC Manual).

If an electrical node needs to be represented within a module, then PSCAD must be informed of this requirement so that the network of electric nodes can be properly mapped. The manner in which this is accomplished is as follows:

1. Create a new Module or edit the definition of a Module to which you want to transfer data. See 'Creating a New

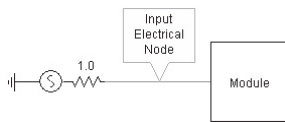
Chapter 5: Features and Operations

Component or Module' or 'Editing a Component or Module Definition' in this chapter for more details.

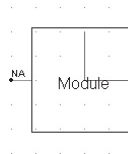
2. Define an electrical Connection in the Graphic section (Graphic tab) of the Module. See 'Connections' in Chapter 9 for more details.
3. Return to Circuit View. See 'Returning to Circuit View' in Chapter 9.
4. Add an External Electrical Node (XNode) component within the Module Circuit view. Each XNode must be named the same as its corresponding Connection.

EXAMPLE 5-3:

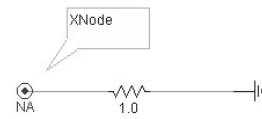
Consider a simple case, which consists of a single-phase source line being input into a Module. The Module definition (i.e. Graphic view) includes a single external Connection named 'NA,' which is hard wired to the incoming line from the source on the main page. The Module canvas (i.e. Circuit view) consists simply of a Resistor component connected directly to a Ground component.



Main Page Circuit View
of Module and Source



Graphic
View of
Module
Definition



Circuit View of Module

Here are a few important factors to remember when transferring electrical signals into and out of Modules:

- XNodes cannot be placed on the project main page.
- Only one instance of an XNode may exist within the Module for a single Connection.
- An XNode cannot be directly grounded. That is, do not connect a Ground component directly to an XNode. A ground may be connected directly to the corresponding electrical Connection however.

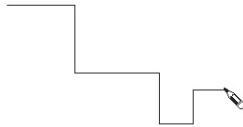
PSCAD

Wire Mode

PSCAD includes a special wire drawing feature called Wire Mode, which allows the user to quickly draw interconnecting Wires between components.

To invoke Wire Mode, click the **Wire Mode** button in the Main Toolbar or press **Ctrl + W** on your keyboard. With the Project open in Circuit view, move the mouse pointer onto the Project page. The mouse pointer will have turned into a pencil, which indicates you are in Wire Mode.

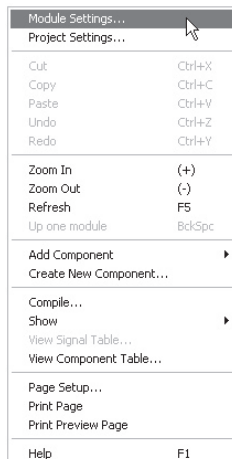
To draw a wire, move the cursor to the node where you want the line to start and left-click. Move the cursor to where you want the line to end and left-click again to complete the wire. Multi-segment Wires may be built by continuing to left click at different points.



To turn-off Wire Mode, either press the **Esc** key, press **Ctrl + W**, or click the **Wire Mode** button again.

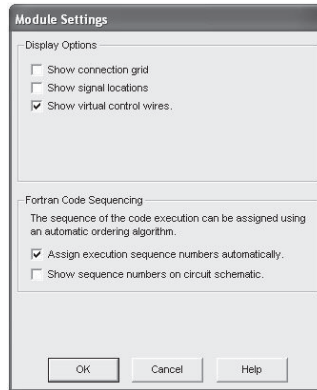
Editing Module Settings

Right-click on a blank part of the Circuit page and select **Module Settings...** These settings are module specific and they will vary for each individual module in the project.



Chapter 5: Features and Operations

A dialog window entitled 'Module Settings' should open, as shown below:



The Module Settings dialog inputs are described in the following sections.

Display Options

These options are related to the page display.

Show Connection Grid

Select this option to view the major grid points on the module page.

Show Signal Locations

When this option is enabled, PSCAD will use icons placed on data signal wires and connections so as to allow for easy, graphical differentiation between feed-forward and feed back signals. For more on this, see 'Component Ordering' in Chapter 11.

Show Virtual Control Wires

Select this option to 'virtually' connect signals on a page that are connected by Data Labels. For more on this, see 'Virtual Control Wires' in Chapter 11.

Fortran Code Sequencing

These options are related to the way in which System Dynamics code is ordered. For more on this option, see 'Component Ordering' in Chapter 11.

Assign Execution Sequence Numbers Automatically

Select this option to ensure that a PSCAD smart algorithm will automatically order your control components. This algorithm systemati-

PSCAD

cally scans all control systems and sub-modules with the Module and determines the sequence that each component should appear in the in EMTDC System Dynamics.

This option is by default enabled, and should remain that way unless the user wants to order these components manually. For more on this option, see 'Component Ordering' in Chapter 11.

Show Sequence Numbers on Circuit Schematic

When this option is enabled, PSCAD will label each Component and Module in the Project with an auto-generated sequence number. This sequence number represents the sequential placement of the Component or Module code in the Project Fortran files. For more on this option, see 'Component Ordering' in Chapter 11.

Editing The Page Setup

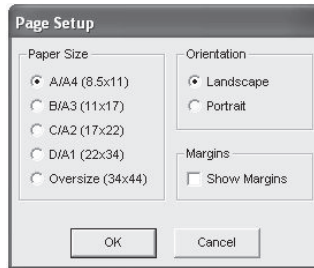
The Page Setup dialog contains features related specifically to a particular module canvas. That is, these features can vary for each individual Module in the project.

Right-click on a blank part of the canvas in Circuit view and select **Page Setup...**



Chapter 5: Features and Operations

A dialog window entitled 'Page Setup' should open, as shown below:



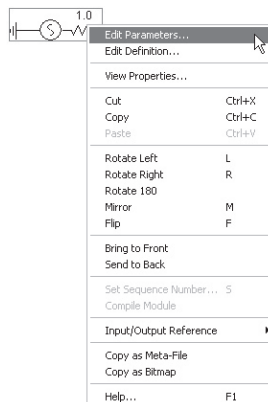
The Page Setup dialog inputs are described below:

- **Paper Size:** Select a standard paper size for this Module.
- **Orientation:** Select the Module page orientation.
- **Show Margins:** Select this option to show borders of one paper size smaller than the current size. This is especially convenient when attempting to reduce the page size of a canvas containing components and other objects.

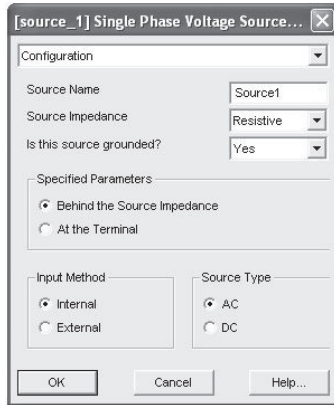
NOTE: The **Oversize (34x44)** paper is included for compatibility with the PSCAD V2 large canvas size. However, it can still be used as a valid paper size.

Editing Component Parameters

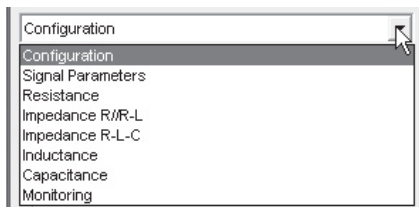
To edit the parameters of a component, either left double-click the component, or right-click on the component and select **Edit Parameters...** from the pop-up menu.



This will bring up the properties dialog for that particular component Instance. The figure below shows the first page of the Parameters dialog for one of the single-phase source models in the PSCAD Master Library.



Dialogs for other components will differ, but all include the same basic features and most contain multiple pages. At the top of the dialog is a drop list, which contains a list of all the Parameter dialog pages. In this example, the first page is entitled 'Configuration.' If only one page exists, then the drop list will be disabled. To move through or view any other existing pages, left-click the down arrow on the field as shown below:



If there are too many items in the list, you will need to scroll through the list to see all the form names. To scroll through the list, hold the left button pressed on the list and drag it up and down. Or press the up or down arrow on your keyboard when the list is open. If there are more items in the list that it can display, you will see the list scrolling.

Chapter 5: Features and Operations

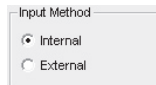
Changing Parameter Values

Each page will usually contain an assortment of Input Fields and Choice Boxes. Depending on the Input Field type, you can make any changes you desire by either selecting one of the choices given, or by adjusting data directly.

To change the parameter value in a Text Field, simply click on the field itself (white background) and type the new value. Some property fields will contain a unit (i.e. [MVA], [sec], [m/sec], etc.), displayed next to the property value, to indicate the unit used. See Unit System for more details.

Drop-list type Choice Boxes will have a downward pointing arrow on the right hand side. Click on that arrow to see the list and then click on the required item. The Choice Box could have more items than visible in the list. To scroll through the list, simply use the up or down arrows on your keyboard when the list is open.

To select a radio button type Choice Box, simply click on the radio button item.

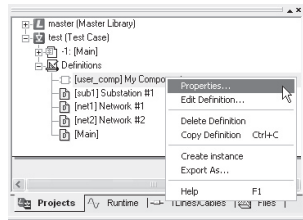


Once you are finished with all of your editing, click on the **OK** button to accept the edited values and to exit the dialog. Clicking on the **Cancel** button will exit the form and ignore all the changes made.

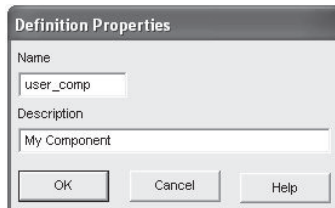
Editing Definition Properties

Editing definition properties, as opposed to Editing a Component or Module Definition, refers to two special properties of the definition that must be accessed separately: The definition Name and Description.

In the Workspace window, access the Definitions Branch and right-click on the component definition listed therein, and select **Properties....**:



This brings up the Definition Properties dialog window for editing the definition Name and Description:



The following list describes the functions of the Definition Properties dialog:

- **Name:** Enter a name for the definition. This name must conform to Fortran standards (i.e. it cannot begin with a number or contain any spaces).
- **Description:** Enter a description of the definition. The description may include spaces and/or other non-standard characters.

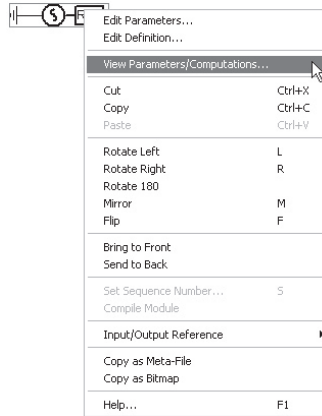
Viewing/Saving Component Parameters

Component input Parameters and Computations section data can be viewed and/or saved to text file from within an easy to navigate environment. All component Parameter attributes are listed in spreadsheet format, including Symbol, Caption, Type, etc. A Computations tab is also provided, which lists the values of computed quantities defined in the component Computations section.

Viewing Component Parameters

To view the calculated results from the Computations section of a particular component Instance or to view the current state of the component input Parameters, right-click on the component and select **View Parameters/Computations...** from the pop-up menu.

Chapter 5: Features and Operations



Column Descriptions

The parameter and computation data is organized into columns. Each column is described below:

Parameters		Computations						
Name	Caption	Type	Unit	Minimum	Maximum	Data	Value	
Ctrl	Source Control:	CHOICE				0	0	
Es	Voltage Magnitude (L-L, RMS)	REAL	kV	0	1e+038	13.8 [kV]	13.8	
Exph	External Phase Input Unit	CHOICE				0	0	
F	Base Frequency	REAL	Hz	0.001	1e+038	60.0 [Hz]	60.0	
F0	Frequency	REAL	Hz	0	1e+038	60.0 [Hz]	60.0	
Ia	Name for Phase A Current(IA)	TEXT						

Parameters Tab

Parameters:

- **Name:** The **Symbol** name of the corresponding Parameter Field.
- **Caption:** The **Caption** of the corresponding Parameter Field.
- **Type:** The type of Parameter Field (Input, Text, Choice Box).
- **Unit:** The **Unit** name of the corresponding Parameter Field. This is the default or Target Unit for the Field.
- **Minimum:** The **Minimum** of the corresponding Parameter Field.
- **Maximum:** The **Maximum** of the corresponding Parameter Field.
- **Data:** The actual data as it appears in the Component Parameters dialog.
- **Value:** The actual value of the parameter following manipulation by the Unit System. This is the actual value in terms of the Target Unit.

Name	Expression	Type	Value
D	$(-H-Ff) / (2 * E)$	REAL	0.0418109792926
D'	$(D > 0 ? D : (H-Ff) / (2 * E))$	REAL	0.0418109792926
E	$Sqrt(S * RNI) * (1 - RNI)$	REAL	-1.40342990777
Ff	$S * Tan(phiIR) * (1 - RNI * RNI)$	REAL	-8.37768345822
G	$Sqrt(S) * S * (RNI - 1)$	REAL	0.345527163495
H	$Sqrt(Ff * Ff + 4 * E * G)$	REAL	8.15037540773

Computations Tab

Computations:

- **Name:** The **Symbol** name of the corresponding Parameter Field.
- **Expression:** The actual PSCAD Script expression in the Component Definition Computations Segment, which defines the value of the constant.
- **Type:** The declared type of the computed constant (i.e. REAL, INTEGER, LOGICAL).
- **Value:** The actual value of the computed constant following pre-compilation.

Formatting Viewed Parameters

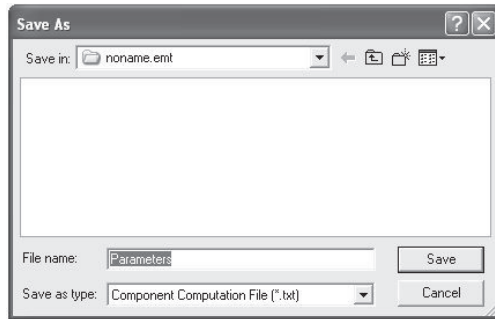
Once the Parameters/Computations window has been opened, the data is open to some simple formatting. Although parameter name and value cannot be altered from this window, users may adjust the order in which data appears; this can be useful if the data is to be saved to a file.

Left-click and hold a listed parameter name in the display window. **Drag** the mouse pointer to where you want to place the parameter and **release** the mouse button.



Saving Component Parameters to a File

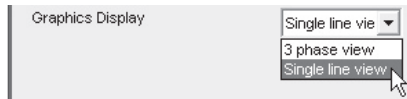
Component Data can be saved to a formatted text file at any time. Select either the **Parameters** or **Computations** tabs depending on what data is to be written. Press the **Save As...** button to bring up the Save As dialog window.



Enter a name for the file and press the **Save** button. Once saved, PSCAD will open the file automatically for viewing.

Changing To Single-Line View

If the component being edited is a three-phase electrical component, then it is likely there will be a Drop List entitled **Graphics Display** on the main properties page.



To toggle the component between single-line and 3-Phase view, open the component properties dialog and select the desired view from this field.

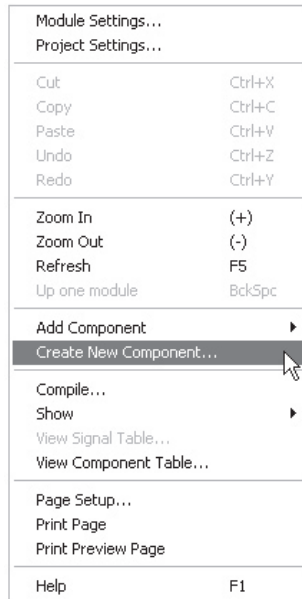
Creating A New Component Or Module

New Components or Modules can be created in PSCAD by using the Component Wizard. The Component Wizard is a graphical tool that creates a definition 'shell,' which the user may then use as a basis for further component or module design.

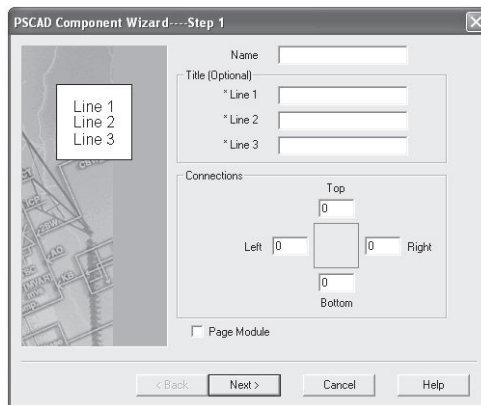
The following topics describe the simple steps involved when using the Component Wizard. For more on component design, see 'Designing Components' in Chapter 9.

Using the Component Wizard

Before proceeding, ensure that the Project you are working with is opened in Circuit view. Press the **New Component** button in the Main Toolbar, or move the mouse pointer over a blank area of any page, right-click and select **Create New Component...**



The first page of the Component Wizard should then appear as shown below:



Step 1

Step 1 in creating a new component involves entering the following data:

- **Name:** Enter a name for the Component Definition. This name must conform to Fortran standards (i.e. it cannot begin with a number or contain any spaces).

Chapter 5: Features and Operations

- **Title (Optional):** If text is detected in any of the three fields provided in this section, a Text Label containing the text will be added to the component Graphic.
- **Connections:** If you would like to automatically create any Connections to your component, enter the number required in either the **Left**, **Top**, **Right** or **Bottom** input fields. For example, entering 2 in the Left field indicates that you want to auto create 2 Connections on the left side of the component Graphic.
- **Page Module:** Select this input if you want this new component to be a Module (also referred to as a page component or a sub-page).

NOTE: If a number is added to any of the **Connections** fields, then you must define each Connection in Step 2.

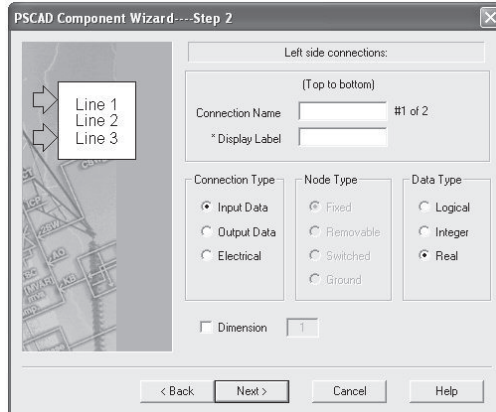
Enter all the parameters that you require according to the above descriptions. When finished, click the **Next >** button. If no Connections were specified above, skip to Step 4.

Step 2

Step 2 in creating a new component involves defining data for any Connections specified in Step 1.

- **Connection Name:** Enter a name for the Connection. This name must conform to Fortran standards (i.e. it cannot begin with a number or contain any spaces).
- **Display Label:** If text is detected in this field, a Text Label containing this text will be added near the Connection in the component Graphic, acting as a Connection label.
- **Connection Type:** Select the type of Connection here. Note that for Input Data type, the Component Wizard will automatically draw an input arrow graphic for the Connection.
- **Node Type:** Select the type of electrical Connection here. This input is only enabled if the Connection Type is selected as Electrical.
- **Data Type:** Select the type of data Connection here. This input is only enabled if the Connection Type is selected as either Input Data or Output Data.

- **Dimension:** Select the check box and enter the dimension of the Connection signal. This input is only used if the signal is not a scalar (i.e. it is an array or vector).



Enter all the parameters that you require according to the above descriptions. This page will be repeated for each Connection specified in Step 1, starting with the left side of the component Graphic. The text box at the top of the dialog will indicate what Connections are currently being defined.

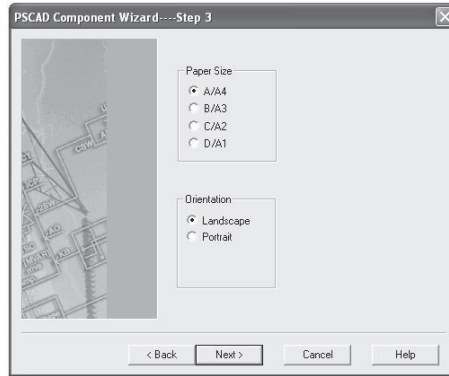
When finished each page, click the **Next >** button. If **Page Module** was not selected in Step 1, skip to Step 4. If a mistake is made, press the **< Back** button at any time.

Step 3

Step 3 in creating a new component involves defining the Page Setup if the **Page Module** input was selected in Step 1.

- **Paper Size:** Select a standard paper size from the list provided.
- **Orientation:** Select either Portrait or Landscape page orientation.

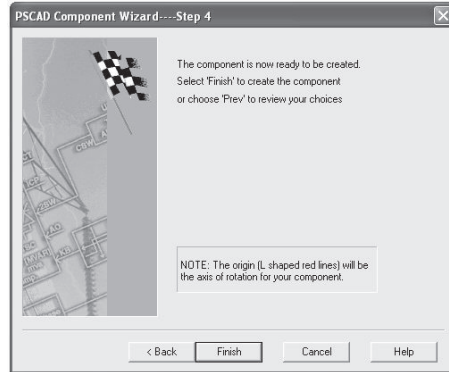
Chapter 5: Features and Operations



When finished, click the **Next >** button. If a mistake is made, press the **< Back** button at any time.

Step 4

Step 4 is simply a confirmation step. Press the **Finish** button if you are satisfied that everything is correct. Press the **< Back** button to go back to the previous step.



Creating a Default Module

You can forego the above procedure when creating a simple Module. Simply press the **Create Default Module** button in the PSCAD Main Toolbar.

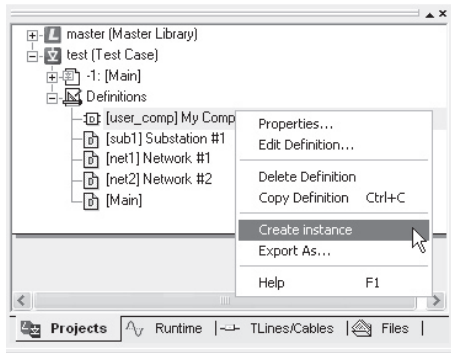


PSCAD

This creates a simple Module with no external connections to its parent page. To create a Module with Connections, follow the procedure outlined above.

Creating The First Instance of a Definition

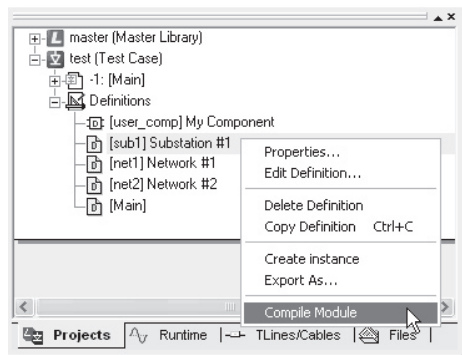
In the Workspace window, navigate to the Definitions branch within the project tree. Either select the desired component Definition from the list, right-click and select **Create Instance** as shown below, or drag and drop the definition directly on the canvas. See 'Drag and Drop' in this chapter for more details.



NOTE: Component definitions should be moved to a Library Project once tested.

Compiling an Individual Module

In the Workspace window, navigate to the Definitions Branch within the project tree. Select the desired Module from the list, right-click and select **Compile Module**.



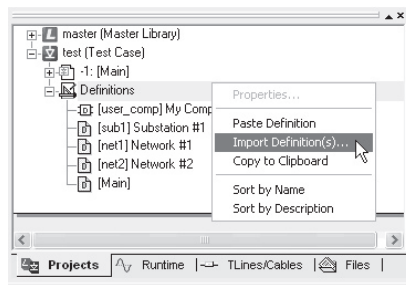
Chapter 5: Features and Operations

Importing/Exporting Definitions

Component Definitions may be imported or exported to and from different Projects, or between users, by first saving the Definition as a special Component Definition (*.cmp) file. A Component Definition file is simply a text file, which contains the text script defining the component as it would appear in a Case or Library Project (*.psc or *.psl) file. Importing and exporting a definition by creating a Component Definition file is similar to cutting and pasting a chunk of script directly from/to a Project file.

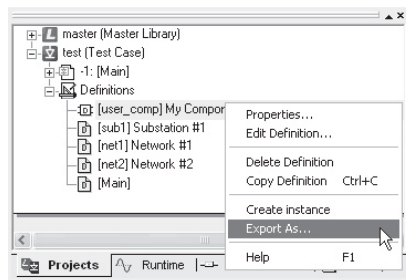
Import a Definition

In the Project Section of the Workspace, left-click on the [+] box beside the desired Project name to expand the Project Tree. Right-click on the Definitions Branch and select **Import Definition(s)...** See 'Definitions Branch' in Chapter 4 for more details.



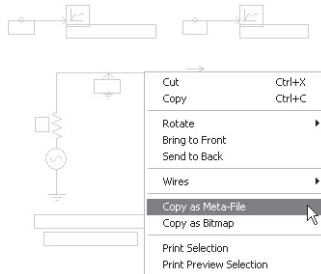
Export a Definition

To export a Definition (or save the Definition to a special Component Definition (*.cmp) file), left-click on the [+] box beside the Project name where the Definition is stored to expand the Project Tree. Expand the Definitions Branch in the same manner. Right-click on the desired Definition from the list and select **Export As...** See 'Definitions Branch' in Chapter 4 for more details.



Copy As Meta-file Or Bitmap

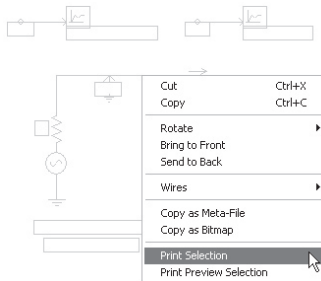
Bring up the desired area in Circuit view. Select the objects to be printed (see Selecting Objects in this chapter), right-click and select **Copy as Meta-File** or **Copy as Bitmap**.



Then, simply paste the copied selection into a report program of your choice.

Printing Selected Objects

Bring up the desired area in Circuit view. Select the objects to be printed (see Selecting Objects in this chapter), right-click and select **Print Selection**. A print dialog window should appear - select your desired printer properties and click **OK**.



To preview the selection before printing, select the objects to be printed, right-click and select **Print Preview Selection**.

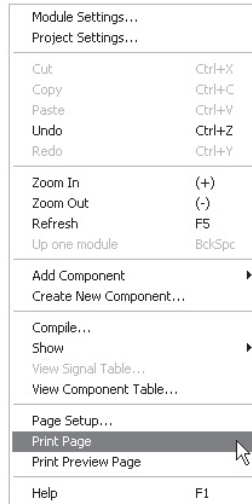
NOTE: More than one object must be selected to print a selection.

Printing a Module Page

Bring up the desired module in Circuit view. Right-click on a blank portion of the page and select **Print Page**, or press the **Print** button

Chapter 5: Features and Operations

in the Main Toolbar. A print dialog window should appear - select your desired printer properties and click **OK**.



To preview before printing, bring up the desired module in Circuit view. Right-click on a blank portion of the page and select **Print Preview Page**.

TUTORIAL: CREATING A NEW PROJECT

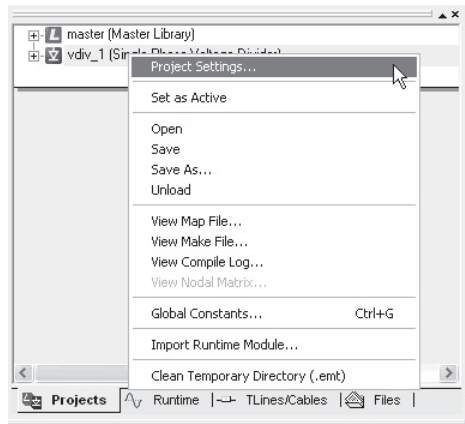
Follow the steps outlined in the following sections to create a new project.

Adding a New Case Project

Press the **New** button in the PSCAD Main Toolbar.

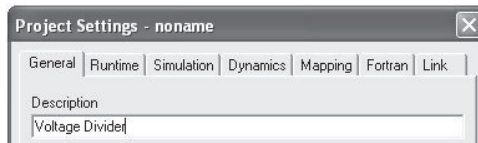


A new Case Project should appear in the Workspace window entitled 'noname.' Right-click on the Project name and select **Project Settings....**

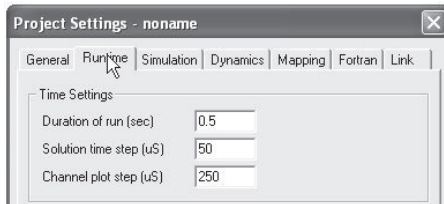


NOTE: If the Workspace is not visible, select **View | Workspace** in the Main Menu bar.

This action will open the Project Settings dialog. Click the General tab at the top of the dialog, left-click inside the **Description** field and type a description for the case, say, 'Voltage Divider.'

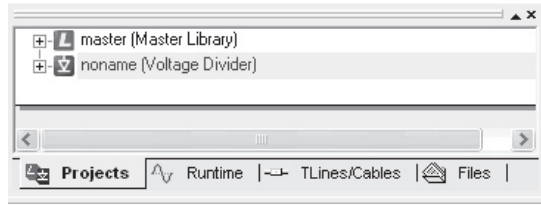


Click the **Runtime** tab and get familiar with the inputs displayed therein (see 'Runtime' in Chapter 7 for details on parameters).



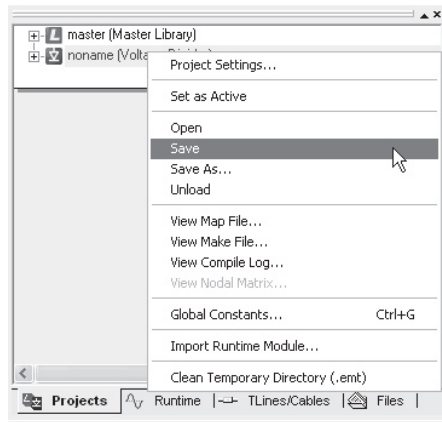
The Project will be automatically set up with a run duration of 0.5 seconds with a 50 μ s time step by default (these settings are sufficient for now). Click the **OK** button. The description you entered should now appear in the Workspace beside the Project filename.

Chapter 5: Features and Operations

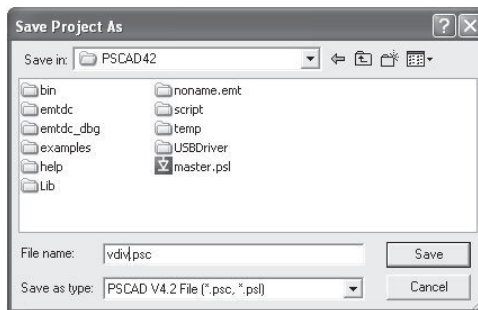


Saving the Project

To save your newly created Case Project, right-click on the Project name in the Workspace and select **Save** from the pop-up menu.



This will open the Save Project As dialog.



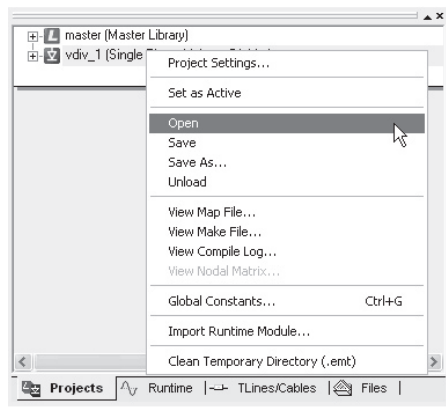
Use the buttons at the top of this dialog to navigate to the desired directory in which you want to save the Project. Type the name of the file (say 'vdiv.psc') in the **File name** field near the bottom of the dialog. Make sure that you include the extension '*.psc.' Click

the **Save** button to save the changes. The Project name in the Workspace will turn to black and the new filename should appear.

NOTE: Although we strive to avoid spurious crashes, you should save your case project frequently in order to avoid loss of work.

Opening the Project Main Page

To open the main page of your new Case Project, left double-click on the Project name in the Workspace, or right-click on the name and select **Open**.



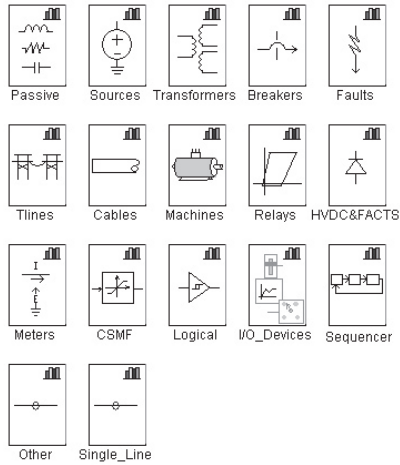
In either instance, an empty page should be opened in the Circuit window of the Design Editor. This is the main page of your new Project, on which you will draw the circuit.

Opening the Master Library

The first project listed in the Workspace window is always the Master Library (master.psl). It contains most of the components you will ever need to build any circuit. All of the components that will be used to create this new Case Project example are available here.

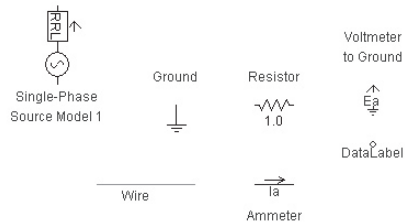
To open the Master Library, follow the same procedure as outlined in 'Opening the Project Main Page' in this chapter. The picture shown below is the top left corner of the Master Library. Each icon is a module that opens into another page window, giving access to components belonging to that group.

Chapter 5: Features and Operations



Assembling a Voltage Divider Circuit

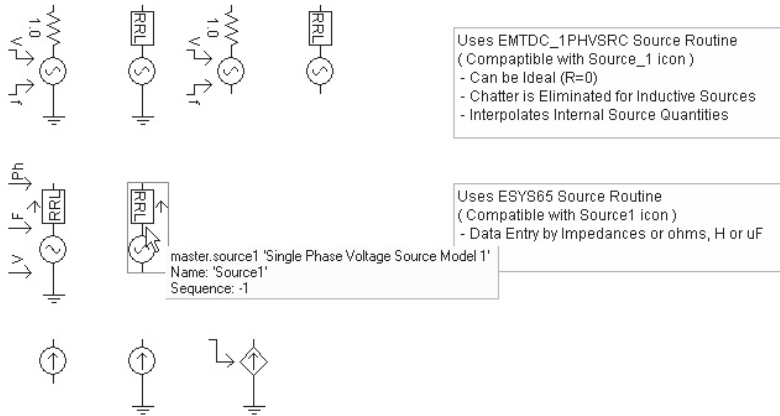
The voltage divider circuit in this example will use eight different components as shown below.



NOTE: Initially, it may be a challenge to navigate through the Master Library and find these components one by one.

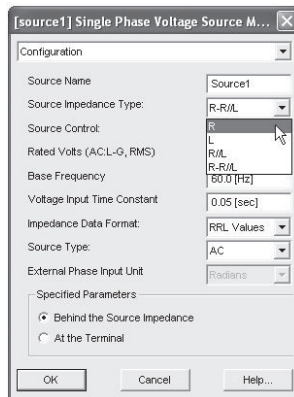
Locating the Single-Phase Source

The first step is to locate the single-phase source model to be used in your circuit (in Master Library 'Sources' Module). There are three different source models available in the Master Library - we will be using the Single Phase Voltage Source Model 1 component.



Source 1 Model Within Sources Module in Master Library

Once you find it, add it to your new Case Project main page using one of the techniques outlined in 'Adding Components to a Project' in this chapter. Move the source component to an appropriate place on the page. Left double-click on the component to bring up the component Parameters dialog (see 'Editing Component Parameters' in this chapter). On the **Configuration** page, change the **Source Impedance Type** Drop List to 'R' (for purely resistive).



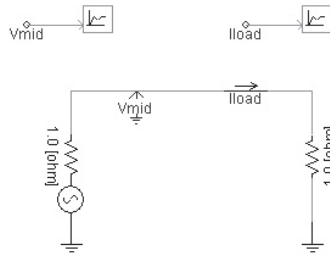
Also on the **Configuration** page, change the **Rated Volts (AC:L-G, RMS)** input field to from '110.00 [kV]' to '70.71 [kV]'. This will give an internal source voltage of 100 kV peak.

Chapter 5: Features and Operations

Save the Project.

Add and Assemble

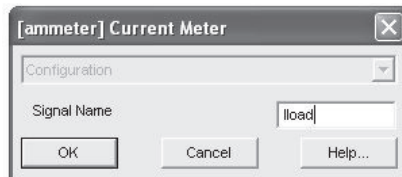
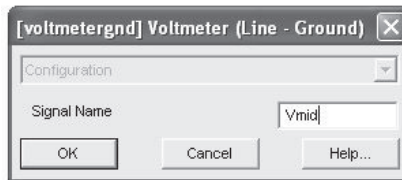
The next step is to add the remaining components (i.e. Wire, Resistor, Current Meter, Data Label, Output Channel, and Ground components). Arrange all components to form the simple voltage divider shown in the following diagram:

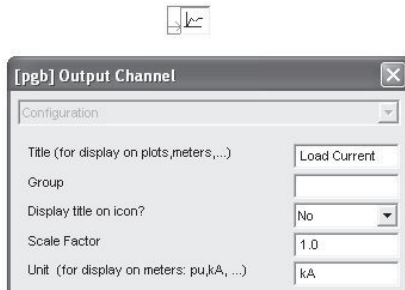
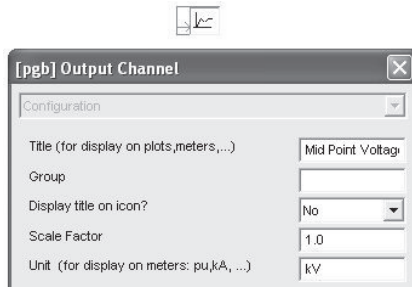


Save the Project. See Adding Components to a Project, Selecting Objects and Moving or Dragging an Object for more details.

Editing the Remaining Component Parameters

Use the properties shown below for the remaining components (see 'Editing Component Parameters' in this chapter). Only the properties to be changed from their default values are mentioned. Use the resistor with its default properties.





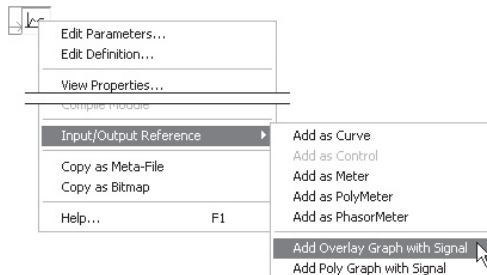
Save the Project.

Plotting Results

In order to view any results from our voltage divider circuit, we must add a Graph Frame and set it up to display the waveforms. The following sections describe only those aspects of plotting needed in this exercise. See 'Preparing Data for Control or Display' in Chapter 6 for further reading.

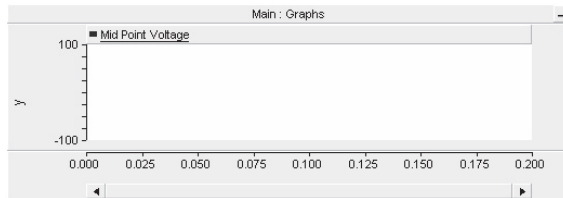
Adding a Graph Frame

Right-click on the Output Channel component called 'Mid Point Voltage' to bring up the pop-up menu. Select **Input/Output Reference | Add Overlay Graph with Signal**.

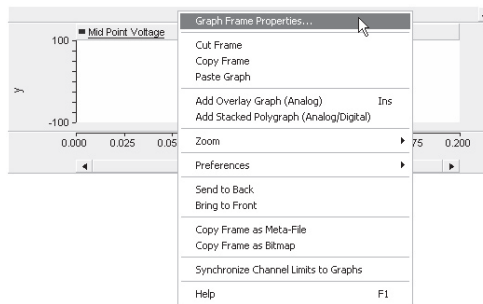


Chapter 5: Features and Operations

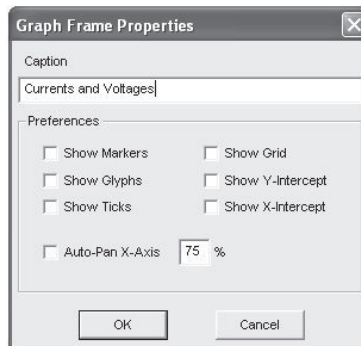
This should create a new Graph Frame, Overlay Graph and a Curve simultaneously as shown below:



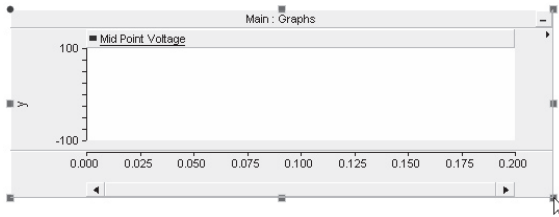
Right-click on the Graph Frame title bar (top bar on the plot labelled 'Main : Graphs') and select **Graph Frame Properties...** from the pop-up menu.



This should bring up the Graph Frame Properties dialog window. In the **Caption** field, change the title to 'Currents and Voltages.' See 'Graph Frames' in Chapter 6 for more details on the options in this dialog.

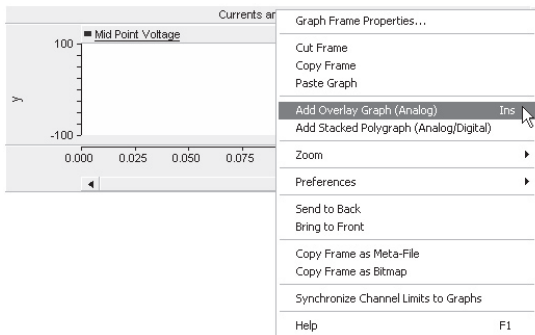


To resize the Graph Frame at any time, left click on the Graph Frame title bar so that Grips appear. Left-click and hold on a corner Grip and drag mouse outwards. Resize the Graph Frame to approximately 5x5 inches.

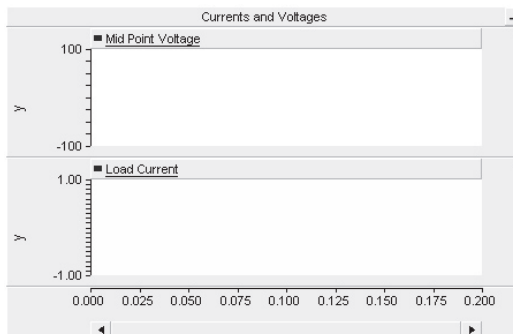


Adding Another Overlay Graph and Curve

Right-click on the Graph Frame title bar and select **Add Overlay Graph (Analog)** (or press the **Insert** key while the mouse pointer is over the Graph Frame).

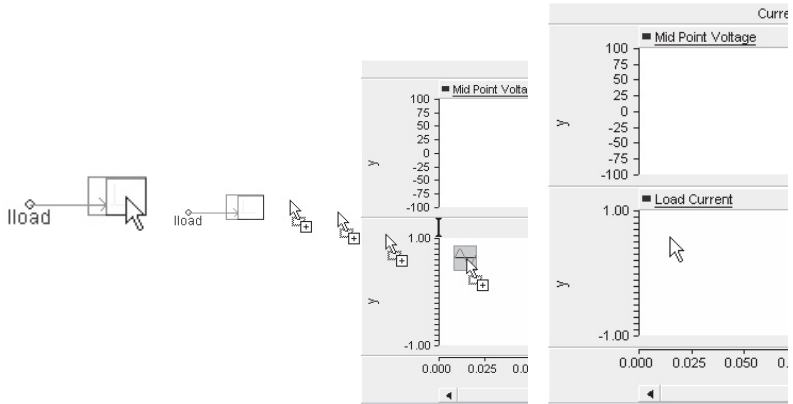


A new Overlay Graph should appear within the Graph Frame directly below the existing graph (see 'Graphs' in Chapter 6 for more).



Chapter 5: Features and Operations

Add a Curve to this graph to monitor load current: Hold down the **Ctrl** key and click and hold the **left mouse button** while over the Output Channel with caption 'Load Current'. Drag the mouse pointer over the new Graph and release the mouse button. A Curve should appear in the new Graph.



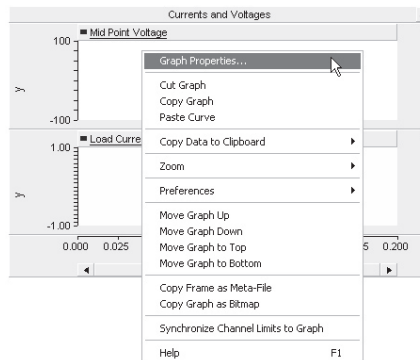
Select the
Output
Channel
(**Ctrl + left
mouse
hold**)

Drag the Mouse Pointer
Over the New Graph

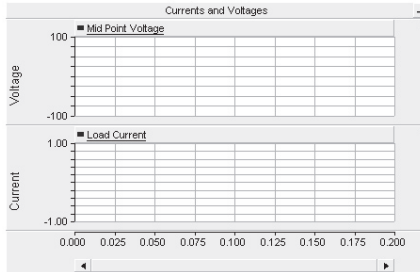
Release the Mouse
Button

Editing the Graph Properties

To customize an individual Graph title and/or vertical axis label, right click over top the Graph and select **Graph Properties...** Edit the Graph Properties as you see fit (see 'Graphs' in Chapter 6 for more).



For example, change the **Y-Axis Title** input field to show 'Voltage' on the voltage graph, and 'Current' on the load current graph. You may also want to turn on/off Grid Lines and adjust the scaling. The Graph Frame should appear similar to that shown below once completed.

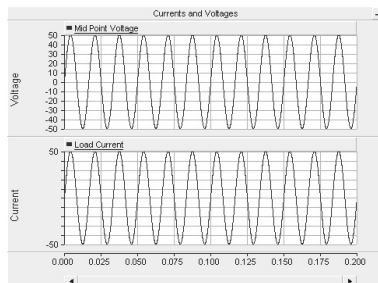


Save the Project.

Running the Project

Before running your new Project, ensure that it is the active project in the Workspace window (see 'Setting the Active Project' in this chapter). Press the **Run** button in the Main Toolbar to run the simulation (see 'Running a Simulation' in this chapter).

This is the last step, assuming there are no errors. If there are any, they will be logged in the Output Window. See 'Common Output Window Messages' in Chapter 11 for more details. Your simulated results should look similar to the following, once the simulation has completed.





NOTE: It may be necessary to readjust your X and Y-axes, as well as zoom and time frame. See 'Online Plotting and Control' in Chapter 6 for more details.

DRAG AND DROP

The addition of drag and drop functionality to PSCAD V4, greatly improves the efficiency of designing Projects - especially the construction of online plots and controls. Drag and drop may be performed directly on the drawing canvas in Circuit view, or items may be dragged and dropped from the Workspace onto the Circuit canvas.

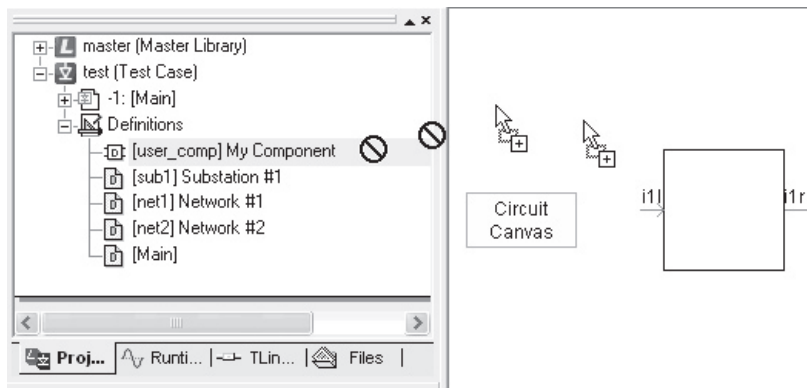
The drag and drop feature uses mouse pointer icons to indicate whether or not a dragged object may be placed under the current mouse position:

-  Drop position is valid
-  Drop position is invalid

Create a Component Instance

Component Instances can be created directly from the corresponding Definition in the Projects Section of the Workspace.

1. Navigate to the Definitions Branch in the Projects Section of the Workspace.
2. **Select and hold** the desired component Definition with your left mouse button.
3. Drag the mouse pointer to a blank area on the Circuit canvas and release the mouse button to drop.



Drag and Drop onto Circuit Canvas from Project Section of Workspace

Copy a Component Instance

All component Instances appearing on the Circuit canvas may be copied and pasted using drag and drop.

1. Hold down the **Ctrl** key.
2. Move the mouse pointer over the component Instance to be copied.
3. Select and hold with your **left mouse button**.
4. Drag the mouse pointer to a blank area of the Circuit canvas and release the mouse button to paste.



Copying a Component Instance

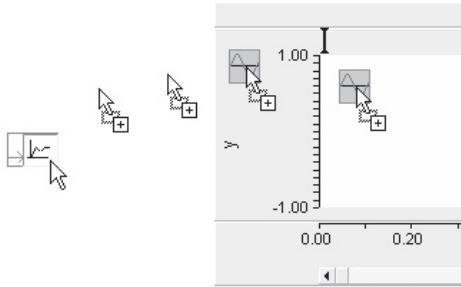
Adding a Curve to a Graph

A Curve can be added directly to a Graph in one of two ways:

- From an Output Channel Instance
- From a Runtime Object in the Workspace.

From an Output Channel

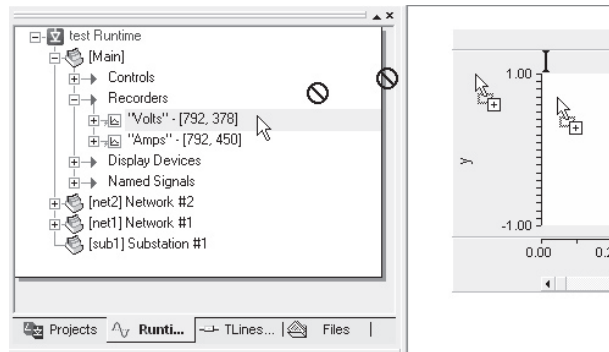
1. Hold down the **Ctrl** key (use the **Shift** key if Enable 'Shift' key to create controls and curves is enabled in the Workspace Settings dialog).
2. Move the mouse pointer over the desired Output Channel Instance.
3. Select and hold with your **left mouse button**.
4. Drag the mouse pointer to the desired Graph and release the mouse button to paste.



Adding a Curve to a Graph from an Output Channel

From a Runtime Object

1. With the current Project set as active, click the Runtime tab in the Workspace.
2. Navigate to the corresponding Runtime Object for the desired Output Channel (will be in the Recorders sub-branch).
3. Select the object and hold with your **left mouse button**.
4. Drag the mouse pointer to the desired Graph and release the mouse button to paste.



Adding a Curve to a Graph from a Runtime Object

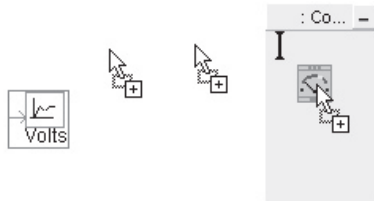
Adding a Meter to a Control Panel

A Meter can be added directly to a Control Panel in one of two ways:

- From an Output Channel Instance
- From a Runtime Object in the Workspace.

From an Output Channel

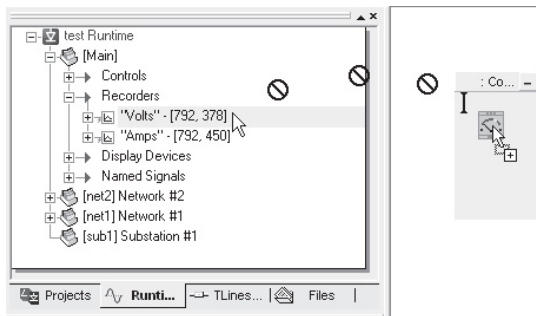
1. Hold down the **Ctrl** key (use the **Shift** key if Enable 'Shift' key to create controls and curves is enabled in the Workspace Settings dialog).
2. Move the mouse pointer over the desired Output Channel Instance.
3. Select and hold with your **left mouse button**.
4. Drag the mouse pointer over the desired Control Panel title bar and release the mouse button to paste.



Adding a Meter to a Control Panel from an Output Channel

From a Runtime Object

1. With the current Project set as active, click the Runtime tab in the Workspace.
2. Navigate to the corresponding Runtime Object for the desired Output Channel (will be in the Recorders sub-branch).
3. Select the object and hold with your **left mouse button**.
4. Drag the mouse pointer over the desired Control Panel title bar and release the mouse button to paste.



Adding a Meter to a Control Panel from a Runtime Object

Chapter 5: Features and Operations

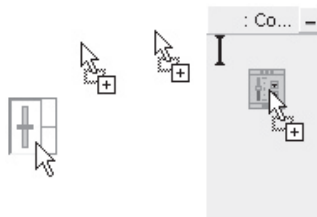
Adding a Control Interface to a Control Panel

A Control Interface can be added directly to a Control Panel in one of two ways:

- From a controls component Instance (i.e. Slider, Dial, Switch or Push Button)
- From a Runtime Object in the Workspace.

From a Controls Component

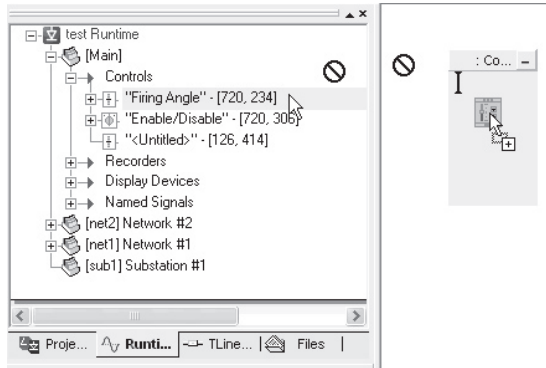
1. Hold down the **Ctrl** key (use the **Shift** key if Enable 'Shift' key to create controls and curves is enabled in the Workspace Settings dialog).
2. Move the mouse pointer over the desired controls component Instance.
3. Select and hold with your **left mouse button**.
4. Drag the mouse pointer over the desired Control Panel title bar and release the mouse button to paste.



Adding a Control Interface to a Control Panel from a Output Channel

From a Runtime Object

1. With the current Project set as active, click the Runtime tab in the Workspace.
2. Navigate to the corresponding Runtime Object for the desired Output Channel (will be in the Controls sub-branch).
3. Select the object and hold with your **left mouse button**.
4. Drag the mouse pointer over the desired Control Panel title bar and release the mouse button to paste.



Adding a Control Interface to a Control Panel from a Runtime Object

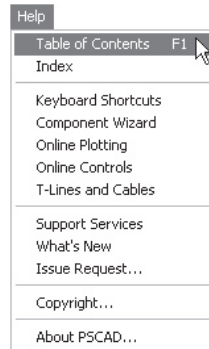
Move/Copy Meters and Curves Between Graphs/Panels

Once a Curve or Meter Interface has been placed in a Graph or Control Panel, it may then be copied or moved to any other Graph or Control Panel present in the Project. In fact, Control/Meters Interfaces and Curves can be moved/copied within the same Graph or Control Panel.

- To **Copy** a Curve/Meter Interface from one Graph/Control Panel to another, hold down the **Ctrl** key and then left-click and hold the object, drag and drop as described above.
- To **Move** a Curve/Meter Interface from one Graph/Control Panel to another, left-click and hold the object, drag and drop as described above.

ACCESSING THE ONLINE HELP SYSTEM

The On-Line Help System Table of Contents (TOC) and Index can be accessed directly through the **Help** menu in the Main Menu Bar, as shown below:



You may also bring up the On-Line Help by simply pressing the **F1** key.

Master Library component specific help can be accessed by one of the following methods:

- Select the component and then press **F1**.
- Right-click on the component and select **Help** from the pop-up menu.
- Edit the component parameters and click the **Help** button.

In addition, other various dialog windows in PSCAD will have a Help button associated with them (usually in the bottom right corner). Click this Help button to bring-up the On-Line Help system topic specific to that dialog. See 'PSCAD On-Line Help System' in Chapter 4 for more details on the help system in general.

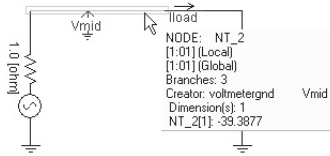
TOOL TIPS

Tool Tip windows (otherwise known as 'Flybys') are a PSCAD feature especially designed for obtaining circuit information 'on the fly.' Tool Tips are available either for providing pop-up help on component instances, or for monitoring electrical or data quantities during a simulation.

To monitor an electrical or data signal during a run:

1. **Left-click** on a blank part of the Circuit canvas (this is to ensure that the Circuit canvas is the active window).

2. Move and hold the mouse pointer over the signal Wire where you want to monitor the data. In a second or two, a Flyby window should pop-up as shown below:



NOTE: Flyby information cannot be read unless the **Store Feed-Forward Signals for Viewing** parameter in the Dynamics section of the Project Settings dialog is disabled. Deselect this option first before using Flybys.

If the mouse pointer is held over a Wire carrying an electrical signal, the Flyby will indicate the voltage in kilovolts at that electrical node. The Flyby shown above indicates that this is EMTDC node NT_2, which is a scalar quantity (i.e. single-phase node). The value of the node voltage at the instant this snapshot was taken was -39.3877 kV.

If the mouse pointer is held over a data (control) signal, the value of the data itself will be shown in the Flyby.

SEARCHING

If the user wishes to search a project for a signal name, Connection name, node number, etc., the Find feature in PSCAD can be used. To bring up the Find dialog window, either:

- Select **Edit | Search...** from the Main Menu bar



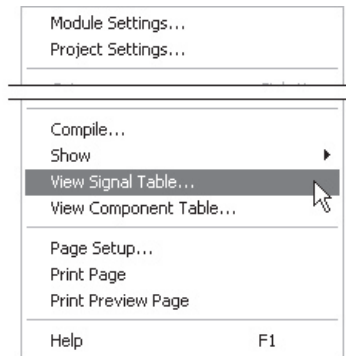
Chapter 5: Features and Operations

- Press **Ctrl + F** on your keyboard, while the Circuit window is open.
- Press the **Find** button in the Main Toolbar.



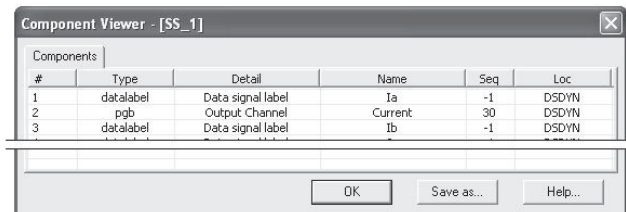
COMPONENT AND SIGNAL TABLES

The Component Viewer and Signal Tables are available in spreadsheet viewing format. Each summarize and present data in an organized and efficient manner. To invoke table viewer, right-click on a blank part of the PSCAD canvas in Circuit view. Select either **View Component Table...** or **View Signal Table...**



Component Table

The Component Table lists all Component Instances currently existing in the Project. Along with the component name and type, some other important information is also summarized to help users in debugging or understanding the Project.



The columns displayed in this viewer are described below:

- **#:** Just a simple list number.
- **Type:** The Component Definition Name, on which the Instance is based.
- **Detail:** The Description of the Component Definition.
- **Name:** If the component has an input parameter entitled 'Name,' then it's contents will be displayed here. For example, a voltmeter component will display the name of the measured signal created.
- **Seq:** The System Dynamics sequence number of the component instance. See 'System Dynamics' in Chapter 2 of the EMTDC Manual and 'Component Ordering' in Chapter 11 of this manual for more.
- **Loc:** The location within the System Dynamics where the Instance resides. This can be either DSDYN or DSOUT.

This viewer also includes a mechanism by which you may navigate directly to a Component Instance in the list. Simply right-click on the row representing the Instance and select **Point to...**

23	ammeter	Current Meter		170
24	nodelabel	Electrical node label	Copy	60
25	resistor	Resistor	Point to	-1

Signal Table

The Signal Table summarizes all pertaining to both control and electrical signals in the Project. Information is given, such as the EMTDC control signal variable name and from where the signal is sourced. For electrical signals, node name and index is given.

Control Signals		Electrical Signals			
Name	Type	Local Index	Global Index	Source	Sinks
IT_1	DIGITAL	1	1	nodeloop	1
RT_1	ANALOG	1	1	var_switch	'bloding'
ISC	ANALOG	21	21	sources	'SRC'
VfAa	ANALOG	22	22	voltmeter	'VfAa'
JfAa	ANALOG	23	23	source	'dc1'
VAFa	ANALOG	24	24	voltmeter	'VAFa'

Control Signals		Electrical Signals			
Name	Flags	Local Index	Global Index	Creator	branches
GND	S	0:00	0:00	ground	26
NT_1	R	1:01	1:07	pin	4
NT_2	R	1:02	1:06	pin	4
NT_20	S	1:27	1:27	gogocou	2
NT_21	S	1:20	1:26	g66200	2
Nb		1:21	1:31	nodeloop	5
Na		1:22	1:30	nodeloop	5

Chapter 5: Features and Operations

The columns displayed in this viewer are described below:

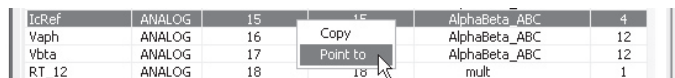
Control Signals:

- **Name:** EMTDC signal variable name. That is, the name given to the signal node by the PSCAD compiler for use by EMTDC.
- **Type:** The signal type. Can be ANALOG (REAL), DIGITAL (INTEGER) or LOGICAL.
- **Local Index:** A local (i.e. Module based) index number given to the signal.
- **Global Index:** A global (i.e. Project based) index number given to the signal.
- **Source:** The Component Instance, from which the signal is created.
- **Sinks:** The number of points, to which the signal is being sent.

Electrical Signals:

- **Name:** EMTDC electrical node name. That is, the name given to the electrical node by the PSCAD compiler for use by EMTDC.
- **Flags:** Electrical node type: S = Switched, R = Removable. See 'Electrical Node Types' Chapter 9 for more.
- **Local Index:** A local (i.e. Module based) index number given to the signal. For example, **1:01** -> Subsystem #1, Node #1.
- **Global Index:** A global (i.e. Project based) index number given to the signal.
- **Creator:** The Component Instance, by which the signal is created.
- **Branches:** The total number of electrical branches attached to the node.

This viewer also includes a mechanism by which you may navigate directly to a Control Signal or Electrical Node in the list. Simply right-click on the row representing the Instance and select **Point to....**



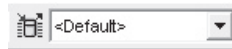
The screenshot shows a table with 6 columns and 5 rows. A context menu is open over the second row, with the 'Point to' option selected. The table data is as follows:

IcRef	ANALOG	15	15	AlphaBeta_ABC	4
Vaph	ANALOG	16	Copy	AlphaBeta_ABC	12
Vbta	ANALOG	17	Point to	AlphaBeta_ABC	12
RT_12	ANALOG	18	18	mult	1

CONTROL SETTINGS

The Control Settings feature was introduced in PSCAD V4, and provides users with the ability to save the settings of all Dial, Switch and Slider control components in a Project to a Template. When the Project is reloaded, these settings can be reset immediately by selecting the Template. It is possible to store multiple Templates so that the user can easily switch between one Template to another, without having to manually reset all the control components.

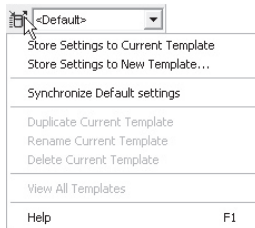
The Control Settings feature can be accessed through the **Runtime Bar**.



If you cannot see this tool bar, go to the Main Menu bar and select **View | Runtime Bar**.

Storing Settings to a Template

Before storing your control settings to a Template, first ensure that all Dial, Switch and Slider control components have been set, and that the current Project is the Active Project. Press the **Control Data Menu for Active Project** button to bring up the Control Data Menu pop-up as shown below:



Select either **Store Settings to Current Template** or **Store Settings to New Template...** depending on what you want to do. If this will be the first Template you have created for this Project, select the latter.

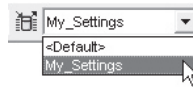
NOTE: You may store all settings to the default Template if you wish. Simply ensure that **<Default>** appears in the control setting window and select **Store Settings to Current Template**. Also, ensure that all controls possess a distinct name, otherwise an error message will be posted.

Chapter 5: Features and Operations

If creating a new Template, a dialog window will pop-up – enter a name for the Template.

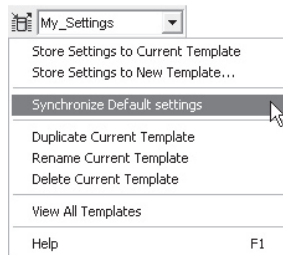


When finished, press the **OK** button. Your control settings will now be stored in this new Template. Whenever you reload the Project, or want to revert back to these settings, simply left-click the down arrow on the Template drop list and select the Template.



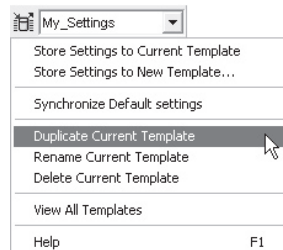
Synchronizing Default Settings

At any time, the Project default control settings may be synchronized with the current settings. Press the **Control Data Menu for Active Project** button to bring up the Control Data Menu pop-up. Select **Synchronize Default Settings**.



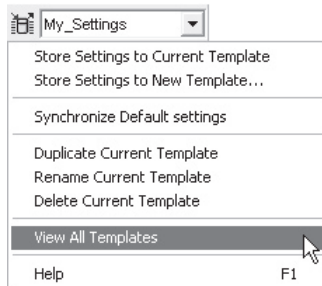
Duplicate, Rename or Delete Template

You can duplicate, rename or delete a Template by simply selecting the corresponding function from the Control Data Menu pop-up as shown below:



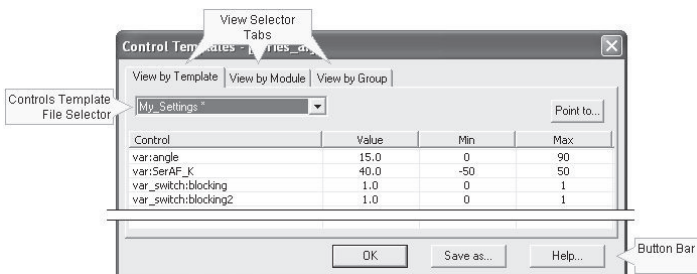
Template Settings Viewer

Once at least one Template has been created for a Project, PSCAD provides a Template Settings Viewer so that users can view and verify control settings in a visually organized environment. To bring up the Template Settings Viewer, simply select the corresponding function from the Control Data Menu pop-up as shown below:



Features and Functions

The Template Settings Viewer window contains a few simple features to help view and organize Template settings data. The viewer consists mainly of three View Selector Tabs, where the settings data may be viewed by Template, Module or by Control Group.



In the case of **View by Template**, the Template view may be switched by using the Template Selector drop list. Under **View by Module** and **View by Group**, this drop list becomes a Module selector and Group selector respectively.

NOTE: The current Template is marked with an asterisk (*).

The button bar includes three buttons. The **OK** button closes the viewer window. The **Save As...** button allows the user to save the

Chapter 5: Features and Operations

data in the current view to a formatted text file. The **Help** button of course invokes the online help.

The **Point To...** button may be used to have PSCAD point directly to the control of interest. Simply select the desired control from the display list and click the button:



GLOBAL SUBSTITUTIONS

Substitutions are well defined when used with Component Definitions. These types of substitutions operate using Component input parameters as a source, while using the Definition to define the actions. This means that all components of a certain type will process substitutions in the same way. If substitutions are required on a per Instance basis however, then this process cannot be used. In PSCAD versions previous to V4.2, implementations of a Component Instance based substitution was referred to as a 'Global Constant,' which allowed the user to enter a defined text string. This was then inserted as a numerical value into an Input Field, but was limited for many other actions.

This mechanism has been reworked and the concept of a 'Global Constant' has been replaced by a Project based Global Substitution. A Global Substitution is a raw text string that is represented by a unique identifier, which can be used within any Component Instance input parameter. The big difference with the new approach is the process is completely general, and does not require that a substitution be of a numerical type. This allows for raw text substitutions combining both pre-fix and post-fix patterns. The syntax is the same as used for Component Definition substitutions:

$\$(\langle\text{Key}\rangle)$

The substitution contains an item $\langle\text{Key}\rangle$, where:

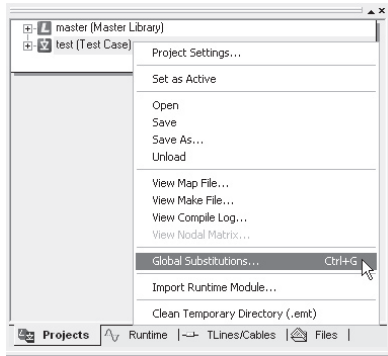
- **$\langle\text{Key}\rangle$** : Item key (defined global parameter) that is to be placed at the substitution point.

As with other substitutions, there is no Context scope since the context is always that of the Project. This text will be substituted at

PSCAD

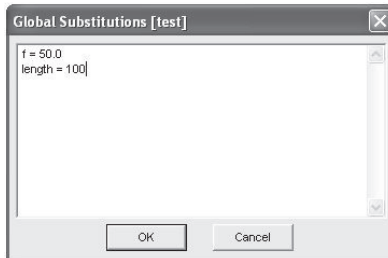
any point within a Component Instance input parameter. The input parameter containing the Global Substitution will be pre-processed before it is returned for use within the component. This allows the input parameter to act exactly as if the user entered that data directly.

To define a Global Substitution, right-click on the desired project in the Workspace window and select Global Substitutions... or simply press Ctrl + G on your keyboard.



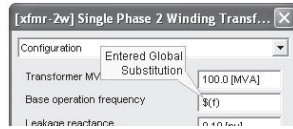
Either of these procedures will bring up the Global Substitutions dialog window.

Enter the constant directly into the dialog. For example, say a user wants to define the system frequency globally as 50 Hz, and the transmission line length globally as 100 km in a project. This would be entered as follows in the Global Substitutions dialog:

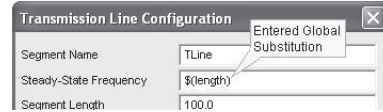


The user would then enter 'f' or 'length' in the appropriate component input parameter field, according to the syntax given above:

Chapter 5: Features and Operations



Entering 'f' in Single-Phase Transformer Parameters Dialog



Entering 'length' in Transmission Line Configuration Dialog

Global Substitutions are saved within the Project file itself, so once defined, they remain as a Project setting until they are removed from the Global Substitutions dialog.

UNIT SYSTEM

In past versions of PSCAD, units were entered for display only, and provided no means to convert or scale component input parameters. The primary purpose of these units was simply to indicate to users that a particular input parameter was expected in the displayed unit. For example, if a default unit for a parameter was set as [kA], then a value of 10 Amps would need to be entered as 0.01 [kA].

In PSCAD V4, component parameter units will now perform limited conversion and scaling, depending on the unit entered and the default (or Target) unit for that particular input parameter. The Unit System includes base units for time, length, weight and speed (both translational and rotational), as well as of course electrical units, such as voltage, current and power.

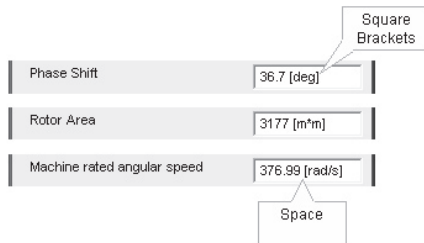
NOTE: The Units System is not backwards compatible to PSCAD versions previous to v4.2. See Backward Compatibility for more. See <http://physics.nist.gov> for more details on the reference standard used to create the PSCAD units system.

Enabling the Unit System

To enable the Unit System, click the associated check box in the Dynamics tab of the Project Settings dialog.

Unit Format

Units are entered exclusively into component Input Fields and may only be associated with numerical input data. That is, units are invalid when using input variables or Global Constants. All entered units must be enclosed with square brackets [`<Unit>`], where the brackets are at least one space from the number. For example:



Examples of Units Entered in Component Input Fields

Base Units

The foundation of the PSCAD Unit System are the base units. Base units represent all units that are recognized by the Unit System. If a type of unit does not exist as a base unit, then it will be ignored. This does not mean however, that the unit cannot be used. It simply means that the unrecognized unit will be used for display only, and will hence not modify the quantity. Scaling and conversions cannot be performed on non-standard base units. Most base units adhere to the International System of Units (SI).

The following lists the valid (standard) base units recognized by the Unit System. All symbols are case sensitive and must appear exactly as shown to maintain validity:

Chapter 5: Features and Operations

	Name	Symbol	Conversion Factor	Description
	Volt	V		Electrical voltage
	Ampere	A		Electrical current
	Ohm	ohm		Electrical resistance
	Siemen	S		Electrical conductance 1
	mho	mho		Electrical conductance 2
Electrical	Watt	W		Electrical power 1
	Volt-Amps	VA	1 VA = 1.0 W	Electrical power 2
	Horsepower	hp	1 hp = 746.0 W	Electrical power 3
	Farad	F		Electrical capacitance
	Henry	H		Electrical inductance
	Tesla	T		Magnetic flux density
	Second	s		Time in seconds 1
	Second	sec		Time in seconds 2
Time	Minute	min	1 min = 60 s	Time in minutes
	Hour	hr	1 hr = 3600 s	Time in hours
	Day	day	1 day = 86400 s	Time in days

PSCAD

	Metre	m		Length in metres
	Inch	in	1 in = 0.0254 m	Length in inches
Length	Feet	ft	1 ft = 0.3048 m	Length in feet
	Yard	yd	1 yd = 0.9144 m	Length in yards
	Mile	mi	1 mi = 1609.344 m	Length in miles
	Gram	g		Weight in grams
Weight	Pound	lb	1 lb = 453.59237 g	Weight in pounds
	Revolutions	rev		Revolutions (1 [rev] = one complete revolution)
Rotational	Radian	rad	1 rad = $1/2\pi$ rev	Angle in radians
	Degree	deg	1 deg = $1/360$ rev	Angle in degrees
	Cycles per Second	rev/s		Revolutions per second
Frequency (Rotational Speed)	Cycles per Second 2	Hz	1 Hz = 1 rev/s	Cycles per second
	Revolutions per Minute	rpm	1 rpm = $1/60$ rev/s	Rotational speed in revolutions per minute
	Per-Unit	pu		Per-unit quantity
Other	Percent	%	1 % = 0.01 pu	Percent quantity

Chapter 5: Features and Operations

Exceptions

It is important to note the following units, which were used in previous versions of PSCAD, are no longer valid base units (i.e. are not recognized by the Unit System). All Master Library models have been changed to reflect this:

- **ohms** and **mhos**: The 's' has been dropped on both of these units to maintain consistency with SI standards.
- **Degree Symbol °**: The degree symbol is no longer valid and is replaced with **deg**, as detailed above.
- **Per-Unit [p.u.]**: Periods are no longer valid in the per-unit symbol and should be written as **pu**, as detailed above.

NOTE: Exceptions may be used, but may not be supported by future versions.

Aliases

Aliases are short forms for writing specific compound units. There are currently two aliases existing in the Unit System:

- **Hz** = 1 rev/s
- **rpm** = 1 rev/min

NOTE: Aliases may **not** be used within Compound Units.

Prefixes

The Unit System utilizes a limited list of SI prefixes in order to allow for scaling of Base Units. Prefixes must precede a valid Base Unit, and may be inserted anywhere within Compound Units.

The following table lists all valid prefixes:

Name	Symbol	Scale Factor
tera	T	10^{12}
giga	G	10^9
mega	M	10^6
kilo	k	10^3
milli	m	10^{-3}
micro	u	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}

Target Units

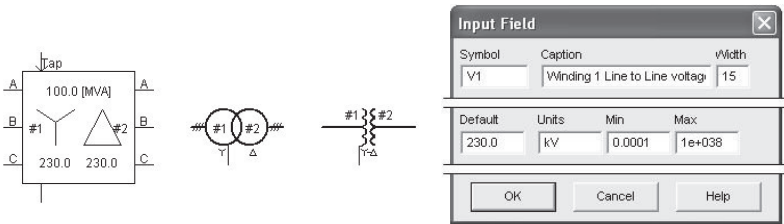
The Unit System will determine the final conversion or scaling factor to apply, based on the Target Unit of the Input Field. The Target Unit is the symbol entered in the **Units** field (i.e. the default unit) in the Input Field Property Settings dialog.

NOTE: You can view the set Target Units of a component by invoking the View Parameters/Computations dialog.

Target units are not limited to the Base Units alone, and may include Prefixes by default (i.e. kA): In instances such as these, any Prefixes in the Target Unit will be considered if further scaling is performed later on. In fact, this is quite common in the Master Library, where many Target Units are specified in [kA], [kV] or [μ F].

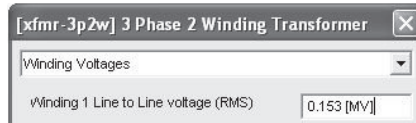
EXAMPLE 5-4:

The Master Library component ‘3-Phase 2-Winding Transformer’ contains an input Parameter called ‘Winding 1 Line to Line Voltage,’ whose Target Unit is specified as ‘kV’ in the Units field.



3-Phase 2-Winding Transformer Input Field Property Settings ('V1')

A user enters data into the ‘Winding 1 Line to Line Voltage’ field as **0.153 [MV]**. Given that the Target Unit contains the Prefix ‘k,’ PSCAD will understand that any quantity entered in this Parameter field must be converted back to kilovolts (not the Base Unit of volts [V]). Therefore in this case, the quantity will be multiplied by a scale factor of 1000 to convert it from 0.153 [MV] back to 153 [kV], before it is considered for further calculations.



3-Phase 2-Winding Transformer Parameter Page

Whether or not Target Units include Prefixes is normally of no concern, unless of course a new component is being designed. Provided that the base of the entered unit matches that of the Target, all scaling and conversion is performed automatically.

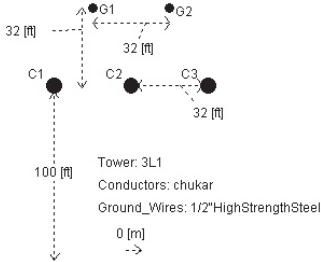
Unit Conversions

The most useful aspect of the Unit System is the ability to convert one unit to another, be it an Imperial/Metric conversion or simply converting from one form to another, such as radians to degrees.

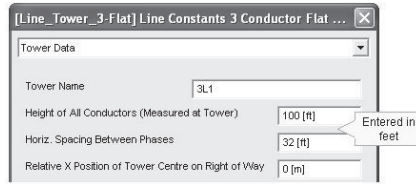
Converting units from one form to another is relatively straightforward. The only rule is that the conversion must take place within the same Base Unit class; for example, [m] to [ft] (both units measure length) or [sec] to [hr] (both units measure time). Valid Prefixes may be included in the conversion as well: For instance, [km] to [mi].

EXAMPLE 5-5:

A user is designing a transmission line tower. The default units for the tower dimensions are in metres, but the user's specification sheets give the dimensions in feet. The Unit System will allow the user to enter this data directly in feet, without the need to convert to metres.



Transmission Line Tower Component



Transmission Line Tower Parameter Page

PSCAD will automatically convert all units entered in feet, back to metres before the Line Constants Program is called to solve the line.

Compound Units

The Unit System will recognize three types of Arithmetic Operator within the unit brackets, in order to allow for combining (or compounding) units together. These are:

Arithmetic Operator	Description
*	Multiply
/	Divide
^	Exponent

When dealing with Compound Units, it is important to note some simple rules. Failure to follow these rules may result in invalid units conversion:

- The sequence in which the Arithmetic Operators occur in the entered unit must match that of the Target Unit.

	Entered	Target
Correct:	[hp*min/MVA]	[MW*s/MVA]
Incorrect:	[hp/MVA*min]	

- The total number of Arithmetic Operators in the entered unit must match that of the Target Unit. That is, you

Chapter 5: Features and Operations

cannot substitute an exponent for multiple 'Divisions' or 'Multiplications.'

	Entered	Target
Correct:	[ft*ft]	[m*m]
Incorrect:	[ft^2]	

	Entered	Target
Correct:	[lb*ft*s^-2]	[kg*m*s^-2]
Incorrect:	[lb*ft/s*s]	

- All units involved in a Compound Unit must be of an equivalent class (this is obvious of course!).

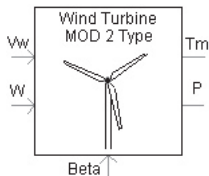
	Entered	Target
Correct:	[lb*ft*s^-2]	[kg*m*s^-2]
Incorrect:	[lb*deg*min^-2]	

- Multiple 'Divide' symbols are not allowed - only one 'Divide' per Compound Unit may exist.

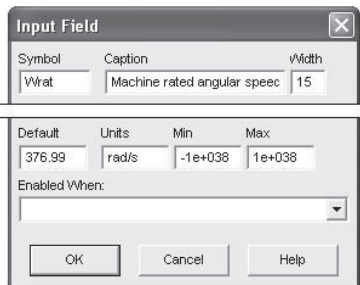
	Entered	Target
Correct:	[ft/min*min]	[m/s*s]
Incorrect:	[ft/min/min]	

EXAMPLE 5-6:

The Master Library component 'Wind Turbine' contains an input Parameter called 'Machine rated angular speed,' whose Target Unit is specified as 'rad/s' in the Units field.

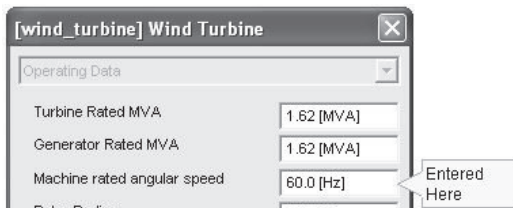


Wind Turbine



Input Field Properties Dialog (Wrat)

A user enters data into this Parameter as **60.0 [Hz]**. This is a valid unit in this case as both 'Hz' and 'rad/s' are essentially the same type of measure, where 2π [rad/s] = 1 [rev/s] = 1 Hz (see Base Units above).

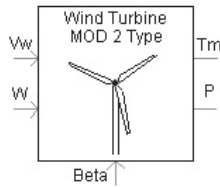


Wind Turbine Parameter Page

EXAMPLE 5-7:

The Master Library component 'Wind Turbine' (described above) also contains an input Parameter called 'Air Density,' whose Target Unit is specified as 'kg/m³' in the Units field.

Chapter 5: Features and Operations



Wind Turbine

Symbol	Caption	Width
Airden	Air density	15

Default	Units	Min	Max
1.225	[kg/m ³]	0	1e+038

Enabled When:

OK Cancel Help

Input Field Properties Dialog (Airden)

A user enters data into this Parameter as the equivalent in Imperial Units **0.07647 [lb/ft³]**. The units converter will apply the appropriate scale factors to this number so that the quantity will appear to EMTDC as it is still in 'kg/m³.'

[wind_turbine] Wind Turbine

Operating Data

Turbine Rated MVA	1.62 [MVA]
Generator Rated MVA	1.62 [MVA]
Machine rated angular speed	376.99 [rad/s]
Rotor Radius	31.5 [m]
Rotor Area	3177 [m ²]
Air density	0.07647 [lb/ft ³]

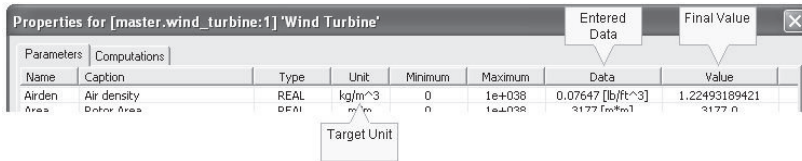
Entered Here

Wind Turbine Parameter Page

Verifying Unit Conversions

It is generally a good idea to ensure that any unit conversions performed in a component are verified before proceeding with the simulation. This can be accomplished easily by using the Parameters/Computations Viewer. This dialog displays the Target Unit, the entered data, and the final value following the conversion.

The following diagram shows these three parameters following the changes made to the 'Air Density' input, as outlined in Example 5-7.



Wind Turbine Air Density Parameter

Backward Compatibility

As mentioned above, previous versions of PSCAD (i.e. previous to V4.2) do not contain a units system. Units are used for display only and do not affect the input quantities whatsoever. Therefore, if you intend to save your Project back to an older version, care must be exercised if you have utilized the Units System.

Make sure the following points are addressed before saving your Project to a previous version:

- **Save the Project Under Another Name:** Before making any changes, save the Project under another name in V4.2 format. For example, '<ProjectName>_v42.psc.'
- **Convert All Entered Units Back to the Target:** All component input parameters where the entered unit does not match the Target Unit must be converted so that the entered unit and Target Unit are the same. The Parameters/ Computations Viewer can be used to help with this.

EMTDC OUTPUT FILES

EMTDC output files are formatted text files, which organize all data into columnar format. Each column, except the first, which is ALWAYS time, represents the recorded data from a respective Output Channel component existing within the project. For example, if two Output Channel components exist in the project, then three columns of data will appear in the EMTDC output file. The following is a segment of text from a typical EMTDC output file:

Chapter 5: Features and Operations

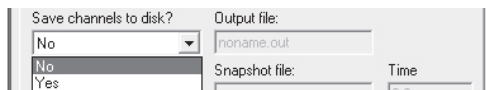
Test Case			
0.00000000000000	0.00000000000000	0.00000000000000	0.00000000000000
0.10000000000000E-02	0.00000000000000	0.00000000000000	0.00000000000000
0.20000000000000E-02	0.86727047422974	0.86727047422974	0.86727047422974
0.30000000000000E-02	1.6650619394029	1.7674567004163	1.7674567004163
0.40000000000000E-02	1.9545665157651	2.2542364437667	2.2542364437667
0.50000000000000E-02	2.0221282586499	2.8373003607589	2.8373003607589
0.60000000000000E-02	1.9264422562260	4.0202957514613	4.0202957514613
0.70000000000000E-02	2.3912531836698	5.5547061594874	5.5547061594874
0.80000000000000E-02	2.8769239640036	6.5791178737352	6.5791178737352
0.90000000000000E-02	2.8473253982397	6.9456914675731	6.9456914675731
0.10000000000000E-01	2.3232656122503	7.1369144410646	7.1369144410646
0.11000000000000E-01	1.2446128466462	7.1669414384869	7.1669414384869
0.12000000000000E-01	1.6208317681211	7.0023750435365	7.0023750435365
0.13000000000000E-01	1.6458908605563	7.1529925186834	7.1529925186834
0.14000000000000E-01	4.2422849293514	9.3453281849245	9.3453281849245

The project description is written as the first row of text at the top of the file. The first column of data is always the EMTDC simulation time. The subsequent columns are not labelled - see Column Identification and the Information File below for more details on this.

EMTDC output files may be used for waveform analysis by a selected post-processing software package. As they are formatted in a delimited columnar format, they can be easily imported into most graphing or data analysis programs. EMTDC output files are given the extension '*.out' and are stored in the project temporary directory.

Creating Output Files

EMTDC output files are created by choosing **Yes** in the **Save channels to disk?** drop list, in the Runtime section of the Project Settings dialog.



Multiple Output Files

The maximum amount of columns per output file is 11 (including the time column). Therefore, if more than 10 Output Channel components exist in a project, multiple output files will be created. For example, if your project contains 23 Output Channels, a total of three output files will be created.

PSCAD

The naming convention for multiple output files is to simply append a sequential number as a suffix. For example, if the output file is named 'abc.out,' and there are three files as described above, the output files would be named 'abc_01.out,' 'abc_02.out' and 'abc_03.out.' This sequential numbering is important when identifying data columns.

Column Identification and the Information File

As mentioned above, EMTDC output file columns are not labelled. In order to determine which column is what, an information file (*.inf) is also created along with the output file(s), that contains cross-referencing information. The information file will be named the same as the output file primary file name. For example, if the output file name is 'abc.out,' the information file will be named 'abc.inf.' Only one information file is created, even for multiple output files.

A typical information file is shown below for a project containing three output channels.

PGB(1) Output Desc="Fund - mag" Group="Main" Max=25.0 Min=0.0 Units=""
PGB(2) Output Desc="2nd harm - mag" Group="Main" Max=25.0 Min=0.0 Units=""
PGB(3) Output Desc="3rd harm - mag" Group="Main" Max=25.0 Min=0.0 Units=""

At the extreme left is the Output Channel number (i.e. PGB(1), PGB(2), etc.). This number indicates the sequence in which the Output Channel data is written to an output file. In other words, this number corresponds to the output file column number. Remember however, that the first column in the output file is time and is not counted. Therefore, Output Channel 2 (PGB(2)) will actually be the third column from the left in the output file. This column can then be identified using the corresponding Output Channel name (i.e. 'Desc'). In this case, Output Channel 2 happens to be '2nd harm - mag.'

The Output Channel numbers in the information file will continue in sequence with the number of Output Channel components in the project. That is, if there are 50 Output Channels in the project, there will be 50 rows in the information file, numbered up to 50. We already know however, that a single output file will only hold up to 11 columns of data (including time). For 50 Output Channels then, PSCAD will create five output files, where the columns are numbered ignoring the time columns in each file. For example, column five in the 3rd output file would be Output Channel 34.

Chapter 5: Features and Operations

Here is a simple formula to help identify an output file column:

Single Output File:

$$\text{Output Channel \#} = \text{Output File Column \#} - 1$$

Multiple Output Files:

$$\text{Output Channel \#} = \text{Output File Column \#} - 1 + (10 \times \text{Output File \#})$$

Online Plotting and Control

PSCAD contains a few special Runtime objects designed to give the user online control of input data, as well as the ability to record and display output data.

Special objects, such as Graph Frames, Graphs, Curves and a few types of Meters are used to interactively display EMTDC output data, both during a run and following completion of a simulation. Control Interfaces are especially designed to give the user online access to EMTDC input variables, so that these variables (and hence the simulation results) can be altered during the course of the simulation run.

The plotting and online tools have been reworked in PSCAD V4, but still maintain a likeness to those found in earlier versions. For example, data could only be plotted with respect to time in previous versions of PSCAD. In V4, the XY Plot, PolyMeter and PhasorMeter tools have been added to allow for both the plotting of one set of data versus another, as well as various representations in the phasor domain.

Portions of Curve data, whole Graphs, or even entire Graph Frames may be copied as a picture, meta-file or comma separated variable (*.csv) files for placement in documents and reports. Many other intuitive features not found in earlier versions are also available.

PREPARING DATA FOR CONTROL OR DISPLAY

The PSCAD environment is a graphical user's interface to the EMTDC simulation engine. Therefore, in order to control input variables and/or view simulation data, the user must provide EMTDC with instruction on which variables to make available for viewing or control. This process is performed graphically in PSCAD through a number of special components, sometimes referred to as 'Runtime Objects.'

In order to record, display (i.e. plot) or control any data signals within the PSCAD environment, the signal must be first linked to a Runtime object. Runtime objects are organized into three main groups:

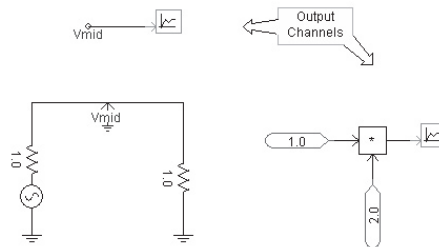
Chapter 6: Online Plotting and Control

- **Controls:** Slider, Dial, Switch and Push Button
- **Recorders:** Output Channel and RTP/COMTRADE Recorder
- **Display Devices:** Control Panel, Graph Frame, XY Plot, PolyMeter and PhasorMeter

Each Runtime object performs a specific task, and may be used in combination with others to control and/or display data. For more on the organization and grouping of Runtime objects, see the section entitled 'Runtime' in Chapter 3.

Channeling Output Data

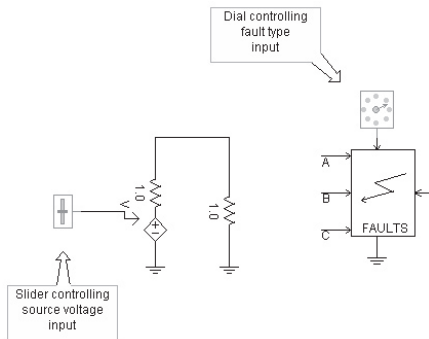
Channeling output data refers to directing data signals for either on-line display in a Graph or Meter, or for output to an Output File. This is accomplished by 'channeling' the desired signal into an Output Channel component. For example, the image below shows how to channel a signal from a Voltmeter component (named 'Vmid'), as well as how to channel an unnamed signal directly (in the PSCAD Circuit canvas):



NOTE: Output channel components can-not be directly connected to an electrical wire. They may be directly connected to any data signal, however.

Controlling Input Data

Controlling input data refers to using one of the Controls Runtime objects (i.e. Slider, Dial, Switch or Push Button) as a source for a certain data signal. This is accomplished by simply adding a Control object to the PSCAD Circuit canvas. This signal can be input at any valid location in the circuit, as shown below:

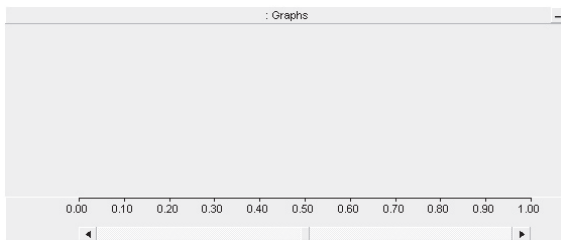


NOTE: The Control objects cannot be manually adjusted (as they appear above) until they are linked to a Control Interface.

GRAPH FRAMES

A Graph Frame is a special Runtime object used for accommodating Overlay or Poly Graphs and can be placed anywhere on the canvas in Circuit view. Once a Graph Frame has been added, you may then proceed to add as many Graphs to it as you wish.

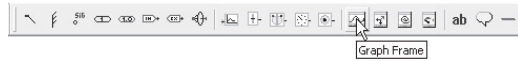
Graph Frames are used exclusively for plotting Curves versus time. That is, the Graph Frame horizontal axis is always the EMTDC simulation time. If you need to a plot a Curve as a function of another variable, see the section entitled 'XY Plots' in this chapter.



Adding a Graph Frame

Open the project in the Circuit window of the Design Editor. Right-click on a blank portion of the page and select **Add Component | Graph Frame**, or press the **Graph Frame** button in the Control Palette bar.

Chapter 6: Online Plotting and Control

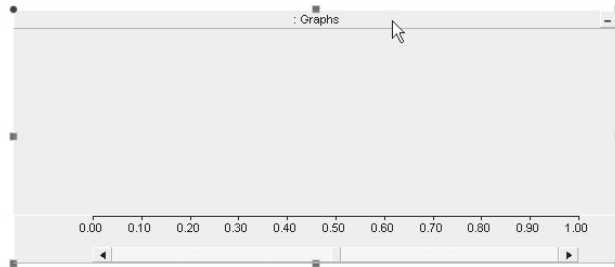


If you cannot see the Control Palette, go to the **View** menu and ensure **Control Palette** is selected.

Moving and Resizing a Graph Frame

To move a Graph Frame, move the mouse pointer over the title bar and then **left-click and hold**. Drag the frame to where it is to be placed and release the mouse button.

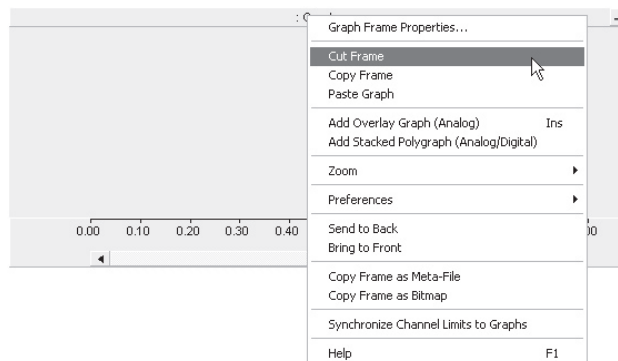
To resize, move the mouse pointer over the title bar and **left-click** to select the Graph Frame. Grips should then appear around the outer edge as shown below.



Move the mouse pointer over one of the grips. **Left-click and hold** and then drag then move the pointer to resize.

Cut/Copy Frames

Right-click over the Graph Frame title bar and select **Cut Frame** or **Copy Frame** respectively.



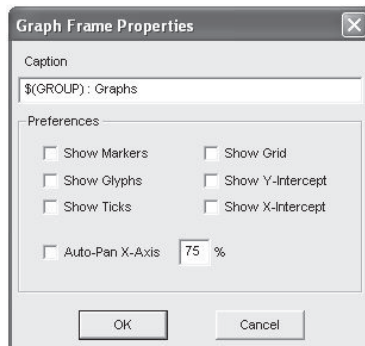
Once a Graph Frame has been cut or copied it may then be pasted into another location in the Project (along with its contents).

Paste Frame

Cut or copy a Graph Frame as described above. Right-click over a blank area of the Project page in Circuit view and select **Paste**. A Graph Frame may be pasted multiple times.

Adjusting Frame Properties

To access the Graph Frame Properties dialog, left double-click the Graph Frame title bar, or right-click over the title bar and select **Graph Frame Properties....**



The properties available in this dialog are as described below:

- **Caption:** Enter a title for the Graph Frame (this text will appear in the Graph Frame title bar). The default text may appear a bit cryptic: This syntax is used as a naming convention for grouping objects in the Workspace. For more information on this syntax, see 'Grouping of Runtime Objects' in this chapter.

Preferences:

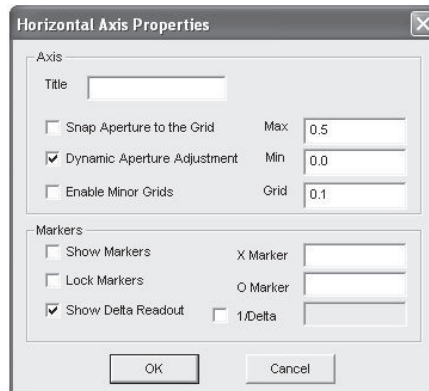
- **Show Markers:** Select this option to show the **X** and **O** Markers on all Graphs in the Graph Frame.
- **Show Glyphs:** Select this option to show Glyph symbols on all Curves in all Graphs in the Graph Frame.
- **Show Ticks:** Select this option to show Ticks along the Y-Intercept in all Graphs in the Graph Frame.
- **Show Grid:** Select this option to show the Grid in all Graphs in the Graph Frame.

Chapter 6: Online Plotting and Control

- **Show Y-Intercept:** Select this option to show the Y-Intercept in all Graphs in the Graph Frame.
- **Show X-Intercept:** Select this option to show the X-Intercept in all Graphs in the Graph Frame.
- **Auto-Pan X-Axis:** This feature is equivalent to the 'Follow End Of Run' feature available in PSCAD V3. The Auto-Pan X-Axis feature however, allows the user to adjust the panning action. The input field directly beside this check box accepts an input representing the percentage of the currently viewed Graph window (or aperture). For example, if the total x-axis view is 0.1 seconds, a 10% Auto-Pan setting will pan the viewing window every 0.01 seconds.

Adjusting Horizontal Axis Properties

To access the Horizontal Axis Properties dialog, left double-click over the Graph Frame horizontal axis, or right-click over the horizontal axis and select **Axis Properties...**



NOTE: Adjusting the horizontal axis properties will affect all Graphs in the Graph Frame.

Axis:

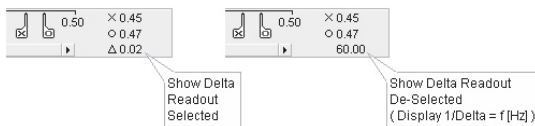
- **Title:** Enter a title for the x-axis. This text will appear in the bottom-left corner of the Graph Frame, directly beside the x-axis.
- **Snap Aperture to Grid:** Select this feature so that when using Dynamic Aperture Adjustment, the aperture view will snap to the major grid while scrolling.

PSCAD

- **Dynamic Aperture Adjustment:** Select this option to enable Dynamic Aperture Adjustment (i.e. horizontal scroll).
- **Enable Minor Grids:** When selected, minor grid ticks will appear on the Graph Frame horizontal axis. Minor grids will always show the halfway point between major grid points, and are not labelled.
- **Max:** Sets the maximum time of the viewed range.
- **Min:** Sets the minimum time of the viewed range.
- **Grid:** Sets the time between the axis major grid points. Major grid points are labelled on the graph frame horizontal axis.

Markers:

- **Show Markers:** Select this option to show the **X** and **O** Markers.
- **Show Delta Readout:** Select this option to display the time difference (i.e. Δt) between the **X** and **O** Markers. When this option is disabled, the equivalent $1/\Delta t$ value (i.e. frequency in Hz) will be displayed.

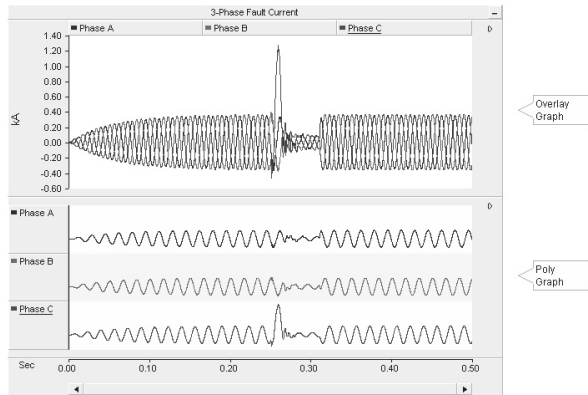


- **X Marker:** Enter the position (in seconds) to place **X** Marker.
- **O Marker:** Enter the position (in seconds) to place **O** Marker.
- **1/Delta:** Select this option to enter the frequency (i.e. $1/\Delta t$) between Markers.

GRAPHS

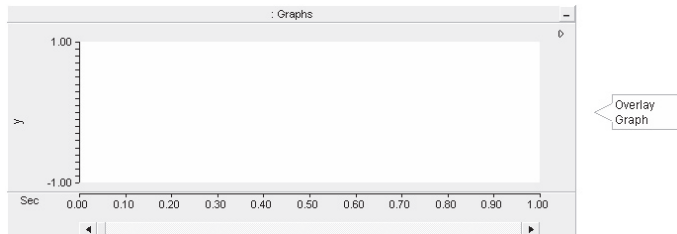
A Graph is a special Runtime object, which can reside only inside a Graph Frame. There are two types of Graphs available in PSCAD: Overlay and Poly Graphs. A single Graph may hold and display multiple Curves, where all Curves in a Graph are based on the same y-axis scale.

The following illustrates a Graph Frame with an Overlay Graph at the top and a PolyGraph below it.



Adding Graphs to a Graph Frame

Graph Frames may accommodate single or multiple Graphs. To add one or more Graphs, right-click on the Graph Frame title bar and select **Add Overlay Graph** or **Add Poly Graph**. You can also add an Overlay Graph directly by pressing the **Ins** key on your keyboard with your mouse pointer over the Graph Frame.



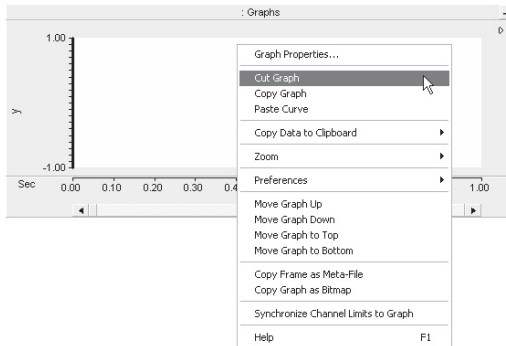
Graph Order

Once multiple Graphs have been added to a particular Graph Frame, you may change the order in which they appear. Right-click over the graph to be moved and select one of the following:

- Move Graph Up
- Move Graph Down
- Move Graph to Top
- Move Graph to Bottom

Cut/Copy Graphs

Right-click over the Graph to be cut (removed) or copied and select **Cut Graph** or **Copy Graph** respectively.



Once a Graph has been cut or copied it may then be pasted into the same or another Graph Frame.

Paste Graph

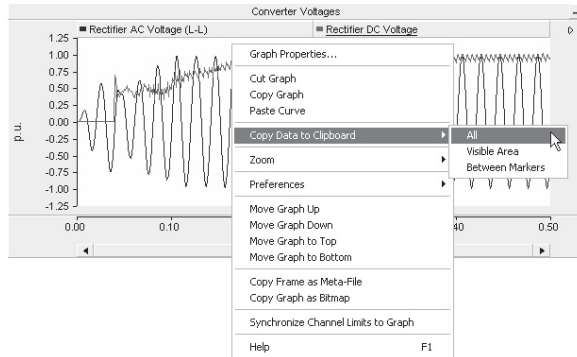
Cut or copy a Graph as described above. Right-click on the Graph Frame title bar and select **Paste Graph**. A Graph may be pasted multiple times, where each paste will replicate the entire Graph.

Copy Data to Clipboard

If a simulation has been run and your Graph contains Curve data, you have the option of copying all or a portion of all Curve data in the Graph to the clipboard. Right-click over the corresponding Graph and select **Copy Data to Clipboard** and then select one of the following from the pop-up menu:

- **All:** Copies all Curve data available.
- **Visible Area:** Copies all Curve data visible in the Graph window.
- **Between Markers:** Copies only Curve data situated between Markers. Note that **Show Markers** must be selected in the Axis Properties dialog.

Chapter 6: Online Plotting and Control



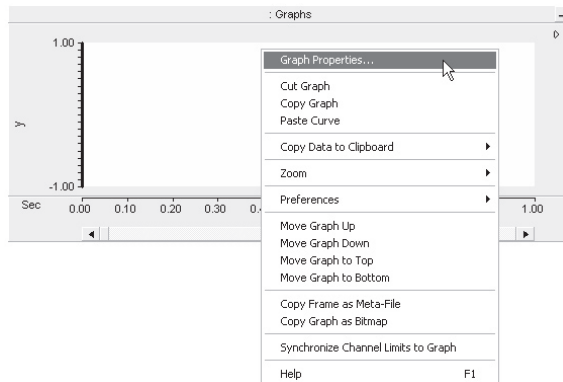
The data is copied as 'Comma Separated Variables' (*.csv) format for easy migration into common data analysis software.

Overlay Graphs

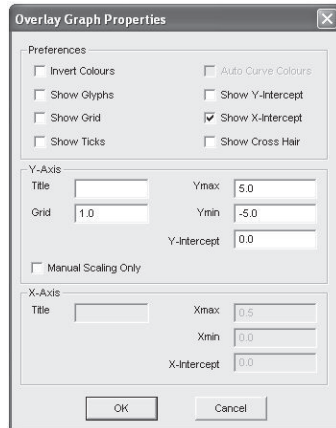
Overlay Graphs are the most common and familiar type of online plotting tool in PSCAD. These Graphs display EMTDC measured data as a function of time, where multiple Curves may be added (or overlaid on top of each other) onto a single Graph.

Adjusting Overlay Graph Properties

Left double-click on the desired Overlay Graph, or right-click on the graph and select **Graph Properties....**



This should bring up the Overlay Graph Properties dialog window as shown below:



There are various parameters that may be edited through this window, each of which are described below.

Preferences:

- **Invert Colours:** Select this option to give the Graph a black background (instead of white or yellow).
- **Show Glyphs:** Select this option to show glyph symbols on all curves in the Graph.
- **Show Grid:** Select this option to display grid lines for the x-axis and y-axis major grids.
- **Show Ticks:** Select this option to show major grid tick marks along the y-axis intercept line.
- **Auto Curve Colours:** Select this option to use automatic colouring of Curves in the Graph. You cannot change Curve colour manually when this option is selected.
- **Show Y-Intercept:** Select this option to display the y-intercept (horizontal) intercept line. The y-intercept line can be adjusted using the Y-Intercept field described below.
- **Show X-Intercept:** Select this option to display the x-intercept (vertical) intercept line. The x-intercept is always at time zero, and cannot be adjusted in Overlay Graphs.
- **Show Cross Hair:** Select this option to invoke the Cross Hairs Mode.

Y-Axis:

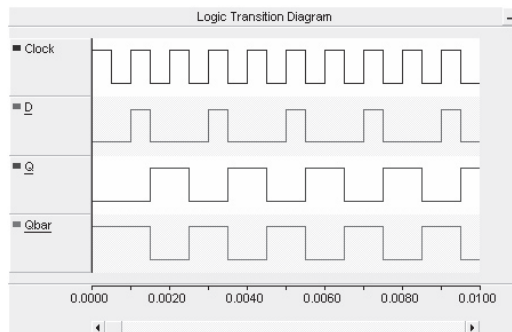
- **Title:** Enter text for display as the Graph title (located on the left side of the Graph).

Chapter 6: Online Plotting and Control

- **Grid:** Specifies the y-axis grid interval. To view the y-axis grid lines, select the option **Show Grid** described above.
- **Units:** Enter text for display beside the Graph **Title**. This text will be appended to the title text and appear within parentheses.
- **Ymin:** Specifies the minimum y-axis viewing limit on the Graph.
- **Ymax:** Specifies the maximum y-axis viewing limit on the Graph.
- **Y-Intercept:** Specifies the y-axis location of the y-intercept line. This line is only visible if **Show Y-Intercept** is selected (see above).
- **Manual Scaling Only:** Select this feature to lock the y-axis limits set in **Ymin** and **Ymax** above. The y-axis will remain locked during any subsequent Zoom operations.

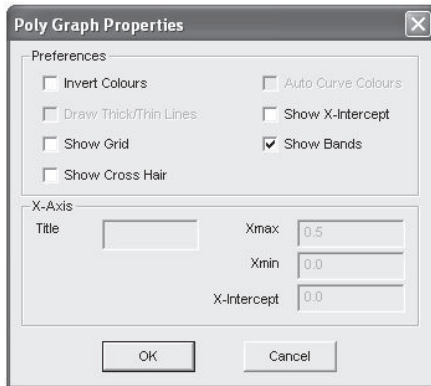
PolyGraphs

PolyGraphs are used specifically to display plotted curves in a 'stacked' format. That is, each Curve is contained within its own viewing space, and are stacked one atop the other. A PolyGraph may be chosen over a standard Overlay Graph if the user needs to either view many single-curve plots in a compact space, or to make use of the Curve Digital Style functions to create a logic transition diagram:



Adjusting PolyGraph Properties

Left double-click on the desired PolyGraph, or right-click on the Graph and select **Graph Properties....** This should bring up the PolyGraph Properties dialog:



There are various parameters that may be edited through this window, each of which are described below.

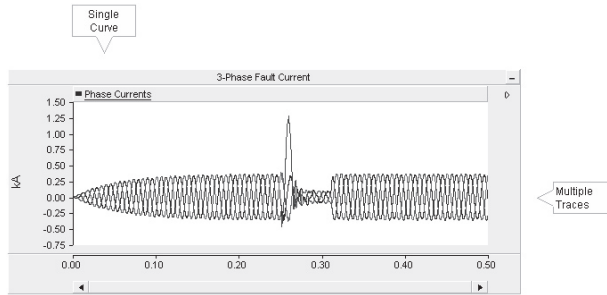
Preferences:

- **Invert Colours:** Select this option to give the Graph a black background (instead of white or yellow).
- **Show Grid:** Select this option to display grid lines for the x-axis and y-axis major grids.
- **Show Cross Hair:** Select this option to invoke the Cross Hairs Mode.
- **Auto Curve Colours:** Select this option to use automatic colouring of Curves in the Graph. You cannot change Curve colour manually when this option is selected.
- **Show X-Intercept:** Select this option to display the x-intercept (vertical) intercept line. The x-intercept is always at time zero, and cannot be adjusted in PolyGraphs.
- **Show Bands:** Selecting this option will give a different background colour between multiple Curves in one Graph, for easy visual differentiation.

CURVES

A curve is a special Runtime object best described as a graphical representation of a string of data points, where each point is associated with a simulation time step. Curves are created by linking to an Output Channel component, to which a scalar or array set of data signals have been input. As such, curves can be multi-dimensional; that is a single Curve may possess many sub-curves or Traces, where each Trace corresponds to a single array value.

Chapter 6: Online Plotting and Control



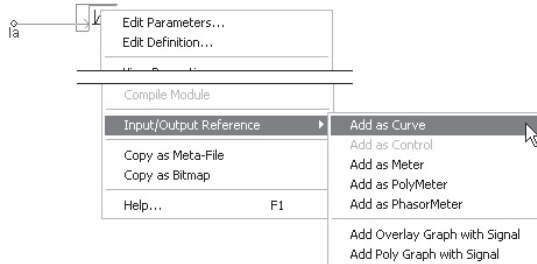
If the Output Channel input signal is a scalar (i.e. single dimension), then the Curve will consist of a single Trace. In previous versions of PSCAD, this was always the case.

NOTE: For more information on accessing and adjusting individual Trace properties, see 'Traces' in this chapter.

Adding a New Curve to a Graph

Adding a Curve to a Graph can be accomplished a couple of different ways:

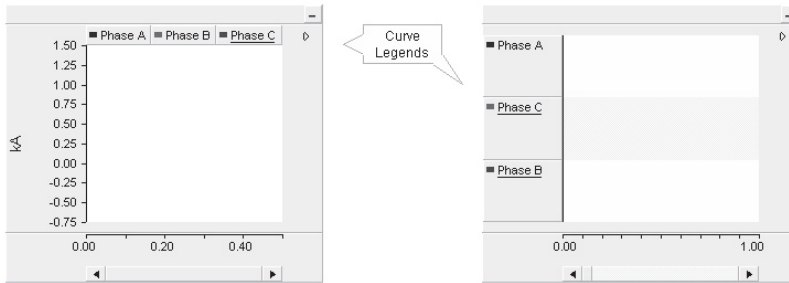
- 1. Drag and Drop:** Hold down the **Ctrl** key. **Left-click and hold** over the Output Channel component from which you would like to extract the Curve.. Drag the mouse pointer over a Graph and release the mouse button. See 'Drag and Drop' in Chapter 5 for more details on this.
- 2. Input/Output Reference:** Right-click on the Output Channel component from which you would like to extract the Curve. Select **Input/Output Reference | Add as Curve**. Select the desired Graph with a left-click, then right-click and select **Paste Curve**.



NOTE: If the simulation has been run already (with this Output Channel inserted), the corresponding Curve data will be displayed immediately in the Graph upon being pasted.

Curve Legends

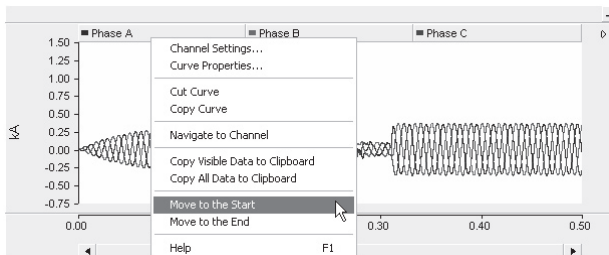
Once a Curve has been added to a Graph, the Curve Title will appear in the Curve Legend.



Curve Order

Once multiple Curves have been added to a particular Graph, you may change the order in which they appear. Ordering Curves can be accomplished in one of two ways:

1. **Drag and Drop:** Left-click and hold over the Curve in the Curve Legend. Drag the mouse pointer to a new position in the Curve Legend and release the mouse button. See 'Drag and Drop' in Chapter 5 for more details on this.
2. **Right-Click Menu:** Right-click over the corresponding Curve Legend and select one of the following from the pop-up menu: **Move to the Start** or **Move to the End**.



Chapter 6: Online Plotting and Control

Cut/Copy/Paste an Existing Curve

Right-click over the Curve Title in the Curve Legend and select either **Cut Curve** or **Copy Curve**, depending on what you want to do. Right-click over any Graph and select **Paste Curve** to add the copied Curve to the Graph. The Curve should then appear in the corresponding Curve Legend.

Copy Data to Clipboard

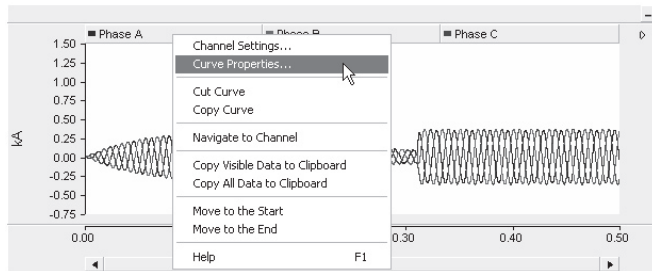
If a simulation has been run and a particular Curve contains data, you have the option of copying all or a portion of this single Curve data set to the clipboard. Right-click over the corresponding Curve name in the Curve Legend and select one of the following from the pop-up menu:

- **Copy Visible Data to Clipboard:** Copies all Curve data currently visible in the graph window.
- **Copy All Data to Clipboard:** Copies all Curve data available.

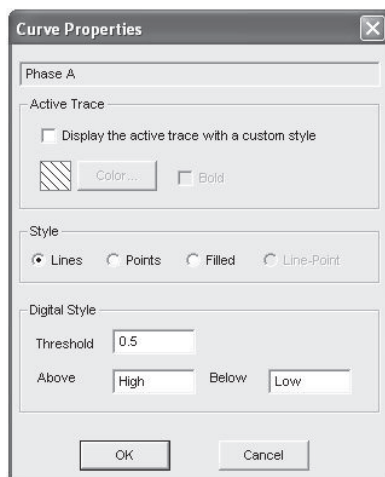
The data is copied as 'Comma Separated Variables' (*.csv) format for easy migration into common data analysis software.

Adjusting Curve Properties

Left double-click on the desired Curve in the Curve Legend, or right-click the curve and select **Curve Properties...**



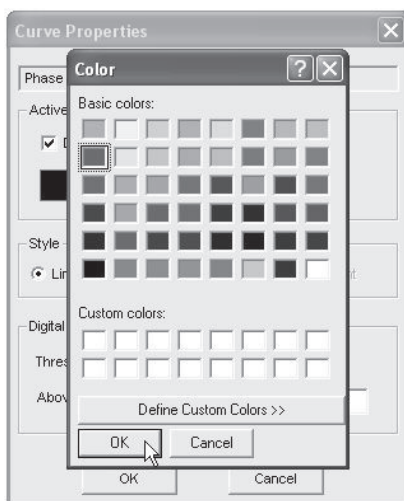
This should bring up the Curve Properties dialog window.



Active Trace

The following list describes the parameters in this section:

- **Display the active trace with a custom style:** Select this option if you wish to change the colour or width of the Active Trace.
- **Colour:** Press the **Colour...** button to select a display colour for the Trace. Press the **OK** button in the Colour dialog. This option is enabled only if 'Display the active trace with a custom style' is selected.



Chapter 6: Online Plotting and Control

- **Bold:** Select this option if you wish the Trace to appear bolded. This option is enabled only if 'Display active trace with a custom style' is selected.

Style

The following list describes the parameters in this section:

- **Lines:** Displays the Curve as a standard line.
- **Points:** Displays the Curve as a series of points according to the set plot step.
- **Filled:** Fills the area under the Curve (between the curve and the 0.0 line) with the curve colour.

Digital Style

These options are only considered if the Curve is part of a Poly Graph. Digital style controls the properties of the Curve Traces when they are in digital Mode. The following list describes the parameters in this section:

- **Threshold:** The threshold value at which to change the display state of the Curve.
- **Above / Below:** Enter the state for the Curve when value is above and below the set threshold respectively.

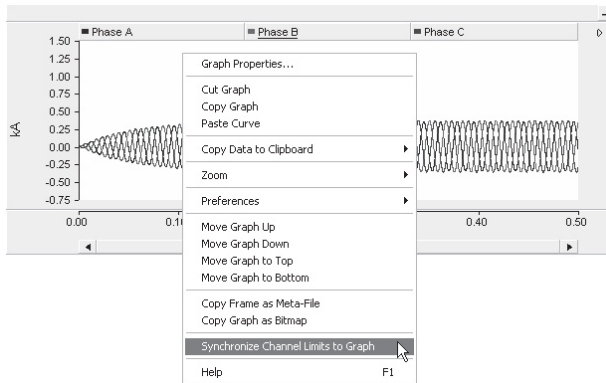
Adjusting Channel Settings

The Output Channel Properties dialog for a particular Curve can be accessed directly from the Curve Legend pop-up menu. **Right-click** the curve and select **Channel Settings...**

Synchronizing Output Channel Limits with Those of the Graph

Once one or more Curves have been added to an Overlay Graph, all corresponding Output Channel **Min / Max Limits** can be set equal to the Graph y-axis minimum and maximum limits (i.e. synchronized).

To synchronize Output Channel limits with the Graph, right-click over the Overlay Graph and select **Synchronize Channel Limits to Graph.**

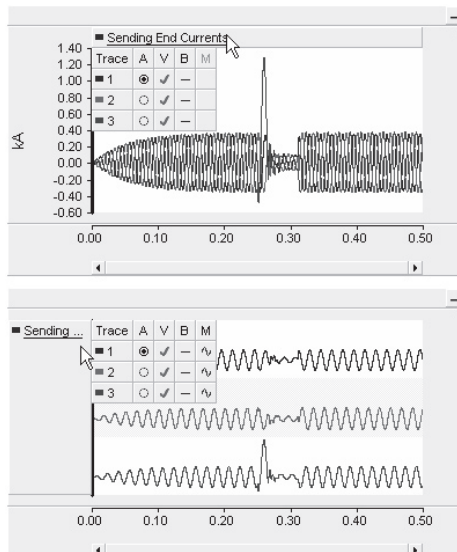


TRACES

Multiple signal Curves can be plotted online as a single entity, where each 'sub-curve' in a multiple signal Curve is referred to as a *Trace*. Each Trace in a Curve may be enabled or disabled separately (i.e. shown or hidden).

Trace Drop Down Menu

Trace properties and control can be accessed through a special drop down menu with a single **left-click** on the Curve title in the Curve Legend.



Chapter 6: Online Plotting and Control

Adjusting Trace Properties

Before adjusting any Trace properties, you must first invoke the Trace Drop Down Menu, as described above. The drop menu consists of four separate columns, which each allow for easy access to certain Trace properties. These columns are described below:

Trace	A	V	B	M
■ 1	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ 2	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ 3	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

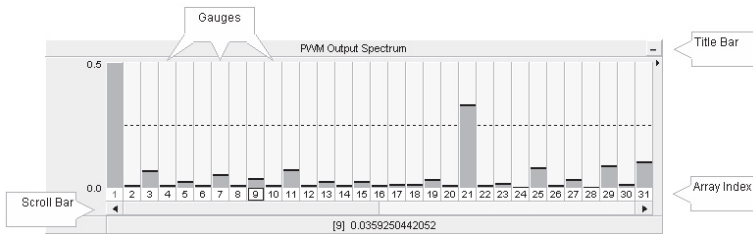
- **Trace:** Indicates the Trace number and corresponding colour setting. The number corresponds to the array index number of the multi-signal Curve. To change individual Trace colour, see 'Adjusting Curve Properties' in this chapter.
- **A:** The 'A' stands for 'Active.' Select the radio button in this column to select the Active Trace. The Active Trace will be the default focus when switching to Cross-Hair Mode. Also, only the properties of the Active Trace may be adjusted: See 'Adjusting Curve Properties' in this chapter.
- **V:** The 'V' stands for 'View.' Left-click the individual check boxes in this column to hide/view individual Traces. You can hide/view all Traces by left-clicking 'V' itself.
- **B:** The 'B' stands for 'Bold.' Left-click the individual check boxes in this column to bold/unbold individual Traces. You can bold/unbold all Traces by left-clicking 'B' itself.
- **M:** The 'M' stands for 'Mode.' This function is only valid when using PolyGraphs. Left-click the individual boxes in this column to change the Trace mode from digital to analog. When in digital mode, the Trace will be displayed in a special two-state format, where the state depends on whether it is above or below a preset Threshold value. For more details on setting Threshold and other digital properties see 'PolyGraphs' in this chapter.

POLYMETERS

A PolyMeter is a special Runtime object used specifically for monitoring a single, multiple-trace Curve. The PolyMeter dynamically displays the magnitude of each Trace in bar type format (called Gauges), which results in an overall appearance similar to a spectrum analyzer. The power of this device lies in its ability to compress a large amount of data into a small viewing area, which is particularly

PSCAD

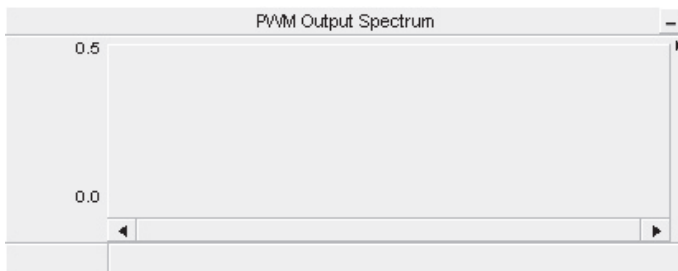
helpful when viewing harmonic spectrums, such as data output from the On-Line Frequency Scanner (FFT) component.



The Gauge width is fixed, and so if the PolyMeter is not wide enough to display all data, a simple horizontal scroll bar is provided. An Array Index display is included directly under the Gauges for easy identification.

PolyMeters are special objects that cannot be added directly from the toolbar. Each PolyMeter is directly linked with a single Curve from a single Output Channel component.

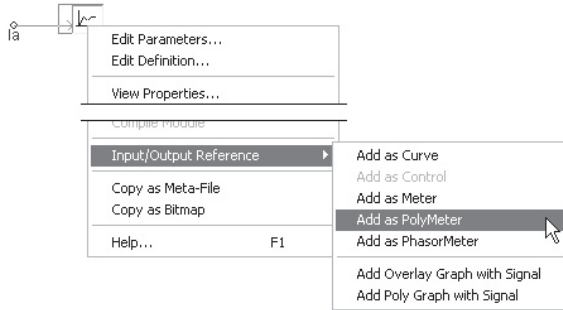
NOTE: PolyMeters will appear as a blank container (as shown below) until the Project is compiled and run.



Adding a PolyMeter

Right-click on an Output Channel component within Circuit view of the Design Editor and select **Input/Output Reference | Add as PolyMeter**.

Chapter 6: Online Plotting and Control

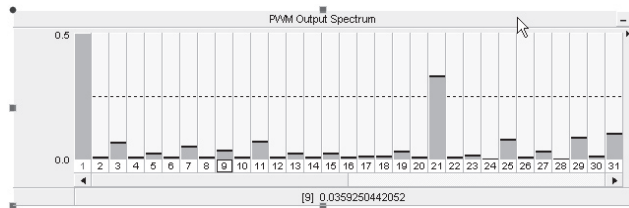


Right-click on a blank part of the Circuit canvas and select **Paste** to insert the PolyMeter.

Moving and Resizing a PolyMeter

To move a PolyMeter, move the mouse pointer over the title bar and then **left-click and hold**. Drag the meter to where it is to be placed and release the mouse button.

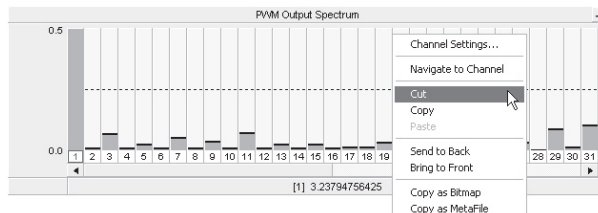
To resize, move the mouse pointer over the title bar and **left-click** to select the PolyMeter. Grips should then appear around the outer edge as shown below.



Move the mouse pointer over one of the grips. **Left-click and hold** and then drag the pointer to resize.

Cut/Copy PolyMeter

Right-click over the PolyMeter and select **Cut** or **Copy** respectively.



PSCAD

Once a PolyMeter has been cut or copied it may then be pasted to another location in the Project.

Paste PolyMeter

Cut or copy a PolyMeter as described above. Right-click over a blank area of the Project page in Circuit view and select **Paste**. A PolyMeter may be pasted multiple times.

Copy PolyMeter as Meta-File/Bitmap

The entire PolyMeter display can be copied to the Windows Clipboard in either Meta-File (*.wmf) or Bitmap (*.bmp) format. Right-click over the PolyMeter title bar and select **Copy as Bitmap** or **Copy as MetaFile**. Go to your report document and paste the image.

Navigate to Channel

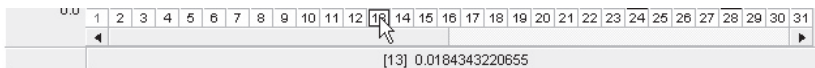
You can navigate directly to the associated Output Channel component by selecting this option. Right-click over the PolyMeter and select **Navigate to Channel**. PSCAD will automatically find the Output Channel and point to it with an arrow.

Adjusting Channel Settings

The y-axis properties of the PolyMeter are set in the corresponding Output Channel Properties dialog. This dialog may be accessed directly from the PolyMeter by right-clicking over the PolyMeter title bar and selecting **Channel Settings...**

Displaying Specific Data

The magnitude of individual array elements can be displayed in the status bar at the bottom of the PolyMeter. Simply left-click on a particular index number in the Array Index, to view the magnitude of that element.

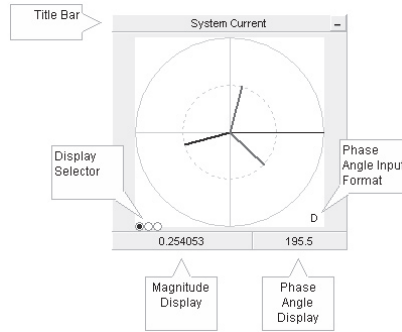


PHASORMETERS

A PhasorMeter is a special Runtime object that can be used to display up to six, separate phasor quantities. The PhasorMeter displays phasors in a polar graph, where the magnitude and phase of each phasor responds dynamically during a simulation run. This device

Chapter 6: Online Plotting and Control

is perfect for visually representing phasor quantities, such as output from the On-Line Frequency Scanner (FFT) component.



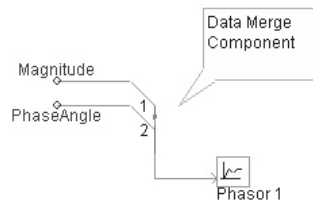
PhasorMeters are special objects that cannot be added directly from the toolbar. Each PhasorMeter is directly linked with a single Curve from a single Output Channel component. In the case of the PhasorMeter, a Curve with at least two Traces (magnitude and phase angle) is the minimum requirement for the device to work properly. The following section describes how to prepare data signals for display in the PhasorMeter.

Preparing Data for Display

A single phasor quantity, as it pertains to the PhasorMeter, is composed of a separate magnitude signal, along with an associated phase angle signal, which combine to represent a single phasor quantity in polar format. That is:

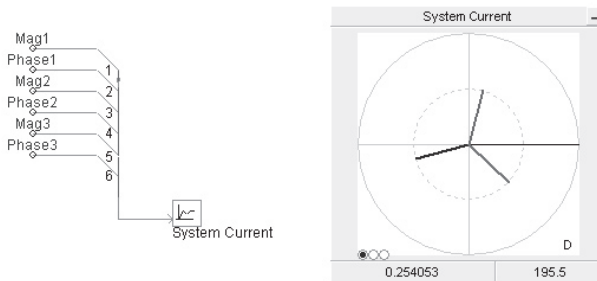
$$X \angle \phi$$

To display a single phasor quantity, the user must construct an array data signal of dimension two, using the Data Merge component. The two data signals will represent the magnitude and phase angle respectively. To this array signal, an Output Channel is attached as shown below:



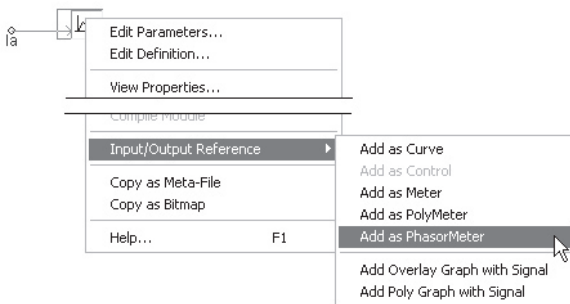
The PhasorMeter always assumes that the elements of the array are in the above displayed order. That is, element 1 is the magnitude and element 2 is the phase angle.

The PhasorMeter allows up to six phasors per display, which corresponds to an array signal of dimension 12. In cases of multiple phasors, the order of each magnitude/phase angle group must appear in the same order as the first (i.e. magnitude, phase angle, magnitude, phase angle, etc.) as shown below for a three phasor display:



Adding a PhasorMeter

Right-click on an Output Channel component within Circuit view of the Design Editor and select **Input/Output Reference | Add as PhasorMeter**.



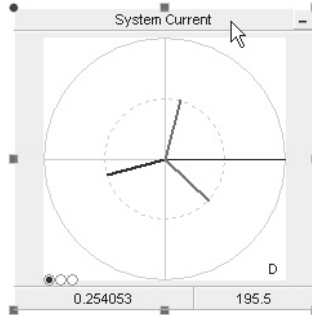
Right-click on a blank part of the Circuit canvas and select **Paste** to insert the PhasorMeter.

Moving and Resizing a PhasorMeter

To move a PhasorMeter, move the mouse pointer over the title bar and then **left-click and hold**. Drag the meter to where it is to be placed and release the mouse button.

Chapter 6: Online Plotting and Control

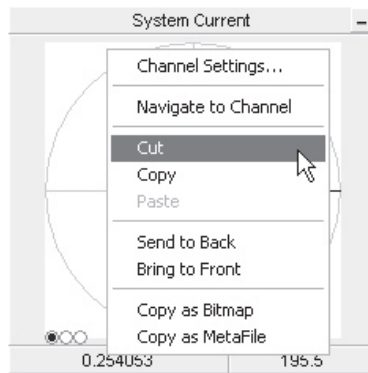
To resize, move the mouse pointer over the title bar and **left-click** to select the PhasorMeter. Grips should then appear around the outer edge as shown below.



Move the mouse pointer over one of the grips. **Left-click and hold** and then drag the pointer to resize.

Cut/Copy PhasorMeter

Right-click over the PhasorMeter and select **Cut** or **Copy** respectively.



Once a PhasorMeter has been cut or copied, it may then be pasted to another location in the Project.

Paste PhasorMeter

Cut or copy a PhasorMeter as described above. Right-click over a blank area of the Project page in Circuit view and select **Paste**. A PhasorMeter may be pasted multiple times.

PSCAD

Copy PhasorMeter as Meta-File/Bitmap

The entire PhasorMeter display can be copied to the Windows Clipboard in either Meta-File (*.wmf) or Bitmap (*.bmp) format. Right-click over the PhasorMeter title bar and select **Copy as Bitmap** or **Copy as MetaFile**. Go to your report document and paste the image.

Navigate to Channel

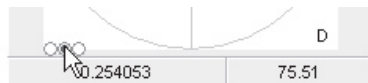
You can navigate directly to the associated Output Channel component by selecting this option. Right-click over the PhasorMeter and select **Navigate to Channel**. PSCAD will automatically find the Output Channel and point to it with an arrow.

Adjusting Channel Settings

The polar axis limits of the PhasorMeter are set in the corresponding Output Channel Properties dialog. This dialog may be accessed directly from the PhasorMeter by right-clicking over the device title bar and selecting **Channel Settings....**

Displaying Specific Data

The magnitude and phase angle of individual array elements can be displayed in the status bar at the bottom of the PhasorMeter. Simply left-click on the Display Selector over the proper button. The buttons represent phasors 1 through 6 from left to right.



Adjusting Phase Angle Input Format

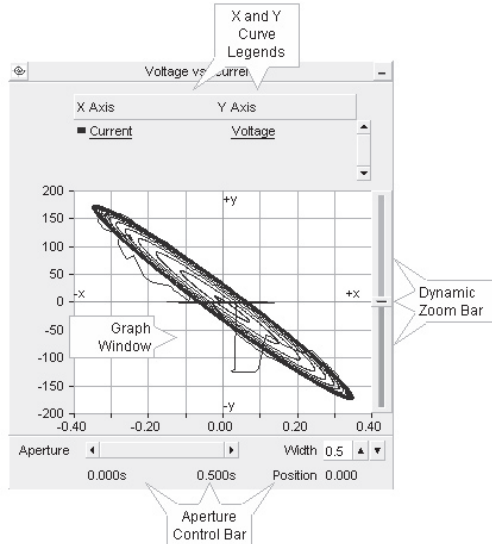
The format that the incoming phase angle data is in must be specified. This is easily accomplished by toggling the 'D' or 'R' display in the bottom-right corner of the graph, where:

- **D:** Degrees
- **R:** Radians



XY PLOTS

The XY Plot is a special type of Runtime object, which is composed of both a Graph Frame and a single, specialized Graph window for the purpose of plotting one Curve versus another. An XY Plot can accommodate multiple Curves on each of the X and Y-axes, and includes a Dynamic Zoom and Polar Grid features.



Although the XY Plot is used to plot one signal versus another in PSCAD, each of these signals is based on the same time scale. It is therefore possible to scroll through the data in the time domain. The XY Plot includes a time domain Aperture Control Bar, located at the bottom of the plot frame.

Adding an XY Plot

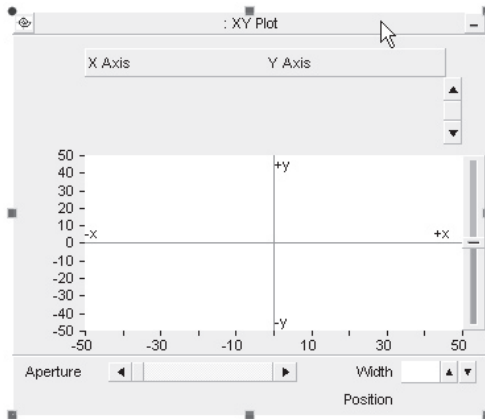
Open the Project in the Circuit window of the Design Editor. Right-click on a blank portion of the page and select **Add Component | XY Plot**, or press the **XY Plot** button in the Control Palette bar.



Moving and Resizing an XY Plot Frame

To move an XY Plot, place the mouse pointer over the title bar and then **left-click and hold**. Drag the XY Plot frame to where it is to be placed and release the mouse button.

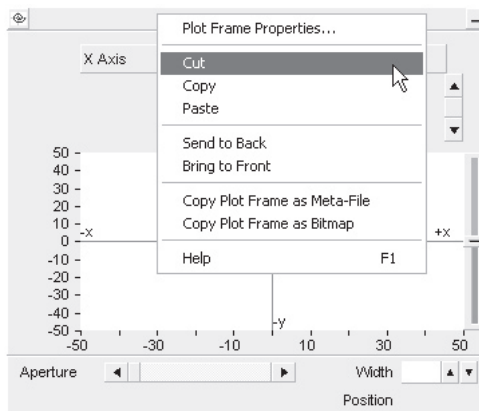
To resize, place the mouse pointer over the title bar and **left-click** to select the XY Plot frame. Grips should then appear around the outer edge as shown below.



Move the mouse pointer over one of the grips. **Left-click and hold** and then drag the pointer to resize.

Cut/Copy XY Plot Frame

Right-click over the XY Plot title bar and select **Cut** or **Copy** respectively.



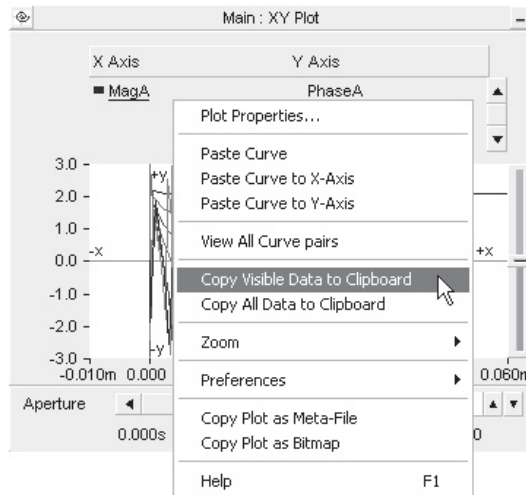
Chapter 6: Online Plotting and Control

Paste XY Plot Frame

Cut or copy an XY Plot, as described above. Right-click over a blank area of the Project page and select **Paste**. An XY Plot may be pasted multiple times.

Copy Data to Clipboard

If a simulation has been run and your XY Plot contains Curve data, you have the option of copying all or a portion of this data to the clipboard.



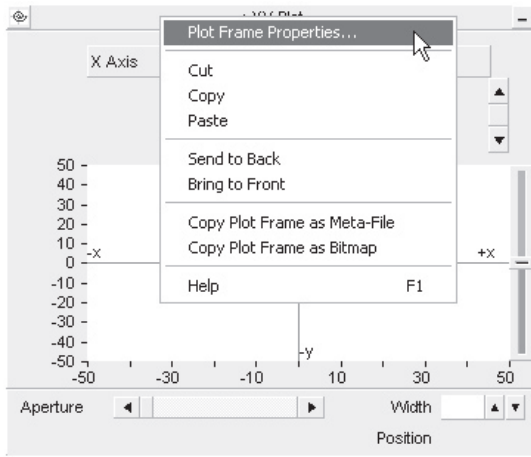
The data is copied as 'Comma Separated Variables' (*.csv) format for easy migration into common data analysis software.

Two choices are given:

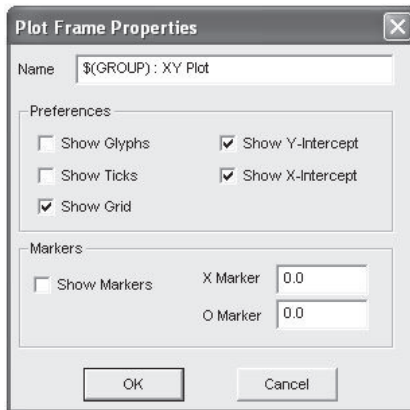
- **All Data:** Copies all Curve data available.
- **Visible Data:** Copies all Curve data visible in the XY Plot Graph window.

Adjusting XY Plot Frame Properties

To access the Plot Frame Properties dialog, left double-click the plot title bar, or right-click over the title bar and select **Plot Frame Properties....**



This should bring up the Plot Frame Properties dialog window.



There are various parameters that may be edited through this window, each of which are described below.

- **Name:** Enter a title for the XY Plot (this text will appear in the Graph Frame title bar). The default text may appear a bit cryptic: This syntax is used as a naming convention for grouping objects in the Workspace. For more information on this syntax, see 'Grouping of Runtime Objects' in this chapter.

Chapter 6: Online Plotting and Control

Preferences:

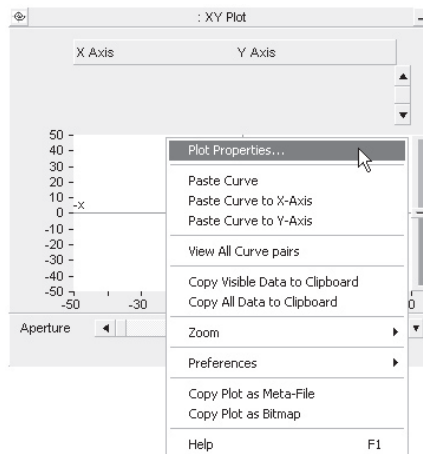
- **Show Glyphs:** Select this option to show glyph symbols on all curves in the graph.
- **Show Ticks:** Select this option to show major grid tick marks along the x and y-axis intercept lines.
- **Show Grid:** Select this option to display grid lines for the x-axis and y-axis major grids.
- **Show Y-Intercept:** Select this option to display the y-intercept (horizontal) intercept line. The y-intercept is always at zero, and cannot be adjusted.
- **Show X-Intercept:** Select this option to display the x-intercept (vertical) intercept line. The x-intercept is always at zero, and cannot be adjusted.

Markers:

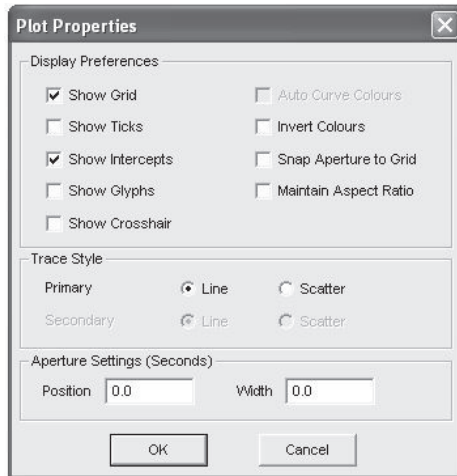
- **Show Markers:** Select this option to show the X and O markers.
- **X Marker:** Enter the position (in seconds) to place X marker.
- **O Marker:** Enter the position (in seconds) to place O marker.

Adjusting Plot Properties

Left double-click over the plot area (white part), or right-click over the plot area and select **Plot Properties...**



This should bring up the Plot Properties dialog window.



There are various parameters that may be edited through this window, each of which are described below.

Display Preferences:

- **Show Grid:** Select this option to display grid lines for the x-axis and y-axis major grids.
- **Show Ticks:** Select this option to show major grid tick marks along the x and y-axis intercept lines.
- **Show Intercepts:** Select this option to display both intercept lines (horizontal and vertical). The intercepts are always at zero, and cannot be adjusted.
- **Show Glyphs:** Select this option to show glyph symbols on all curves in the graph.
- **Show Cross Hair:** Select this option to invoke the Cross Hairs mode.
- **Auto Curve Colours:** Select this option to use automatic colouring of curves in the graph. You cannot change curve colour manually when this option is selected.
- **Invert Colours:** Select this option to give the graph a black background (instead of white or yellow).
- **Snap Aperture to Grid:** Select this feature so that when using dynamic aperture adjustment, the aperture view will snap to the major grid while zooming.

Chapter 6: Online Plotting and Control

- **Maintain Aspect Ratio:** Select this option in order to maintain the aspect ratio of the plotted curve (in both the x and y directions) whenever the plot frame is resized. If this option is disabled, the plotted curve will stretch or compress according to the actual shape of the plot frame.

Trace Style:

- **Primary:** Select whether to draw traces as Line or Scatter view. Scatter view simply adds a single dot for each X-Y coordinate.

Aperture Settings (seconds):

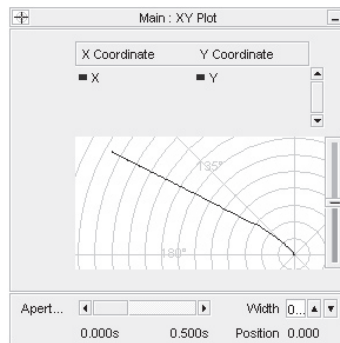
- **Position:** Enter the starting position in seconds of the aperture window.
- **Width:** Enter the width in seconds of the aperture window.

Polar Grid

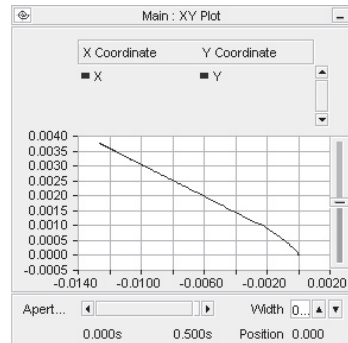
The default cartesian (XY) grid display can be switched to a polar grid by simply toggling the XY/Polar display button at the top-left of the Graph Frame.



The polar and XY grids are simply overlaid on top of the plotted data:



Polar Grid



Cartesian Grid

PSCAD

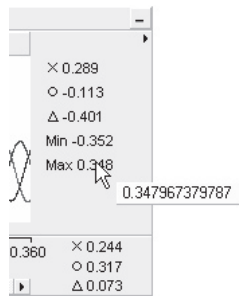
Dynamic Zoom

See 'Dynamic Zoom in XY Plots' in this chapter for more details.

TOOL TIPS IN PLOTS

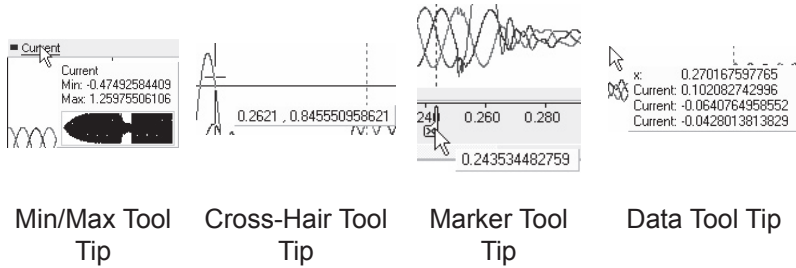
In order for plotting tools to be useful in the analysis of data, it is important that quantities be displayed with an adequate amount of precision. This presents a problem however, as the more precision that is displayed graphically on the plots, the more space (or graphical real estate) is used, leaving less space in the plotting environment. This problem is overcome in PSCAD, through the use of pop-up tool tips.

By default, the most important quantities involved in plotting are displayed directly on the Graph Frame. This is especially apparent when viewing Marker data, which is displayed to 4 significant digits. For most studies, 4 significant digits is not enough to provide the user with accurate assessments of the data. Tool tips however will provide the true quantities up to 12 significant digits. Simply move the mouse pointer over the data displayed.



Existing Tool Tips, such as Cross-Hair and Min/Max curve Tool Tips are also standardized to 12 significant digits. The following images show the locations of other Tool Tips in the plotting environment:

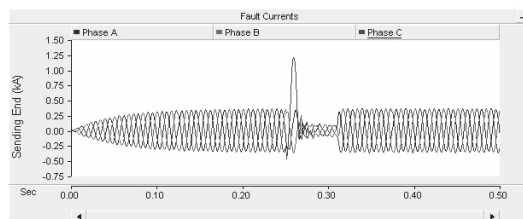
Chapter 6: Online Plotting and Control



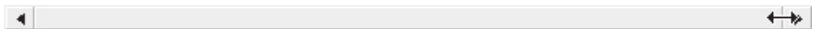
DYNAMIC APERTURE ADJUSTMENT

The Dynamic Aperture Adjustment feature is available on both the Graph Frame and the XY Plot Runtime objects and allows the user to define a fixed time based display window (or aperture), and then dynamically slide this aperture through the entire time scale. The aperture size itself can be re-adjusted at any time.

Although the following example uses a Graph Frame, it may be easily applied to the XY Plot, as well. Run the simulation so that your Graph displays the Curve data. The following image shows a simulation that has been run for 0.5 s.



As can be seen from the Graph, the fault occurs around 0.25 s and lasts for about 0.07 s. The Dynamic Aperture Adjustment feature can be used to close the viewing window to a smaller time width, so that the fault waveforms can be more easily studied. To do this, move your mouse pointer over the horizontal scroll bar at the far right so that the mouse pointer turns into a double-headed arrow.

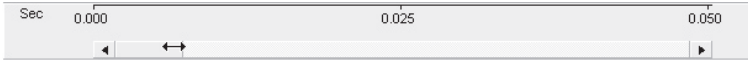


Click and hold the left mouse button and drag the scroll bar aperture slowly to the left.

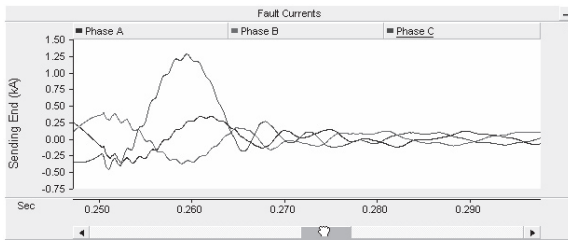


PSCAD

You should see your graph display dynamically adjusting as you change the size of the scroll bar aperture. Watching the horizontal axis display, shrink the aperture to an appropriate size (for the graph above, about 0.05 s or so).



Release the left mouse button and move the pointer over the scroll bar aperture, click and hold the left mouse button again – the mouse pointer icon should become a hand. Drag the mouse so as to scroll across the time frame of the graph.



If desired, you can shrink the aperture further for even more detail. An aperture will also be created automatically when you zoom into a certain data range.

You may also use either the left and right arrow buttons at each end of the scroll bar, or the arrow keys on your keyboard to scroll through data: To scroll in small increments, simply left-click the either arrow, for larger increments hold down the **Ctrl** key and then left-click. When using the arrow keys, make sure that the focus is on the Graph Frame being viewed, as these keys can also function as a scroll mechanism for the Circuit page.

NOTE: This exercise is for the purpose of describing Dynamic Aperture Adjustment. A more efficient way to zoom into this aperture window is to use Horizontal Zoom.

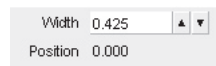
Adjusting and Controlling the Aperture in XY Plots

As mentioned previously, the user is able to set and control the time viewing window for the XY Plot. At the bottom of the XY Plot frame, there is an area devoted solely for the adjustment and control of this viewing window - otherwise known as the aperture.

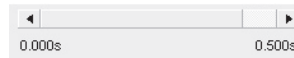
Chapter 6: Online Plotting and Control



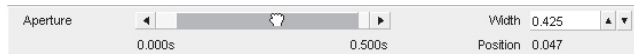
This area contains two sections: The Dynamic Aperture Adjustment scroll bar on the left, and the Manual Aperture Adjustment field on the right. You can increment the aperture width manually by pressing the up/down arrows to the right of the Manual Aperture Adjustment field.



If the aperture **Width** is a fraction of the total time scale for the plot, then the aperture indicator scroll bar will reflect this by showing a smaller aperture window:



You may now move the aperture window along the time axis, maintaining its set width. To do this, move your mouse pointer over the aperture indicator in the scroll bar and left-click and hold. Your mouse pointer should become a hand symbol. Move the mouse left or right along the axis.



The aperture **Position** indicator at the bottom-right corner will indicate the starting time of the aperture.

MARKERS

Markers are a special feature included in both Graph Frames and XY Plots to help users with the analysis of their online data. Specifically, they are used to delineate the data so as to focus analysis to that specific range. Depending on Marker position, convenient legend displays will indicate the difference between the two Markers in both the x and y directions.

PSCAD

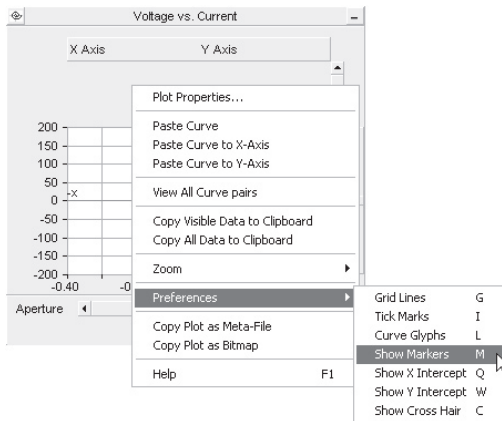
Markers are used only on the x-axis (time axis) and will appear as two adjustable tabs. The Marker tabs are labelled as **X** and **O** and the combination of the two set the specified boundaries. Once Markers are set, analysis can be performed on the data contained within them.

Markers are used in a slightly different manner between Graph Frames and XY Plots. Any differences are noted in the sections below.

Show/Hide Markers

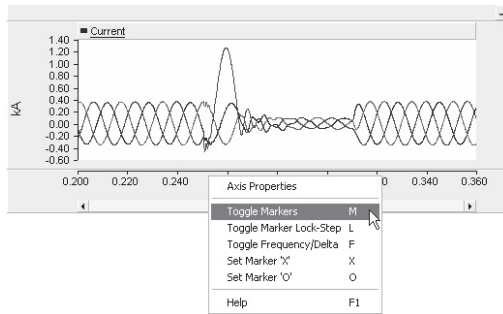
There are a few ways to show or hide Markers:

- Select the desired Graph Frame or XY Plot with a left-click on the graph display area. Right-click over the graph to generate a pop-up menu and select **Preferences | Show Markers**.



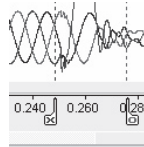
- Left double-click the Graph Frame horizontal axis, or **right-click** over the horizontal axis and select **Axis Properties...** to bring up the **Axis Properties** dialog. Select the **Show Markers** selection box.

Chapter 6: Online Plotting and Control



- Left double-click an Overlay or Poly Graph, or right-click over the Graph and select **Graph Properties...** to bring up the Graph Properties dialog. Select the **Show Markers** selection box.

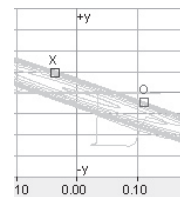
If you are working with a Graph Frame, two tabs should appear along the horizontal axis. Each marker tab, labelled **X** and **O**, correspond to an x-axis (time) position.



If you are working with an XY Plot, an extra display bar will appear at the bottom of the plot frame as shown below:



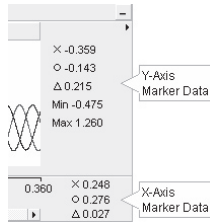
Aperture Control Bar



Marker Display in Graph

Graph Frame Marker Legends

Once Markers are enabled (shown) in a Graph Frame, legends will appear on the right hand side of the frame itself.



Each Graph in the Graph Frame will have a legend directly to the right of the Graph. The values displayed are specific to each Graph and are described as follows:

- **X:** Displays the y-axis value at the **X** Marker position for the Active Trace.
- **O:** Displays the y-axis value at the **O** marker position for the Active Trace.
- **Δ:** Displays the difference between the above two values.
- **Min:** Displays the minimum value of the Active Trace within the Marker boundaries.
- **Max:** Displays the maximum value of the Active Trace within the Marker boundaries.

The x-axis will also have its own legend where the values displayed are similar to those described above, but for the x-axis.

XY Plot Marker Legends

Once Markers are enabled (shown) in an XY Plot, a Marker bar will appear at the bottom of the Plot Frame. As soon as either of these Markers is moved, marker legends will appear directly to the right of each:



The values displayed are described as follows, specific to each Marker:

- **x:** Displays the Marker x-axis value for the active Trace.
- **y:** Displays the Marker y-axis value for the active Trace.
- **T:** Displays the Marker time-axis value for the active Trace.

Chapter 6: Online Plotting and Control

Changing the Active Curve

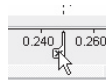
The Markers will only monitor the Active Trace of one Curve at a time. If multiple Curves exist in an Overlay or Poly Graph, you can scroll through them by simply pressing the **Space Bar** on your keyboard. You may also conveniently select the desired Curve in the Graph legend.

When using XY Plots, you must select the desired Curve in the graph legend, as the **Space Bar** function is not available.

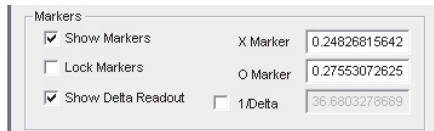
Adjusting Marker Positions

Once Markers are enabled, there are a few ways to adjust the Marker positions:

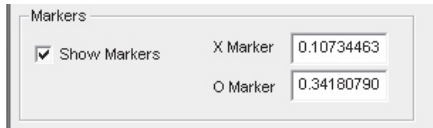
- To manually adjust, place the mouse pointer over the marker tab in either the Graph frame or XY Plot, **left-click and hold** and drag it left or right. Release the left mouse button when the Marker is in the desired position.



- When using Graph Frames, left double-click the Graph Frame horizontal axis, or right-click over the horizontal axis and select **Axis Properties...** to bring up the Axis Properties dialog. Select the **Show Markers** selection box and then set the x-axis values of each Marker directly in the input fields provided. Note that it is also possible to directly set the frequency (i.e. **1/Delta**) between the two Markers.

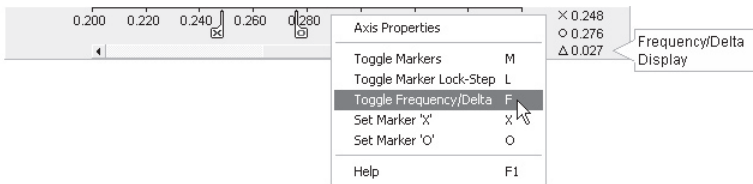


- When using XY plots, **right-click** over the plot title bar and select **Plot Frame Properties...** to bring up the Plot Frame Properties dialog. Select the **Show Markers** selection box and then set the time-axis values of each marker directly in the input fields provided.



Toggle Time Difference Frequency/Delta

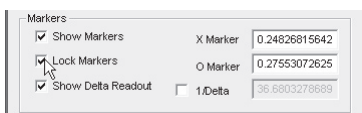
Once the Markers are enabled, you can conveniently invert the time axis difference between the two Markers to display corresponding frequency. Left double-click the Graph Frame horizontal axis, or right-click over the horizontal axis and select **Toggle Frequency/Delta** (or press the **F** key on your keyboard):



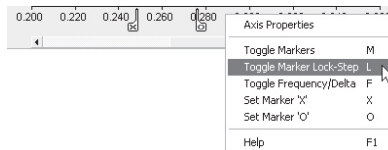
This feature is not available in XY Plots.

Locking/Unlocking Markers

Locking will force the two Markers to maintain a constant distance between each other when they are moved along the x-axis. Left double-click the Graph Frame horizontal Axis to bring up the Axis Properties dialog and select **Lock Markers**, or right-click over the horizontal axis and select **Toggle Marker Lock-Step** (or press the **L** key on your keyboard) to lock the Markers. Perform the same operation again to unlock.



Axis Properties Dialog



Right-Click Menu

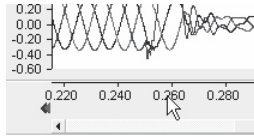
Setting Markers

Markers can be set to a certain position on the Graph Frame time axis as follows:

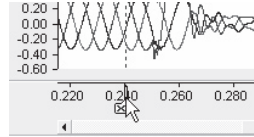
1. Show the Markers by pressing the **M** key.

Chapter 6: Online Plotting and Control

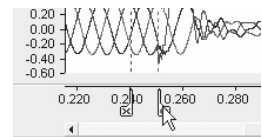
2. Left-click on the axis at the approximate position where you wish to place the **X** Marker. Press the **X** key on your keyboard.
3. Repeat this procedure to set the **O** Marker, except press the **O** key.



Press the M Key

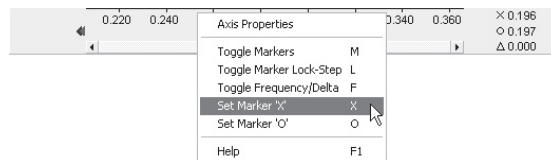


Press the X Key



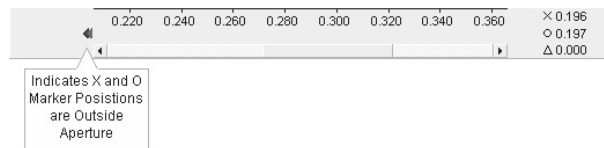
Press the O Key

Instead of pressing the above keys, you can also use the time axis pop-up menu. Left-click on the Graph Frame time axis at the approximate position where you wish to place a Marker, then select the appropriate menu function.



Using Markers as Bookmarks

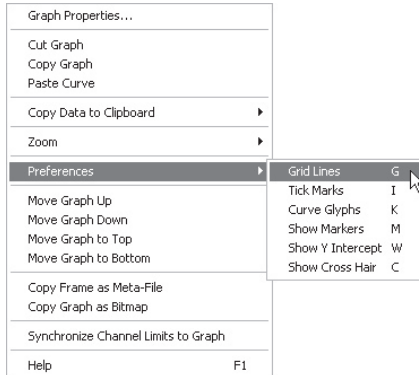
If only a fraction of the total Graph Frame time scale is currently being viewed in a Graph, the Markers may appear outside the viewing frame when they are first enabled. If this happens, blue and red arrow symbols will appear on the time axis indicating the markers are set off the screen.



The current viewing window can be moved directly to the position of either Marker. Simply left-click either the blue or the red arrow (blue for the **X** Marker and red for the **O** Marker). This functionality, along with setting the Markers, can be used to 'bookmark' viewing positions on the Graph.

PREFERENCES

Plotting preferences can be adjusted from either the Graph or XY Plot pop-up menus (depending on which one you are using). Simply right click over a Graph or XY Plot to bring up the corresponding pop-up menu. Then, select **Preferences**.



The functions available in this menu are described below:

- **Grid Lines:** Select this preference to toggle grid line display on the selected graph. You can also use the **G** keyboard shortcut.
- **Tick Marks:** Select this preference to show major grid tick marks along the y-intercept. You can also use the **I** keyboard shortcut.
- **Curve Glyphs:** Select this preference to show glyphs on all curves in the graph. You can also use the **K** keyboard shortcut.
- **Show Markers:** Select this preference to show the markers. You can also use the **M** keyboard shortcut.
- **Show Y Intercept:** Select this preference to display the y-intercept (horizontal) intercept line. The y-intercept line can be adjusted using the **Y-Intercept** field in the Graph Properties dialog. On XY plots, the y-intercept is always along the x-axis. You can also use the **W** keyboard shortcut.
- **Show Cross Hair:** Select this preference to invoke the Cross Hairs mode.

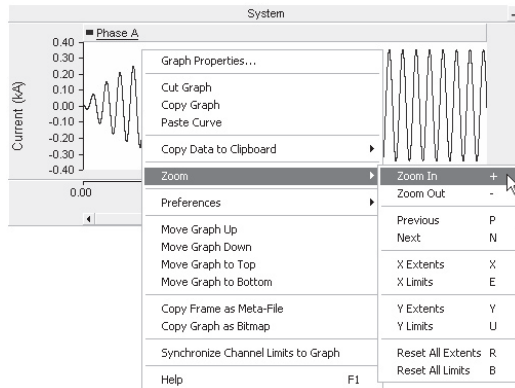
Many of the above preferences can be set within the various dialog windows involved with Graphs and XY Plots.

ZOOM FEATURES

Once a simulation has been run and EMTDC output data has been collected, there are several ways to zoom in and out of the data displayed. The following are some of the more common methods.

General Zoom In and Zoom Out

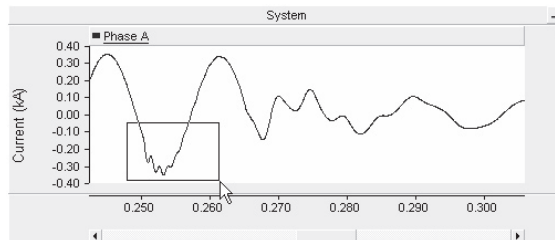
Select the desired Graph or XY Plot with a left-click on the graph display area. Right-click over the graph to generate a pop-up menu and select **Zoom | Zoom In** or **Zoom | Zoom Out**.



As an alternative, you can also use the **+** or **-** keyboard shortcuts for Zoom In or Zoom Out respectively (once the desired graph has been selected).

Box Zoom

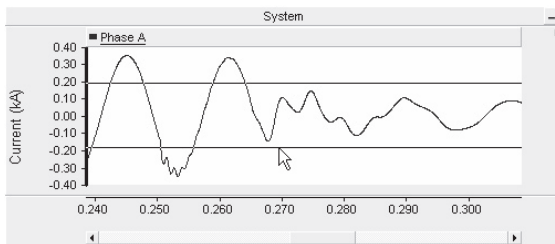
Select the desired Graph or XY Plot with a left-click on the Graph display area. Click and hold the left mouse button and drag the mouse pointer to create a boxed region. Release the left mouse button to zoom to that region.



NOTE: You can also zoom along a single axis by drawing out a narrow box. The zoom will ignore the narrow side of the box.

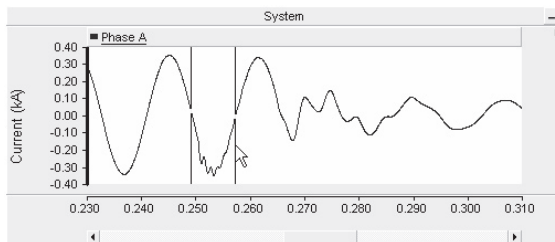
Vertical Zoom

Select the desired Graph or XY Plot with a left-click on the Graph display area. Press **Shift + left mouse hold** and drag the mouse pointer in a vertical direction (i.e. up or down along the y-axis) to create a vertical zoom region. Release the left mouse button to zoom to that region.



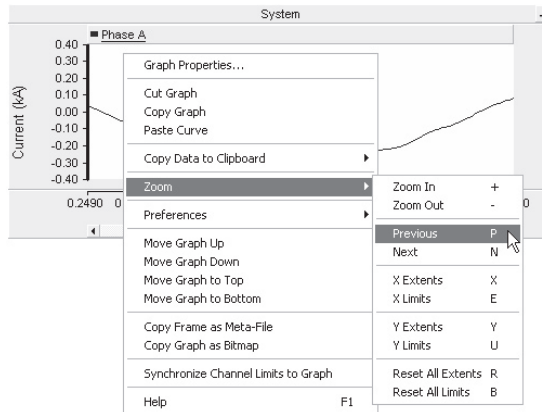
Horizontal Zoom

Select the desired Graph or XY Plot with a left-click on the graph display area. Press **Ctrl + left mouse hold** and drag the mouse pointer in a horizontal direction (i.e. left or right along the x-axis) to create a horizontal zoom region. Release the left mouse button to zoom to that region.



Zoom Previous/Next

Select the desired Graph or XY Plot with a left-click on the graph display area. Right-click over the graph to generate a pop-up menu and select **Zoom | Previous** or **Zoom | Next**.



As an alternative, you can also use the **P** or **N** keyboard shortcuts for Zoom Previous or Zoom Next respectively (once the desired graph has been selected).

Zoom Extents

The Zoom Extents feature allows the user to zoom to the extents of the plotted data. The data 'extents' refers to the absolute maximum and minimum values that exist during the entire simulation run, in either the X or Y direction.

Select the desired Graph or XY Plot with a left-click on the graph display area. Right-click over the graph to generate a pop-up menu and select either **Zoom | X Extents** or **Zoom | Y Extents**.

As an alternative, you can also use the **X** or **Y** keyboard shortcuts for Zoom X Extents or Zoom Y Extents respectively (once the desired graph has been selected).

Zoom Limits

The Zoom Limits feature allows the user to zoom to predefined limits in either the X or Y direction. The y-axis limits are set based on the **Default Display Limits** inputs in the corresponding Output Channels for the Curves. In the event of multiple Curves, the limits are based on the largest and smallest Default Display Limits among all relevant Output Channels. The x-axis limits are based on the desired duration of the simulation.

PSCAD

Select the desired Graph or XY Plot with a left-click on the graph display area. Right-click over the graph to generate a pop-up menu and select either **Zoom | X Limits** or **Zoom | Y Limits**.

As an alternative, you can also use the **E** or **U** keyboard shortcuts for Zoom X Limits or Zoom Y Limits respectively (once the desired graph has been selected).

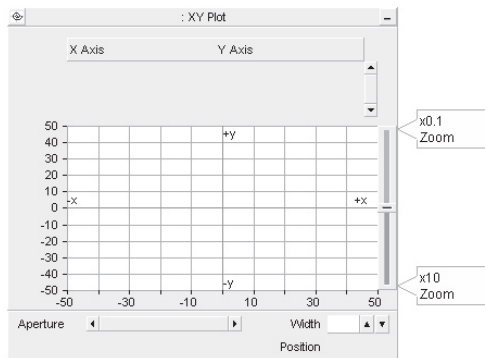
Resetting All Extents and Limits

Select the desired graph or XY plot with a left-click on the graph display area. Right-click over the graph to generate a pop-up menu and select either **Zoom | Reset All Extents** or **Zoom | Reset All Limits**.

As an alternative, you can also use the **R** or **B** keyboard shortcuts for Reset All Extents or Reset All Limits respectively (once the desired graph has been selected).

Dynamic Zoom in XY Plots

Dynamic zoom is a feature specific to the XY Plots. **Left click and hold** with your mouse pointer over the dynamic zoom control bar as shown below:



With a left click, your mouse pointer should become an open hand. Hold down the left mouse button and move the hand down to zoom in, or up to zoom out.

CROSS HAIR MODE

Once a simulation has been run and output data has been plotted, you can trace through Curve values using the Cross Hairs. Right-

Chapter 6: Online Plotting and Control

click over the Graph or XY Plot to generate a pop-up menu and then select **Preferences | Show Cross Hair**. You can also select the Graph or XY Plot and press the **C** key on your keyboard.

Once invoked, Cross-Hair Mode will remain on until it is turned off. While in Cross-Hair Mode, the mouse pointer becomes a 'cross-hair' while over top of the Graph:



Once Cross Hair Mode is enabled, move the mouse pointer over the Graph and **left-click and hold**. Drag the mouse along the Graph. If multiple Curves exist in the Graph, then you can move the Cross Hair from Curve to Curve by simply pressing the **Space Bar** on your keyboard.

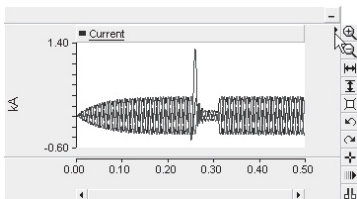


As shown above, the Curve XY data will be displayed beside the cross hair. When your left mouse button is released, the Cross Hairs will disappear, but Cross Hair Mode will remain invoked until you press the **C** key again (or select **Preferences | Show Cross Hair**).

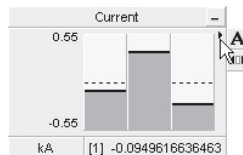
In XY Plots, the Cross Hair will follow the mouse pointer only.

POP-UP TOOL BARS

Pop-up toolbars have been added to Graphs and PolyMeters in order to provide a quick and easy means for accessing the most common functions. Pop-up toolbars may be invoked by left-clicking on the small arrow in the top-right corner of the Graph or PolyMeter windows:













Graph





PolyMeter

The pop-up toolbar functions are described as follows:

Graphs:

Button	Description
	Zoom In
	Zoom Out
	Zoom horizontal extents
	Zoom vertical extents
	Reset all extents
	Previous zoom
	Next zoom
	Toggle Cross-Hair Mode
	Toggle Auto-Pan of X-Axis
	Toggle Markers

PolyMeters:

Button	Description
	Toggle index labels
	Toggle scroll view mode

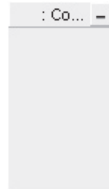
ONLINE CONTROLS AND METERS

Special objects in PSCAD allow the user online access to EMTDC input variables, so that these variables (and hence the simulation results) can be altered during the course of the simulation run. These objects and how to use them are described in the following sections.

Chapter 6: Online Plotting and Control

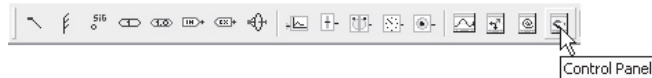
Control Panels

A Control Panel is a special component used for accommodating Control or Meter Interfaces and can be placed anywhere in a project page. Once a Control Panel has been added, you may then proceed to add as many Control or Meter Interfaces to it as you wish.



Adding a Control Panel

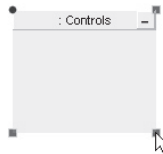
Open the Project in the Circuit window of the Design Editor. Right-click on a blank portion of the page and select **Add Component | Control Panel**, or press the **Control Panel** button in the Control Palette bar.



Moving and Resizing a Control Panel

To move a Control Panel, move the mouse pointer over the title bar and then **left-click and hold**. Drag the frame to where it is to be placed and release the mouse button.

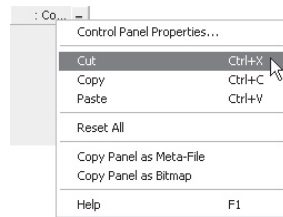
To resize, move the mouse pointer over the title bar and **left-click** to select it the Control Panel. Grips should then appear around the outer edge as shown below.



Move the mouse pointer over one of the Grips. **Left-click and hold** and then drag the pointer to resize.

Cut/Copy Panel

Right-click over the Control Panel title bar and select **Cut** or **Copy** respectively. You can also use the standard **Ctrl + X** or **Ctrl + C**.



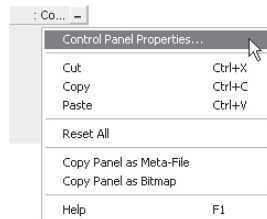
Once a Control Panel has been cut or copied, it may then be pasted into another location in the Project (along with its contents).

Paste Panel

Cut or copy a Control Panel as described above. Right-click over a blank area of the project page and select **Paste**. A Control Panel may be pasted multiple times.

Adjusting Panel properties

To access the Control Panel Properties dialog, left double-click the Control Panel title bar, or right-click over the title bar and select **Control Panel Properties...**



Presently the only adjustable panel property is the Caption:

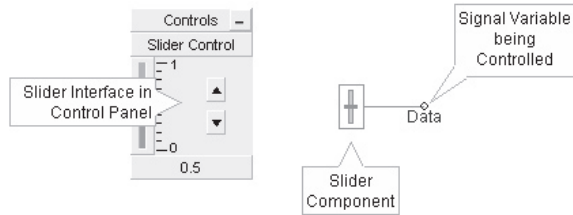
- **Caption:** Enter a title for the Control Panel (this text will appear in the Control Panel title bar). The default text may appear a bit cryptic: This syntax is used as a naming convention for grouping objects in the Workspace. For more information on this syntax, see 'Grouping of Runtime Objects' in this chapter.

Control Interfaces

A Control Interface is just as its name describes: A user interface object, which allows manual adjustment of an EMTDC data input variable by the user. A Control Interface must first be linked with one of the I/O Devices available in the Master Library (i.e. Variable Real/Integer Input Slider, Two State Switch, Rotary Switch (Dial),

Chapter 6: Online Plotting and Control

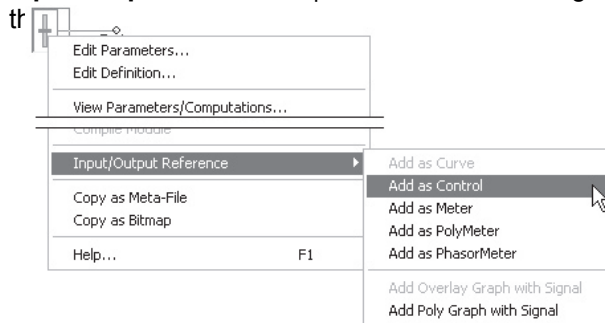
and Push Button). The Control Interface will then control the output of the linked control component. For example, the following image shows a Slider component linked to a control interface in a control panel.



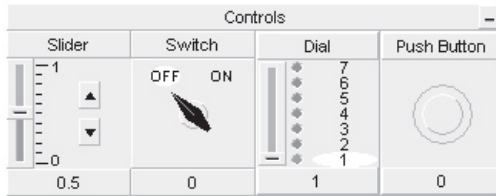
Adding a Control Interface to a Control Panel

There are two methods to accomplish this:

- **Drag and Drop:** Hold down the **Ctrl** key, **left-click and hold** the associated control component and drag and release over the Control Panel. See 'Drag and Drop' in Chapter 5 for more details.
- **Right-click** on the associated control component and select **Input/Output Reference | Add as Control**. Right-click over the



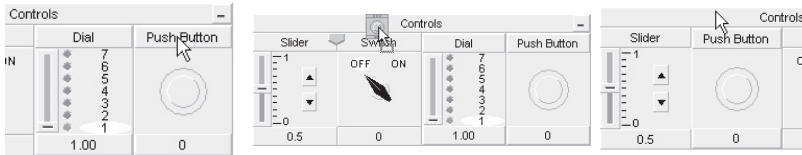
Of course, each type of control component will have a different control interface when added to a control panel. The following figure shows the available control components and their corresponding control interfaces:



Control Interface Order

Once multiple Control Interfaces have been added to a particular Control Panel, you may change the order in which they appear. This is accomplished either through the Drag and Drop feature, or the right-click menu:

- Drag and Drop:** **Left-click and hold** over the title bar of a particular Control Interface and then **drag** the mouse pointer over the Control Panel title bar. A downward arrow will appear between existing interfaces depending on the position of the mouse pointer. **Release** the left button to drop the interface.



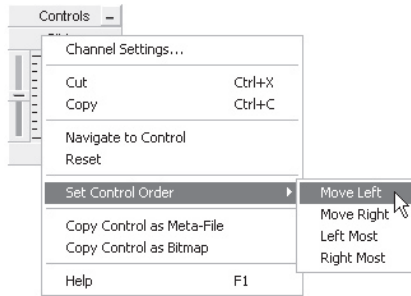
Left-Click and Hold

Drag

Release Button

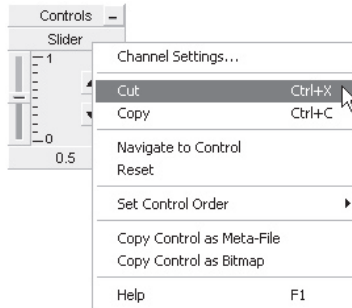
- Right-click over the Control Interface and select **Set Control Order**. Choose one of the options given.

Chapter 6: Online Plotting and Control



Cut/Copy Control Interface

Right-click over the Control Interface title bar and select **Cut** or **Copy** respectively.



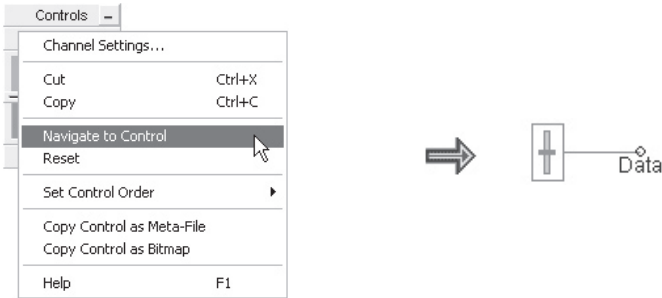
Once a Control Interface has been cut or copied, it may then be pasted into control panel in the project.

Paste Control Interface

Cut or copy a Control Interface as described above. Right-click over a Control Panel title bar and select **Paste**. A Control Interface may be pasted multiple times.

Navigate to Control

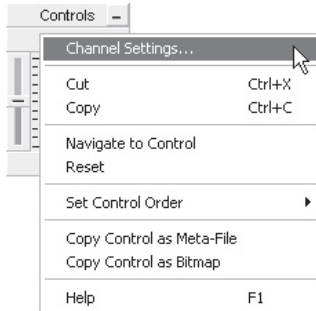
You can navigate directly to the I/O Control component associated with a particular Control Interface, by selecting this option. Right-click over a Control Interface and select **Navigate to Control**. PSCAD will automatically find the Output Channel and point to it with an arrow.



Select **Navigate to Control** PSCAD Points to Object

Adjusting Control Interface Properties

Control Interface properties and the corresponding control component properties are one and the same. Therefore, you can either adjust these properties through the control component directly, or through the Control Interface as follows: Left double-click the Control Interface title bar, or right-click over the Control Interface title bar and select **Channel Settings...**



Using the Control Interfaces

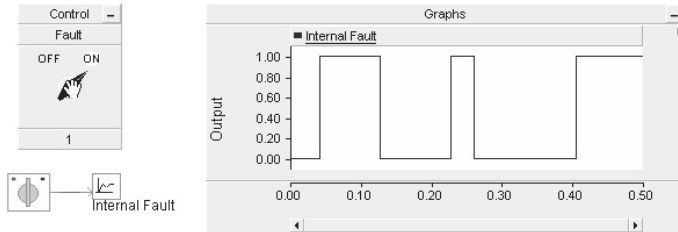
Once a particular Control Interface has been added to a Control Panel, the user may operate manually, either before or during a simulation run. Of course, each Control Interface (and control component) will perform a different operation on its corresponding output signal.

Switch and Push Button Components:

To operate either the Switch or Push Button components, simply left-click over top of the Control Interface itself. For example, each time the Switch Interface is clicked, it will toggle the corresponding Switch component between its two output states.

Chapter 6: Online Plotting and Control

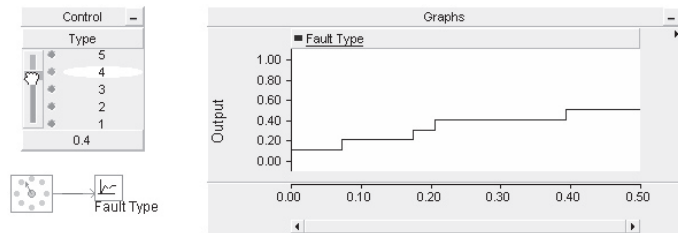
The image below shows a plot of a Switch component output, where the user has changed its state (i.e. turned it on and off) during a 1.0 second simulation run.



Dial Component:

To operate the Dial component, left-click and hold on the slider control knob and move the knob up or down. As the control knob is moved, the Dial interface will indicate graphically, which of the multiple states is currently being output from the corresponding Dial component.

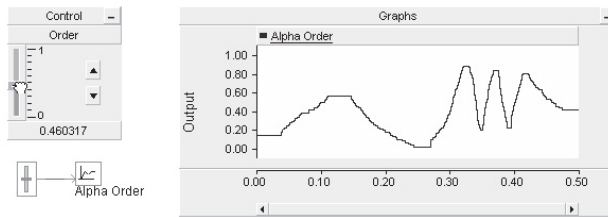
The image below shows a plot of a Dial component output, where the user has changed its state during a 1.0 second simulation run.



Slider Component:

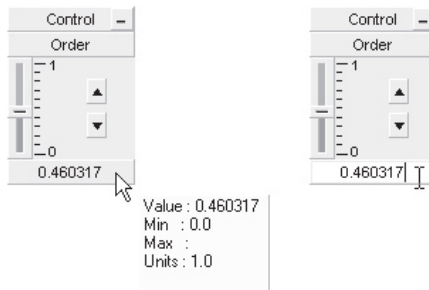
The Slider component is operated in the same manner as the Dial interface described above. The only difference is that the Slider output does not operate in discrete states, but provides a continuously variable output.

The image below shows a plot of a Slider component output, where the user has varied its output during a 1.0 second simulation run.



It is important to note that the Slider component will output exactly what is shown on the Slider interface. The Slider interface has a maximum precision of 6 significant digits. If more precision is required, you can add successive Slider outputs together, each handling different data ranges. Once you have fine-tuned the Slider output, you can replace it with a Real Constant component, which is capable of higher precision.

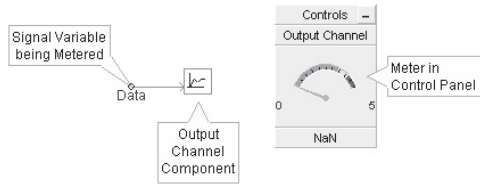
The Slider interface also allows you to enter an exact value (up to 6 significant digits) directly into its display window. Simply **left double-click** on the display window and then enter the data. Press the **Enter** button to exit and save the entered data, or press **Esc** to exit without saving changes.



Meters

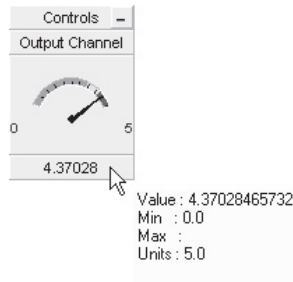
A meter is similar to a Graph in that it is used for displaying an EMTDC output signal and is linked to a corresponding Output Channel. Instead of displaying the signal as a Curve (as a graph would), the signal is used to operate a realistic meter display, with a pointer position proportional to the signal magnitude. For example, the following image shows an Output Channel component linked to a meter in a control panel.

Chapter 6: Online Plotting and Control



Meters can exist only within a control panel.

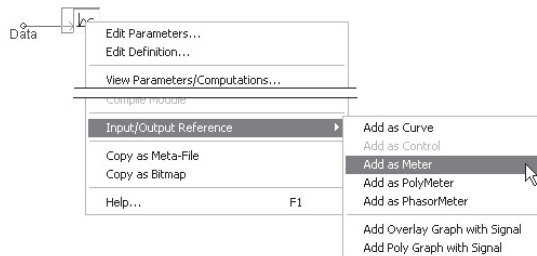
Like a Control Interface, Meter Interfaces display a maximum of 6 significant digits. However, twelve significant digits can be viewed via the Tool Tip.



Adding a Meter to a Control Panel

There are a couple of methods to accomplish this:

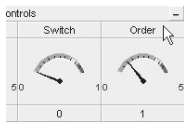
- **Drag and Drop:** Hold down the **Ctrl** key, **left-click and hold** the associated Output Channel component and drag and release over the Control Panel. See 'Drag and Drop' in Chapter 5 for more details.
- **Right-click** on the associated Output Channel component and select **Input/Output Reference | Add as Meter**. Right-click on the desired control panel title bar and select **Paste**.



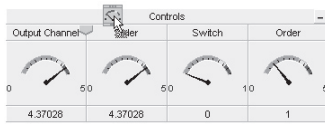
Meter Order

Once multiple Meter Interfaces have been added to a particular Control Panel, you may change the order in which they appear. This is accomplished either through the Drag and Drop feature, or the right-click menu:

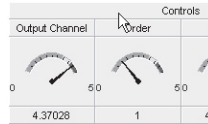
- **Drag and Drop:** **Left-click and hold** over the title bar of a particular Meter Interface and then **drag** the mouse pointer over the Control Panel title bar. A downward arrow will appear between existing interfaces depending on the position of the mouse pointer. **Release** the left button to drop the interface.



Left-Click and Hold

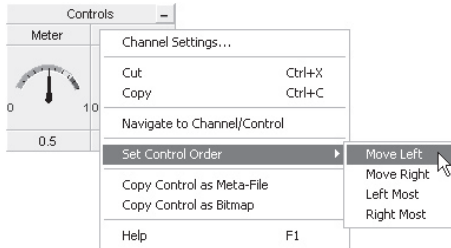


Drag



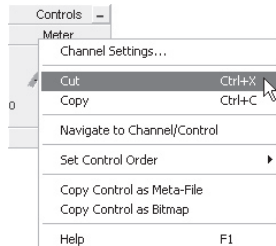
Release Button

- Right-click over the Meter Interface and select **Set Control Order**. Choose one of the options given.



Cut/Copy Meter

Right-click over the meter title bar and select **Cut** or **Copy** respectively.



Chapter 6: Online Plotting and Control

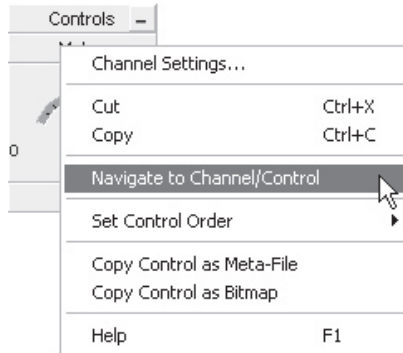
Once a Meter has been cut or copied, it may then be pasted into any control panel in the project.

Paste Meter

Cut or copy a Meter as described above. Right-click over a Control Panel title bar and select **Paste**. A Meter may be pasted multiple times.

Navigate to Channel/Control

You can navigate directly to the Output Channel or I/O Control component associated with a particular Meter Interface, by selecting this option. Right-click over the Meter Interface and select **Navigate to Channel/Control**. PSCAD will automatically find the Output Channel or I/O Control component and point to it with an arrow.



Select Navigate to Channel/Control

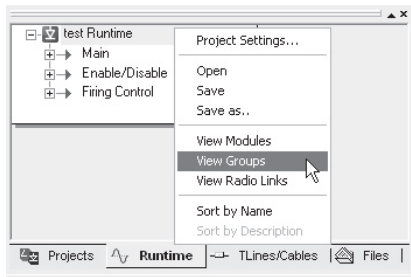
PSCAD Points to Object

Adjusting Meter Properties

Meter properties and the corresponding Output Channel properties are one and the same. Therefore, you can either adjust these properties through the Output Channel directly, or through the meter as follows: Left double-click the Meter title bar, or right-click over the title bar and select **Channel Settings...**

GROUPING OF RUNTIME OBJECTS

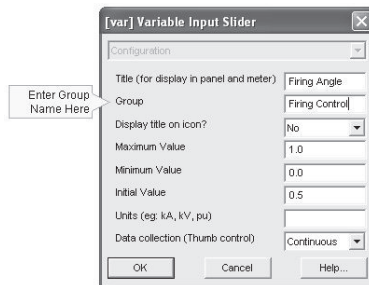
It is possible to organize both Control (i.e. Slider, Dial, Switch or Push Button) and Output Channel Runtime objects according to a Group Name. Once Group Names have been specified, objects may be viewed according to their Group Name in the Workspace Runtime Section.



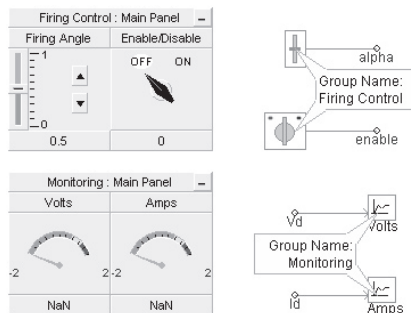
Grouped Runtime Objects may be Accessed in View Groups Mode

Creating a Runtime Object Group

Creating a Runtime Group is straightforward. All that is required is that a Group Name be added to the object input Parameter called **Group**. The image below shows the **Group** parameter in the Slider parameters dialog.



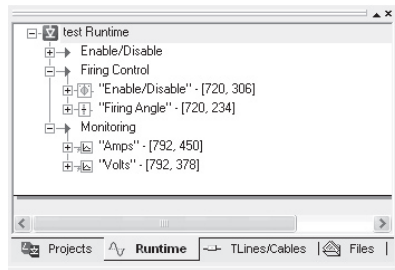
For example, consider the following Project page, where the user wishes to group existing Control objects according to function:



Both the 'Firing Angle' Slider and 'Enable/Disable' Switch controls are grouped under the name **Firing Control**, whereas the two Output Channels 'Volts' and 'Amps' are grouped under the name

Chapter 6: Online Plotting and Control

Monitoring. These objects are automatically organized by PSCAD for viewing in the Workspace Runtime Section.



Runtime Section in View Groups Mode

Viewing Runtime Object Groups

Runtime Objects may be viewed according to Group Name in the Runtime section of the Workspace (in View Groups Mode). For more details, see 'The Runtime Section: Groups View' in Chapter 4.

Displaying Group Name on Graph Frames and Control Panels

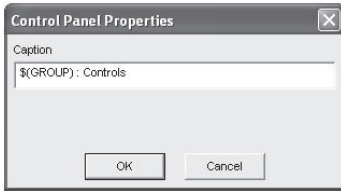
The Group Name of any Control or Meter Interface in a Control Panel, or any Curve in a Graph can be automatically displayed through the use of a special syntax in the title bar:

$\$(GROUP)$

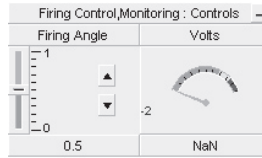
Wherever this character string is encountered within either a Graph Frame or a Control Panel **Caption** input, PSCAD will substitute the Group Name(s). If the Control Panel or Graph Frame contains objects with different Group Names, these names will be substituted as a comma separated list.

EXAMPLE 6-1:

A user wants to display the Group Name of all Interfaces within a Control Panel. The default **Caption** parameter in the Control Panel Properties dialog is $\$(GROUP) : Controls$. Two interfaces are then inserted into the Control Panel, one called 'Firing Angle' Group Name 'Firing Control,' and the other named 'Volts' with Group Name 'Monitoring.' The resulting Caption will appear as illustrated below:



Control Panel Properties

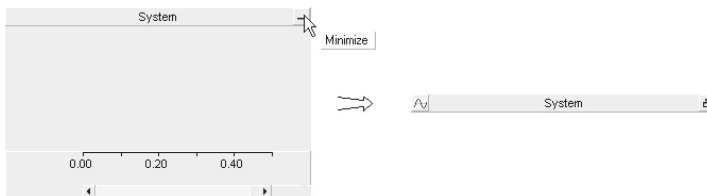


Control Panel with Objects from Two Different Groups

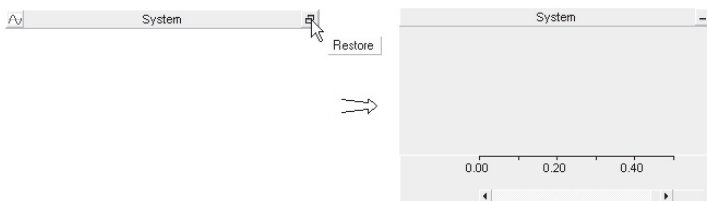
FRAME/PANEL MINIMIZATION

All Graph Frames, Control Panels, Polymeters and XY Plot Frames can be minimized in order to reduce clutter and save space when viewing data. This feature is similar to the PSCAD V3 feature 'Iconify Plot,' however additional functionality has been added to make the minimization process more intuitive.

Minimizing a Frame/Panel is accomplished simply by clicking the **Minimize** button in the top right-hand corner:








The Frame/Panel can then be restored by clicking the **Restore** button in the right-hand corner:



Chapter 6: Online Plotting and Control

Icons appear at the left of the minimized objects so that the different types of the Frame/Panels can be identified. A summary of these icons is given below:

-  Graph Frame
-  Control Panel
-  PolyMeter
-  PhasorMeter
-  XY Plot

Frames/Panels may also be minimized directly from the Workspace. See 'Runtime Objects' in Chapter 4 for more details.

Moving and Organizing Minimized Frames/Panels

Once a Frame/Panel has been minimized, the minimized object can be moved in a similar fashion as a restored Frame/Panel. Simply **left-click and hold** the minimized object and drag it to wherever it needs to be placed on the canvas.

The power in using minimized Frame/Panels for organization is that PSCAD remembers the last position of the object before it is minimized/restored. Therefore, it is possible to 'stack' the minimized objects to optimize space, yet when any of the objects are restored, they will appear in the same position as they were before being minimized. The reverse is true, as well.

DISPLAYING PLOTS AND CONTROLS IN REPORTS

Any plotting or control object (i.e. Graph Frames, Graphs, XY Plots, Control Panels, Meters, etc.) may be copied to the Windows clipboard for inclusion in reports or other documents. Options are provided to store the objects as either a picture in bitmap format (*.bmp), or Windows Meta-File format (*.wmf). Note that meta-file format is limited to the current screen size. Use bitmaps for larger graphs.

Setting Panel Style

You can change the panel style of all Graph Frames, XY Plots and Control Panels in the Workspace Settings dialog, before including them in reports. To access the Workspace Settings dialog, go to

PSCAD

Edit | Workspace Settings... in the Main Menu bar. See 'Views' in Chapter 3 for more details.

Copy as Meta-File or Bitmap

Right-click over the plotting or control object and select **Copy <object> as Meta-File** or **Copy <object> as Bitmap** from the pop-up menu. Depending on the object selected, the <object> part of the command will be replaced by the object type.

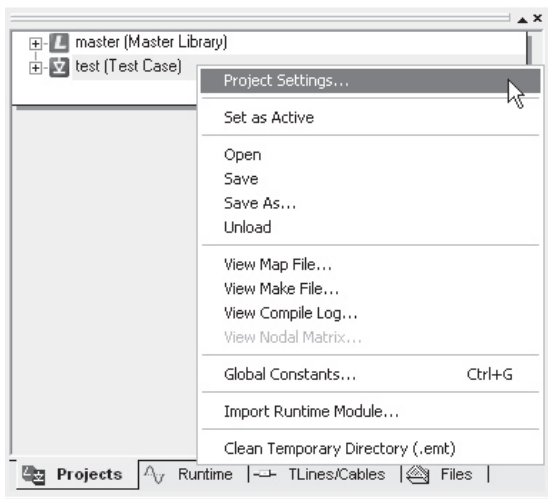
You may then go directly to a report document and paste the image, while it is still in the Windows clipboard.

Project Settings

Project Settings Overview

Most features and settings related directly to the control of the actual simulation in PSCAD are contained in the Project Settings dialog. Important parameters, such as total simulation time and time step, are included here along with more advanced, Project specific PSCAD and EMTDC features and processes.

This dialog can be accessed through a right-click on the project title itself (either a case or a library) in the Workspace window, and selecting **Project Settings...** from the pop-up menu.



The features and parameters available in the Project Settings dialog control window are divided into seven specific areas:

- General
- Runtime
- Network
- Dynamics
- Mapping
- Fortran
- Link

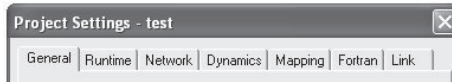
Chapter 7: Project Settings

As mentioned above, there are a number of more advanced features available for the enhancement of speed, accuracy and efficiency of simulations performed in PSCAD. The Project Settings dialog gives the user access and control of most of these features. Many users need not concern themselves with these and should leave most Project Settings to their default. However, there may be some instances where more advanced users would wish to disable and enable some processes at their discretion.

NOTE: The Project Settings window for a Library Project contains only a few of the parameters that appear for a Case Project, as simulations cannot be performed on a Library Project.

GENERAL

The properties contained within the General section of the Project Settings dialog control window are related to Project naming and version tracking. These are explained in detail in the sections below.



Description

This field allows for the entry of a single-line description of the Project. This description will be displayed beside the Project filename in the Workspace Viewer. Do not use quotes (") and apostrophes (') in this field.



File Path

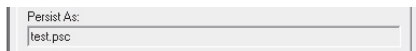
File Path displays the path and filename information for the Project. This field is for display only and cannot be changed through the Project Settings dialog.



PSCAD

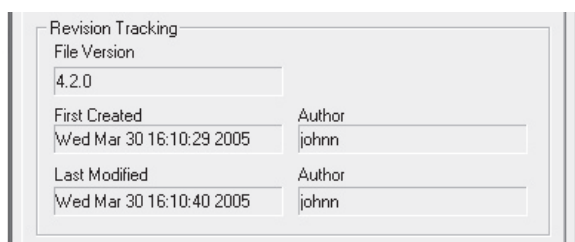
Persist As

Persist As displays the Project filename. This field is for display only and cannot be changed through the Project Settings dialog.



Revision Tracking

These settings display particular information regarding the Project revision history.



File Version

This is the PSCAD version used to create the project.

First Created

This is the date and time at which the project was first created.

Last Modified

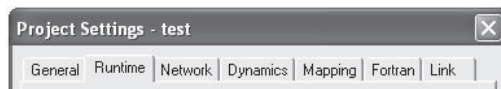
This is the date and time at which the project was last modified.

Author

This is the name (userid) of the person who created and/or modified the project.

RUNTIME

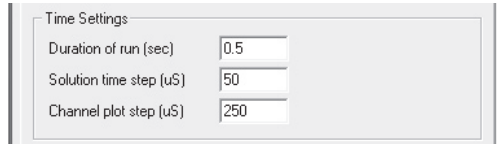
The properties contained within the Runtime section of the Project Settings dialog control window are the most commonly accessed project parameters. These are explained in detail in the sections below.



Chapter 7: Project Settings

Time Settings

These settings are very important and are used quite often in every day simulation studies.



Time Settings	
Duration of run (sec)	0.5
Solution time step (uS)	50
Channel plot step (uS)	250

Duration of Run (sec)

This is the total length of the simulation run, entered in seconds. If you start from time zero, this is the finish time of the run. If you start from a snapshot file (pre-initialized state), this is the length of run from the snapshot time.

Solution Time Step (us)

This is the EMTDC simulation time step, entered in microseconds. The default is 50 μ s, which is an optimum step for most practical circuits. However, users should make sure that the time step selected is suitable for their simulation. This input sets the value of the EMTDC internal variable DELT.

Channel Plot Step (us)

This is the time interval at which EMTDC sends data to PSCAD for plotting, as well as writing data to Output Files. It is always a multiple of the EMTDC simulation time step. Usually a 250 μ s plot step provides a reasonable resolution and speed.

Lower sampling intervals (higher sampling rate) can considerably slow down the simulation speed due to an excessive transfer of data from EMTDC to PSCAD (without adding much to the plot resolution). Users can experiment with this number for a given Project. If the sampling interval is too large, the waveforms may appear 'choppy.' If you are debugging the case, it is a good practice to plot every point in the simulation; that is, plot sampling time is equal to EMTDC simulation time step.

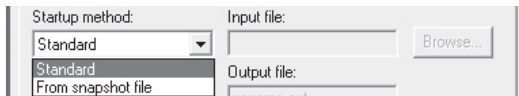
A trap that even the most experienced engineers can readily fall into is the setting of plot step too broad with respect to the level and period of noise in the signal. If a signal is periodic at a frequency similar to the plot step interval, the perceived output may be quite different to the actual signal. As a basic rule, if you are puzzled by the results

observed from a plotted simulation output, run the case with the plot step equal to the EMTDC time step and compare the results.

NOTE: The plot step can be modified during a run (or after starting from a snapshot). You can change the value from the Project Settings dialog or from the Settings bar.

Start-up Method

There are two ways to start a simulation in PSCAD: The standard method (i.e. from time = 0.0 seconds) and from a Snapshot File.



Startup method: Standard
Standard
From snapshot file

Input file: _____ Browse...

Output file: _____

Standard

The standard method to start an EMTDC case is to simply start from an un-initialized state (i.e. from time $t = 0.0$). This is how simulations are started most of the time.

From Snapshot File

There may be instances when you would like to start your simulation from a pre-initialized state. Although initial conditions are not yet available as direct entry into certain components, it is possible in PSCAD to run a case to a steady state and then to take a snapshot at a certain instant during the run. All relevant network data will be saved to a snapshot file, from which you may start your case already pre-initialized.

An input field is included directly beside this field called **Input File**. Enter a name for the snapshot file to be used here. See the section entitled 'Starting from a Snapshot' in Chapter 5 of this manual for more details on how to create and start from a snapshot file.

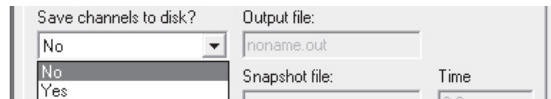
NOTE: When you start from a Snapshot File, make sure that you have not altered the circuit from which the snapshot was taken, otherwise an error may occur.

Chapter 7: Project Settings

Save Channels to Disk?

You can save all Output Channels in your project to a file for post processing. Output files are saved in standard ASCII format and all data is stored in columnar format in steps of time according to the set plot step.

An input field is included directly beside this field called **Output File**. Enter a name for the Output File here.

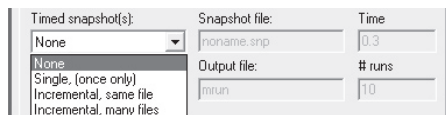


Save channels to disk?	Output file:
No	noname.out
No	Snapshot file:
Yes	Time

Timed Snapshot

There are two ways to utilize a snapshot file: Single and incremental snapshots.

An input field is included directly beside this field called **Snapshot File**. Enter a name for the snapshot file to be created here. Another input field is included called **Time**. Enter the time in seconds at which the snapshot is to be taken.



Timed snapshot(s):	Snapshot file:	Time
None	noname.snp	0.3
None	Output file:	# runs
Single, (once only)	mrun	10
Incremental, same file		
Incremental, many files		

Single (Once Only)

A single snapshot will be taken at the time specified in the **Time** field.

Incremental (Same File)

An incremental snapshot in the same file will take the first snapshot at the specified time and subsequent snapshots at equal intervals equal to this time. The snapshot file will be over-written every time to save disk space and you will end-up with only one file from the last interval.

Incremental (Many Files)

If you would like to save all Snapshot Files, select this option. You can save up to 99 distinct snapshots. If the number of files exceeds 99, the names will be reused starting from the beginning.

PSCAD

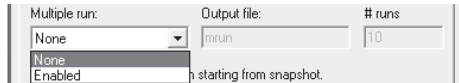
The file naming convention is:

base_name_##.snp

You are required to provide only the 'base_name' (in the **Snapshot File** field). The extension will be added automatically.

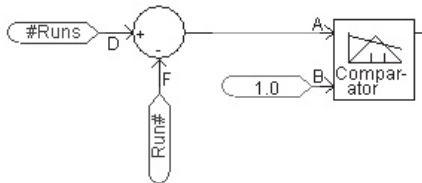
Multiple Run

There are currently two different methods for performing multiple runs in PSCAD - this is the simpler and most basic of the two. This method is used in conjunction with the Current Run Number and Total Number of Multiple Runs components, which are available in the PSCAD Master Library.



EXAMPLE 7-1:

Consider the simple comparator below. The Multiple Run parameter is enabled from the Project Settings dialog and set to 10 runs.



In this case, the comparator output is set to go low (0) when input A is less than or equal to 1.0 (i.e. the last two runs).

An additional input field is included directly beside this field called **Output File**. Enter a name for the Multiple Run Output File to be created here. Another input field is included called **# runs**. Enter the total number of runs here (this value is used to set the Total Number of Multiple Runs component).

NOTE: The other multiple run method involves the Multiple Run component, which is available in the PSCAD Master Library. DO NOT enable this multiple run feature when using the Multiple Run component.

Miscellaneous

The remaining Runtime parameters are outlined below:

- Remove time offset when starting from snapshot.
- Send only the output channels that are in use.
- Start simulation manually to allow use of integrated debugger.
- Enable component graphics state animation.

Remove Time Offset When Starting From a Snapshot

This parameter is used in correlation with starting the simulation from a snapshot file. Enabling this parameter will force the initial start time on all plots to display a 0.0 start time - regardless of what time the snapshot was taken. If disabled, the initial start time will be displayed the time at which the snapshot file was created.

EXAMPLE 7-2:

A user runs a case project to steady state, and then takes a snapshot at 0.5 seconds. The user then re-starts the simulation from the snapshot file created, and sets the **Duration of Run** to 0.05 s.

If **Initial Start Time Always Zero** is enabled, the simulation run will be displayed from 0.0 to 0.05 seconds. If disabled, from 0.5 to 0.55 seconds.

It is important to note here that this option changes displayed start time only. That is, if you are controlling breakers or faults for example, you must use the actual time. In Example 6-2 above, if the user wanted to open a breaker 0.01 seconds after starting from the snapshot, the breaker logic must indicate 0.51 seconds (not 0.01 seconds).

PSCAD

Send Only the Output Channels That are In Use

Selecting this option will essentially turn off all Output Channels that are not being plotted in Graphs, or monitored in Meters. Enabling this option has the potential to greatly reduce the amount of storage for the simulation, as well as a slight simulation speed improvement.

NOTE: Unused Output Channel data will still be written to EMTDC output files if Save Channels to Disk? is selected.

Start Simulation Manually to Allow Use of an Integrated Debugger

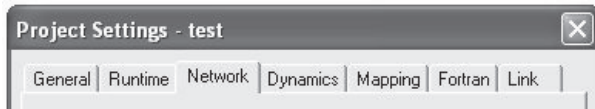
This option allows you to start a PSCAD run that can be connected with a manually started EMTDC case. See the section entitled 'Using an Integrated Debugger' in Chapter 11 of this manual for more details.

Enable Component Graphics State Animation

This option will enable/disable Active Graphics in PSCAD components. Since the Active Graphics algorithm is processor intensive, disabling it will help speed up those cases that contain a lot of animation.

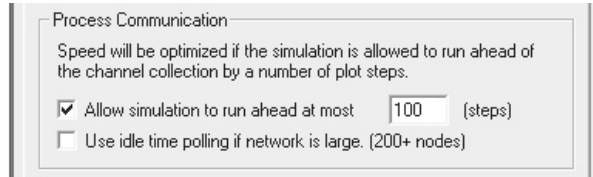
NETWORK

The properties contained within the Network section of the Project Settings dialog control window are related to the EMTDC Electric Network Solution and diagnostics. These are explained in detail in the sections below.



Process Communication

The input parameters involved here are outlined below.



Allow Simulation to Run Ahead at Most

This parameter controls the maximum amount of time (in time steps), by which EMTDC is allowed to run ahead of the PSCAD display. Adjusting this parameter can have the following implications:

1. If the run-ahead time is too small, PSCAD will hold EMTDC back and the overall simulation speed will be reduced.
2. If the run-ahead time is too large, interactive control actions (such as resetting a slider value or pushing a button) are potentially conveyed to EMTDC with a maximum delay equal to this margin.

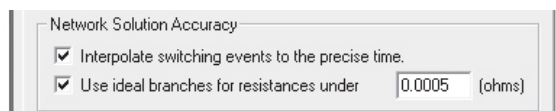
Run-ahead time is something users can optimize based on the case and has no implication on the accuracy of simulation. The default value of 100 steps is a good compromise.

Use Idle Time Polling if Network is Large (200+ Nodes)

To ensure a fast response in the data communication between PSCAD and EMTDC, the simulation of smaller systems is continuously polled, providing maximum performance. On larger networks however, the simulation can run significantly slower and it is not necessary to poll the simulation continuously. To free up processor cycles (and to allow other applications to run more smoothly), this control allows the polling to only occur during periods when the application is not busy, referred to as 'idle time.' If this option is set, larger networks will use this method instead, reducing the processor loading from PSCAD.

Network Solution Accuracy

The following input parameters can affect solution speed.



PSCAD

Interpolate Switching Events to the Precise Time

To account for inter time step switching of switching devices (i.e. those components whose electrical properties change during a run), the EMTDC Electric Network Solution uses an interpolation algorithm in order solve at the exact switching instant. This is the default behaviour of EMTDC and is essential for the accurate simulation of frequently switching devices, such as FACTS models.

Use Ideal Branches for Resistances Under

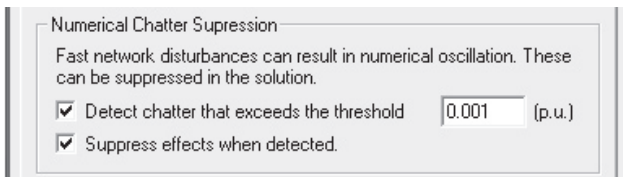
The Ideal Branch algorithm allows for zero resistances and true infinite bus voltage sources in EMTDC.

With this option enabled, you can create an infinite bus by either entering a value less than the threshold (set in the threshold field directly to the right of this check box) or 0.0 Ω for the source resistance. Similarly, for a zero resistance branch, enter a value less than the threshold or 0.0 Ω for the ON resistance of a diode, close resistance of a breaker, etc.

The default threshold value is preset to 0.0005 Ω . As this algorithm involves extra computations and so a non-zero value greater than this threshold is recommended, unless ideal like results are paramount.

Numerical Chatter Suppression

The following input parameters are related to the detection and removal of numerical oscillations called chatter.



Numerical Chatter Suppression

Fast network disturbances can result in numerical oscillation. These can be suppressed in the solution.

Detect chatter that exceeds the threshold (p.u.)

Suppress effects when detected.

Detect Chatter that Exceeds the Threshold

Chatter is a numerical oscillation phenomenon inherent to the trapezoidal integration method used in EMTDC, and is usually caused by sudden network disturbances (either voltage or current). EMTDC continuously looks for chatter and removes it, if required.

Chapter 7: Project Settings

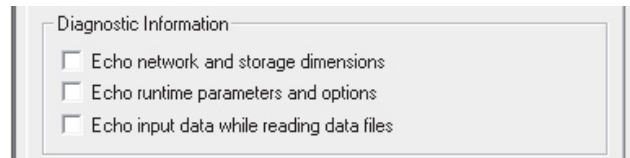
An input field is included directly beside this option. Enter a threshold value for chatter detection, below which chatter will be ignored (the default is 0.001 p.u.).

Suppress Effects When Detected

When chatter is detected, a chatter suppression procedure is invoked if this option is enabled.

Diagnostic Information

The following input parameters are important during the debugging process of your simulation project and are recommended for more advanced users.



Echo Network and Storage Dimensions

Prints network and storage array dimensions into the Output window for debugging purposes.

Echo Runtime Parameters and Options

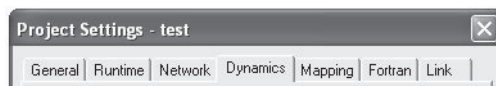
Prints runtime options into the Output window for debugging purposes.

Echo Input Data While Reading Data Files

Prints all the data read from the Data File and Map File into the Output Window for debugging purposes.

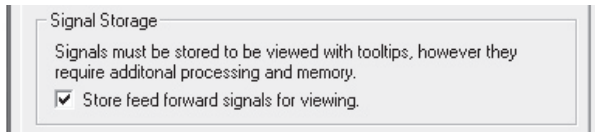
DYNAMICS

The properties contained within the Dynamics section of the Project Settings dialog control window are related to the EMTDC System Dynamics. These are explained in detail in the sections below.



Signal Storage

The input parameters involved here are outlined below.



Store Feed Forward Signals for Viewing

When this control is enabled, all declared data signal variables (be they feed-forward or feed-back signals), will be transferred to and from their designated EMTDC storage array each time step. If this control is disabled, only those declared variables that must be transferred to storage (such as feed-back signals) are considered; all other variables are deemed temporary and are discarded at the end of each time step.

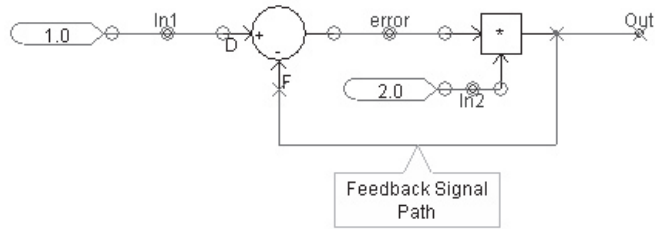
NOTE: It is important to note that the Tool Tip (flyby) feature in PSCAD extracts the value of the signal being monitored from storage. Therefore, if the Store Feed Forward Signals for Viewing control is disabled, **all feed-forward signal values will not appear in the Tool Tips!**

In smaller projects, this toggling control will have very little effect on the simulation speed. As Projects become larger and contain many control signals, this feature may help to speed things up.

EXAMPLE 7-3:

In the simple control system shown below, there will be a total of four, type REAL variables declared in the project Fortran file when this Project is compiled; **In1**, **error**, **Out** and **In2**. Three are feed-forward variables (indicated by the O symbol), and one is a feedback variable (indicated by the X symbol).

Chapter 7: Project Settings



NOTE: Users can display these symbols at any time by enabling the **Show Signal Locations** parameter in the Module Settings dialog. For a more detailed description of these symbols, see the section entitled 'Show Signal Locations' in Chapter 11 of this manual.

If Store Feed Forward Signals for Viewing is **enabled**, all four variables will be written to storage every time step as indicated in the following excerpt from the Project Fortran file:

```
!=====
      SUBROUTINE DSDyn ()
!-----
! Transfers from storage arrays
!-----
      In2      = STOF (ISTOF + 1)
      In1      = STOF (ISTOF + 2)
      error    = STOF (ISTOF + 3)
      Out      = STOF (ISTOF + 4)
!-----
! Feedbacks and transfers to storage
!-----
      STOF (ISTOF + 1) = In2
      STOF (ISTOF + 2) = In1
      STOF (ISTOF + 3) = error
      STOF (ISTOF + 4) = Out
```

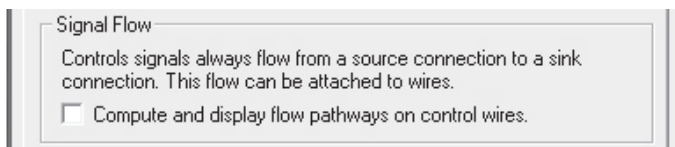
If Store Feed Forward Signals for Viewing is **disabled**, all four variables will be written to storage every time step as indicated in the following excerpt from the Project Fortran file:

PSCAD

```
!-----  
SUBROUTINE DSDyn()  
!-----  
! Transfers from storage arrays  
!-----  
Out = STOF(ISTOF + 4)  
!-----  
! Feedbacks and transfers to storage  
!-----  
STOF(ISTOF + 4) = Out
```

Signal Flow

The input parameters involved here are outlined below.

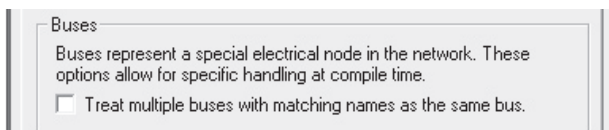


Compute and Display Flow Pathways on Control Wires

Select this option to show signal flow direction on control Wires (i.e. Wires carrying REAL, INTEGER or LOGICAL data types). See the section entitled 'Control Signal Flow Indicators' in Chapter 11 of this manual for more details on this.

Buses

The input parameters involved here are outlined below.



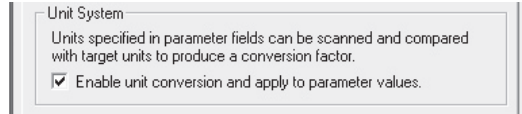
Treat Multiple Buses with Matching Names as the Same Bus

Select this option to treat multiple buses with the same name as the same bus. That is, graphically separated bus points with the same name will represent the same EMTDC node points in the electrical grid. See the Bus component for more.

Chapter 7: Project Settings

Unit System

The input parameters involved here are illustrated below:



Enable Unit Conversion and Apply to Parameter Values

Select this option to enable the Unit System.

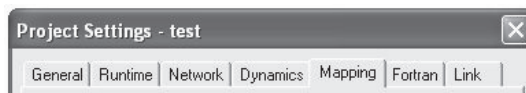
NOTE: Enabling the Unit System may cause compilation failures in previously created Case Projects. This is normally due to input parameter units in existing components that are not recognized by the Unit System compiler. If compilation failure does indeed occur:

1. Consult the Output Window for error messages: Point to the error source by Locating the Problem Source (see the section entitled 'Output Window' in Chapter 4).
2. View the parameters of the component (see the section entitled 'Viewing/Saving Component Parameters' in Chapter 5).
3. Scan through the Value column and look for a #NaN (Not a Number) symbol. This usually indicates that the entered unit does not match the Target Unit defined for this input parameter.

This parameter is enabled by default in new Case Projects created in PSCAD V4.2+. It is disabled by default in existing cases.

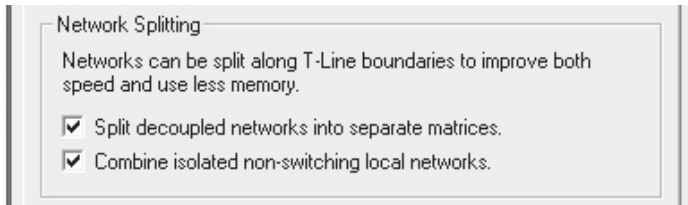
MAPPING

The properties contained within the Mapping section of the Project Settings dialog control window are related to the EMTDC Network Solution conductance matrix optimization. These are explained in detail in the sections below.



Network Splitting

The input parameters involved here are outlined below.



Split Decoupled Networks into Separate Matrices

Matrix solution methods can be intensive, requiring a large amount of computing power, especially if frequently switching branches are involved. If this option is enabled, larger networks will be reduced into smaller sub-networks (see the section entitled 'Subsystems in Electric Networks' in the EMTDC manual for more details on setting up subsystems).

Once the main electric network has been split into subsystems, each subsystem can then be solved independently. Say for example, a large FACTS device (with many frequently switched branches) is located within its own subsystem, but part of a much larger network. By splitting the main network into subsystems, only the local subsystem, housing the FACTS device, need be resolved when a switching event occurs. The speed benefits here can be tremendous.

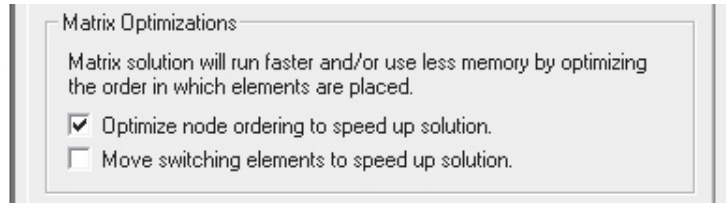
In PSCAD V3, a well-known shortcoming of the subsystem splitting algorithm was that the main project page could not contain any electrical nodes. In PSCAD V4, this problem has been resolved.

Combine Isolated Non-Switching Local Networks

This feature is available only if Split Decoupled Networks into Separate Matrices is enabled. When this control is enabled, all subsystems that do not contain switching devices are lumped together. This process can substantially lower the memory usage by EMTDC when storing electric network data.

Matrix Optimizations

The input parameters involved here are outlined below.



Optimize Node Ordering to Speed Up Solution

Optimize Node Ordering is a PSCAD based algorithm, which re-numbers nodes in the EMTDC Electric Network conductance matrix, so as to optimize solution speed. The electric network conductance matrix is optimized using a Tinney algorithm to exploit matrix sparsity. For more information on matrix optimization, please see References [7], [8], [9] and [10].

Move Switching Devices to Speed Up Solution

Frequently switching branches are identified and re-ordered, so as to optimize conductance matrix re-triangularization following a change in branch conductance (a switch). The switch-ordering algorithm splits nodes into two types:

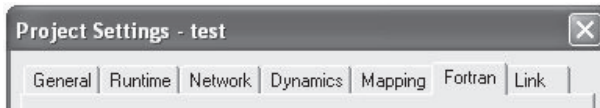
1. Nodes not attached to switching elements or nodes that have switching elements connected, but do not switch frequently (i.e. breakers and faults).
2. Nodes that have switching elements connected that switch frequently (i.e. thyristor, GTO, IGBT, diode, arrester, Variable RLC, etc.).

NOTE: Electrical Connections in user-defined components are set as node type 2 by changing the electrical Connection type to 'Switched.'

The switch-ordering algorithm moves the frequently switched nodes (i.e. type 2) to the bottom of the conductance matrix. The benefits to EMTDC solution speed are proportional to the number of nodes and the number of switching branches in a given electric network. For more information on matrix optimization, please see References [7], [8], [9] and [10].

FORTRAN

The parameters contained within the Fortran section of the Project Settings dialog are used for the control of Fortran compiler based error and warning messages. Also, any additional Fortran source files required for the compilation of the project are indicated here, as well.



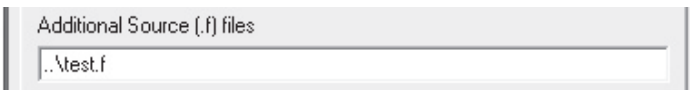
Additional Source (.f) Files

This input field allows you to link one or more external source code files, so as to include the contained code during compilation of the associated project. An external source code file could contain one or more subroutines, which are called from within the Fortran segment of a user component, for example.

Files may be referenced in this field either by an absolute, or relative path specification. For example, a file referenced with an absolute path may appear as follows:



If only the filename is entered, PSCAD will assume that the source file is located in the same directory as the project file itself, and will append this path to the filename. You may also use standard directory navigational features when working with relative paths. For example, if a source file called 'test.f' is located in the directory directly above the project file, then the file entry would appear as follows:



When entering multiple source files, each entry must be separated either by a blank space, a comma or a semicolon:

Chapter 7: Project Settings



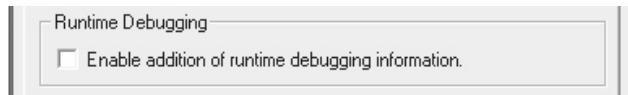
This input field will only accept the following source file types:

- **Fortran:** *.f, *.f90, *.for
- **C:** *.c

In PSCAD V3, linking to external files was accomplished through the use of a File Reference component - this method is still valid.

Runtime Debugging

The input parameters involved here are outlined below.



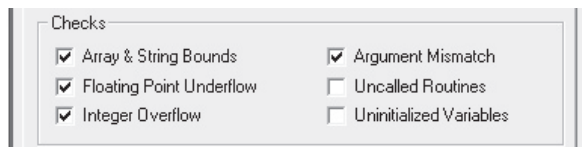
Enable Addition of Runtime Debugging Information

This option will add some additional information to the build, so as to allow for the effective use of the Fortran debugger. If you have selected this option, you should also select all of the Checks available in the Fortran section.

When a program crashes on a PC, the operating system brings up a dialog asking if you would like to debug the case. If you select yes and if you had this option selected, the Fortran debugger can load the source file and point to the line of code that is causing the crash.

Checks

The input parameters involved here are outlined below.



Array & String Bounds

With this option selected, the program will stop if you are accessing an invalid array index. For example, if the array X is declared to be of dimension 10, and you have a line of source code that has X(J). If 'J' ever goes greater than 10, the program will stop with a proper message. If this option is de-selected, the program will continue execution and may eventually crash, adding difficulty to tracing the cause.

This option should be used when testing your new models. It will slow down the simulation and the speed penalty will vary from machine to machine - you can disable this option if speed is a concern. Please read the Fortran compiler documentation you are using for more details.

Floating Point Underflow

This option is useful for debugging but has a speed penalty that is again architecture dependent. Please read the Fortran compiler documentation for more details.

Integer Overflow

This option is useful for debugging but has a speed penalty that is again architecture dependent. Please read the Fortran compiler documentation for more details.

Argument Mismatch

The Fortran compiler issues a warning if the argument type (REAL, INTEGER, etc.) of the CALL statement does not match with the sub-routine statement (this is applicable to functions, as well).

Do not ignore this warning as it may have the potential to create unpredictable and hard to trace results.

Uncalled Routines

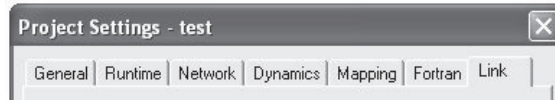
The Fortran compiler issues a warning if a routine is not called in the program.

Uninitialized Variables

A warning is issued if a variable is used before it is assigned a value. Normally, the cause of this is a typographical mistake in the source code.

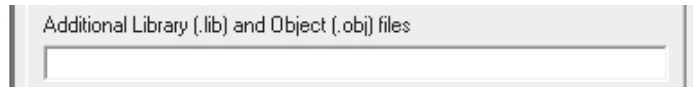
LINK

The parameters contained within the Links section of the Project Settings dialog are used for the linking of pre-compiled library (including MATLAB related) and object files.



Additional Library (.lib) and Object (.obj) Files

This input field allows you to indicate extra Fortran object (*.obj) or library (*.lib) files, which must be linked to the project before compilation. Object and library files would normally be used if you have a lot of custom models associated with user-written subroutines, where it would be more efficient to create a library or object file rather than maintain a collection of Fortran files.



Object and library files provide a way to avoid the compilation of user-written subroutines every time they are used in different projects. They are also useful for sharing custom models with others, where you do not want to share the source code itself.

User Library Path Method

In PSCAD V4, a new compiler specific directory convention has been added for the purpose of specifying compiler specific object and library files. Now, object and library files can be compiled with one or more different Fortran compilers, and then placed in a corresponding sub-directory of the specified User Library Path directory. Then, the user may freely switch between compilers without the need to re-specify the files.

The user must manually add these sub-directories to the directory specified by the User Library Path input in the Workspace dialog (go to **Edit | Workspace Settings...** in the Main Menu). Each sub-directory represents a specific Fortran compiler, and the same library or object file (compiled by the corresponding compiler) may be placed. Since there are three types of Fortran compilers that may be

PSCAD

used with PSCAD, there can be up to three sub-directories specified, which must be named as follows:

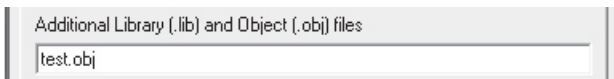
- gnu (EGCS/GNU Fortran 77 compiler)
- df5 (Digital Fortran 90 compiler)
- cf6 (Compaq Fortran 90 compiler)

For example, if a user specifies '**C:\my_libs**' as the User Library Path, and intends to use either the EGCS/GNU Fortran 77 compiler or the Compaq Fortran 90 compiler, then two sub-directories should be added to 'my_libs,' as shown below:



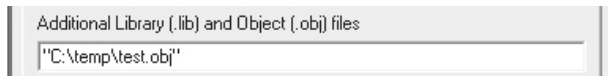
Any library or object files required should be created with both compilers, and then added to the corresponding sub-directory.

If only the filename is entered, PSCAD will assume that the library or object file is located in the specified sub-directory in the User Library Path directory. For example, if a user specifies '**C:\my_libs**' as the User Library Path and is using the Compaq Fortran 90 compiler, entering 'test.obj,' as shown below, would indicate to PSCAD that the specified file is located at **C:\my_libs\cf6\test.obj**.



Absolute Path Method

Files may also be referenced by either an absolute, or relative path specification. If this method is chosen, then the directory specified in the User Library Path directory will be overridden. For example, a file referenced with an absolute path may appear as follows:

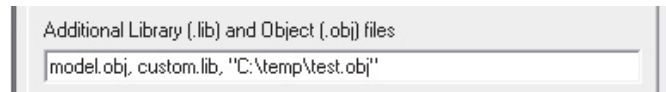


Chapter 7: Project Settings

NOTE: Although it is not a necessity, the absolute path should appear within quotes. This will ensure that if any spaces exist in a path directory or the filename itself, it will be parsed properly.

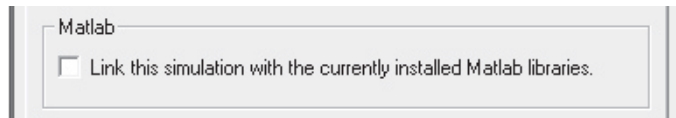
Entering Multiple Files

When entering multiple files into this field, they must be separated with a space, a comma or a semicolon. Both User Library Path and absolute path methods may be used simultaneously.



Matlab

The input parameters involved here are outlined below.



Link This Simulation with the Currently Installed Matlab Libraries

Select this option if you intend to utilize the MATLAB®/Simulink® interface with this project. Note that this option is disabled unless you have specified a MATLAB® version in the Workspace Settings dialog.

Transmission Lines and Cables

Overhead transmission line and underground cable corridors (or right-of-ways) are represented in PSCAD in two main parts: By defining the configuration of the transmission corridor itself, where the definition can include either the admittance/impedance data or the conductor and insulation properties, ground impedance data, and geometric position of all conductors within the corridor. This definition is then interfaced to the rest of the electrical system through electrical interface components (one at each end). If the transmission line is set up in Direct Connection mode, the Interface components are not required.

Transmission systems of very short length (i.e. less than 15 km for a 50 μ s time step) can also be represented using an equivalent PI Section. This is accomplished by either using the PI Section component in the Master Library, where only the admittance and impedance data of the line segment is required, or by using the PI Section equivalent component creator in the Additional Options component. PI Sections are essentially a network of passive elements, and hence do not represent propagation delay.

NOTE: A 15 km line length to a 50 μ s time step is derived assuming that waves propagate through the line at the speed of light. In general however, wave propagation speed is less than the speed of light.

Using the data provided by the cross-sectional definition of the corridor, the transmission lines and cables are modeled using one of three distributed (travelling wave) models:

- Bergeron
- Frequency Dependent (Mode)
- Frequency Dependent (Phase)

The most accurate of these is the Frequency Dependent (Phase) model, which represents all frequency dependent effects of a transmission line, and should be used whenever in doubt. When using

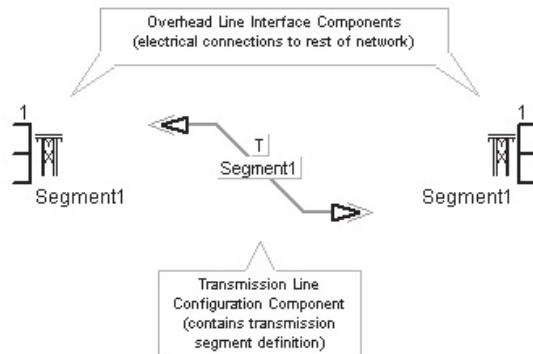
Chapter 8: Transmission Lines and Cables

the Bergeron model, impedance/admittance data can also be entered directly to define the transmission corridor.

For all of these Frequency Dependent models, detailed conductor information (i.e. line geometry, conductor radius) must be given.

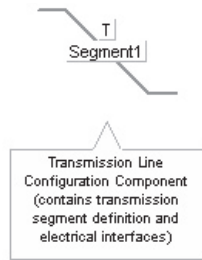
CONSTRUCTING OVERHEAD LINES

There are two basic ways to construct an overhead line in PSCAD. The first is the original method (referred to as the **Remote Ends** method), which involves a Transmission Line Configuration component with two Overhead Line Interface components, representing the sending and receiving ends of the line. The purpose of the Interface components is to connect the transmission line to the greater electric network. This style of line is illustrated below:



An Overhead Transmission Line (Remote Ends Method)

The second and more recently introduced method is to use the **Direct Connection** method, where the interfaces and the corridor properties are housed within a single component. This method however, can only be used for 1-phase, 3-phase or 6-phase single-line systems, where the maximum number of conductors is 6.



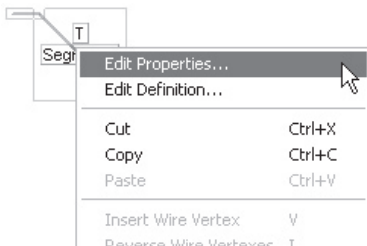
An Overhead Transmission Line (Direct Connection Method)

Building an Overhead Line (/w Remote Ends)

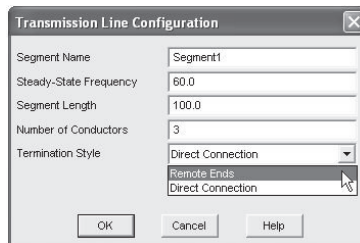
Add one Transmission Line Configuration component and two Overhead Line Interface components to the page. The most straightforward method is to use the Electrical Palette. See Adding Components to a Project for details.



Next, edit the properties of the configuration component and ensure that **Termination Style | Remote Ends** is selected.



Edit the Properties of the Transmission Line Configuration component



Set Termination Style to Remote Ends

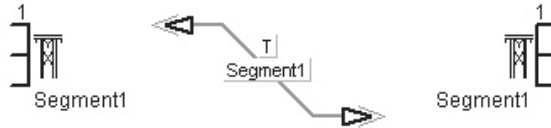
The Transmission Line Configuration component and two Overhead Line Interface components are interlinked through input parameters. Ensure that the following steps are taken:

1. **Segment Name:** Ensure that the Segment Name inputs in both the Transmission Line Configuration component and two Overhead Line Interface components are identical.

Chapter 8: Transmission Lines and Cables

2. **Number of Conductors:** Ensure that the number of conductors in both the Transmission Line Configuration component and two Overhead Line Interface components are identical.

When finished, you should have something similar to that shown below on your project page:



A 3-Conductor Overhead Transmission Line
(Remote Ends Method)

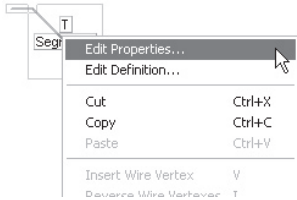
NOTE: These components do not need to be in proximity to one another. As long as they are connected by name, each component can be located anywhere in the Project, including different Modules.

Building an Overhead Line (Direct Connection Style)

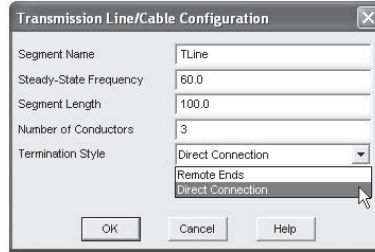
Add one Transmission Line Configuration component to the page. The most straightforward method is to use the Electrical Palette. See 'Adding Components to a Project' in Chapter 5 for details.



Next, edit the properties of the configuration component and ensure that **Termination Style | Direct Connection** is selected.



Edit the Properties of the Transmission Line Configuration component



Set Termination Style to Direct Connection

When finished, you should have something similar to that shown below on your project page:

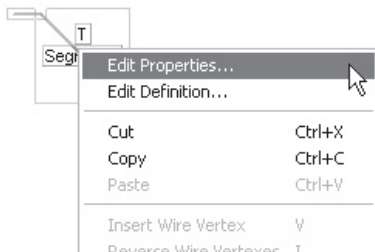


An Overhead Transmission Line (Remote Ends Method)

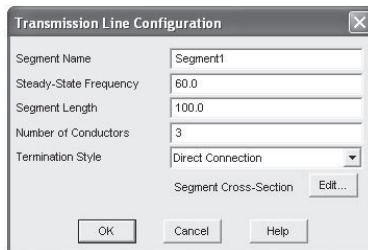
NOTE: The Transmission Line Configuration component is sizable (very similar to a Wire component) and may assume a variety of shapes and sizes.

Editing the Line Main Parameters

The main properties of the overhead line may be adjusted directly within the Transmission Line Configuration component. Simply right-click over the component and select **Edit Properties....**



Edit the Properties of the Transmission Line Configuration component



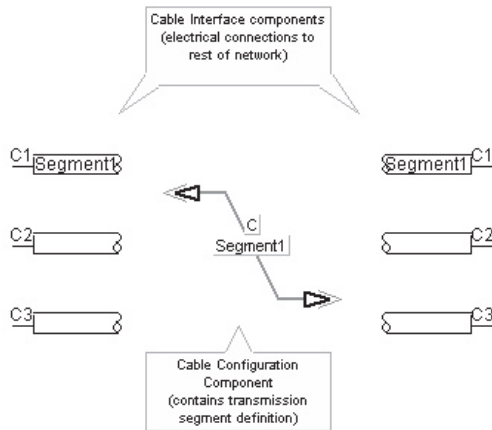
Transmission Line Configuration Component Properties Dialog

Chapter 8: Transmission Lines and Cables

The properties given in this dialog are common to all elements of the transmission line. For information on these parameters, see Transmission Line Configuration. To edit parameters specific to towers, conductors and ground, select the **Edit...** button. See 'Editing an Overhead Line Segment Cross-Section' in this chapter for details.

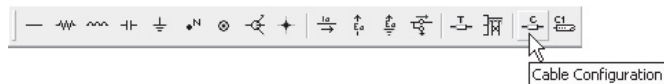
CONSTRUCTING UNDERGROUND CABLE SYSTEMS

Unlike overhead lines, the **Direct Connection** method cannot be used with cable systems. This leaves just the **Remote Ends** method, which involves a Cable Configuration component with two Cable Interface components, representing the sending and receiving ends of the cable. The purpose of the Interface components is to provide connection of the cable to the greater electric network. An underground cable system is illustrated below:



Building an Underground Cable System

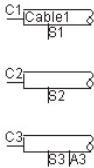
Add one Cable Configuration component and two Cable Interface components to the page. The most straightforward method is to use the Electrical Palette. See 'Adding Components to a Project' in Chapter 5 for details.



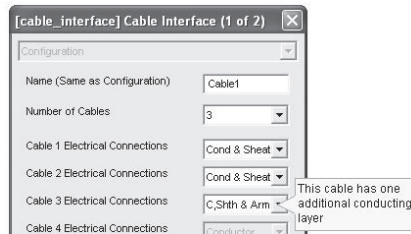
PSCAD

The Cable Configuration component and two Cable Interface components are interlinked through input parameters. Ensure that the following steps are taken:

1. **Segment Name:** Ensure that the Segment Name inputs in both the Cable Configuration component and two Cable Interface components are identical.
2. **Number of Cables:** The Number of Cables parameter in the Cable Interface components specifies how many individual cables there are in the system. Each of these cables however, could contain different numbers of conducting layers, and so the total number of conductors is arbitrary (this is why the **Number of Conductors** parameter is disabled in the Cable Configuration component). The user is required to specify the number of conducting layers for each cable in the Cable Interface components. This data must match what is defined within the Cable Segment Cross-Section.



Cable Interface Component



Specifying Cable Set-up in Cable Interface Parameters Dialog

NOTE: These components do not need to be in proximity to one another. As long as they are connected by name, each component can be located anywhere in the Project, including different Modules.

Editing the Cable Main Parameters

The main properties of the cable may be adjusted directly within the Cable Configuration component. Simply right-click over the component and select **Edit Properties...**

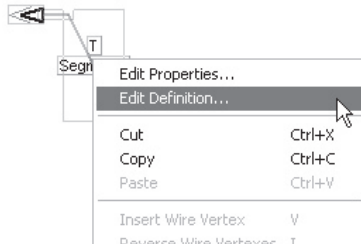
PSCAD

The main section is entitled Editor, and is usually the default section view when the editor is invoked. The four remaining sections are for viewing transmission system related files, after the project has been built.

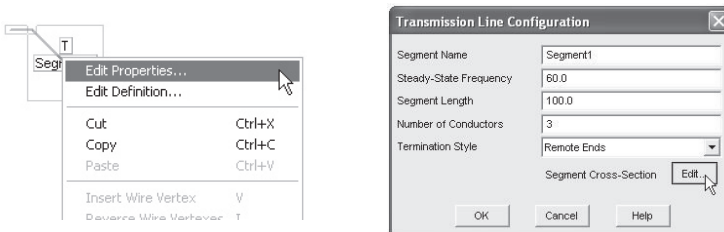
The T-Line/Cable Properties Editor can be accessed through either the Transmission Line or Cable Configuration components. See Editing a Transmission Line Configuration or Editing a Cable System Configuration for more on invoking this editor.

Editing a Transmission Line Segment Cross-Section

To invoke the T-Line / Cable Properties Editor, right-click over the Transmission Line Configuration component (without selecting it) and select **Edit Definition**, as shown below:



Alternatively, right-click over the Transmission Line Configuration component and select **Edit Properties...**. The editor can then be accessed from within the Configuration dialog window by pressing the **Edit...** button, as shown below:

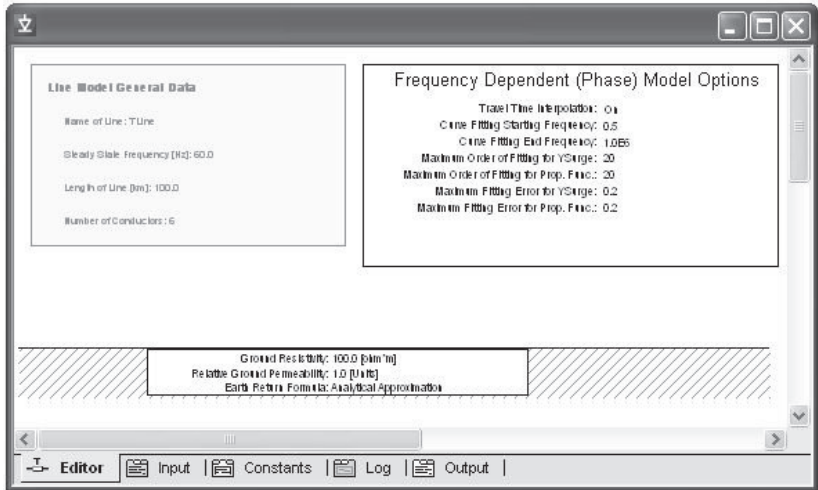


Edit the Properties of
the Transmission Line
Configuration component

Press the Edit Button

Chapter 8: Transmission Lines and Cables

In either case, the T-Line/Cable Configuration Editor will then open overtop of the Design Editor. As shown below, the default view is the Editor section, where the transmission line is graphically defined.

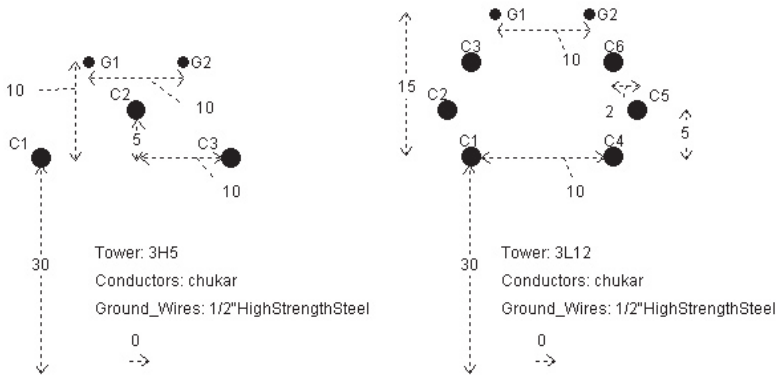


By default, the Editor section will contain three graphical objects:

- **Line Model General Data:** Located in the top-left corner, this object simply displays what has been entered into the corresponding Transmission Line Configuration dialog. This object is for display only and cannot be edited from the T-Line/Cable Configuration Editor.
- **Frequency Dependent (Phase) Model Options:** This component represents the transmission line model being used, and by default is the Frequency Dependent (Phase) model component. The properties of this component may be edited by either a left double-click on the component (or right-click and select **Properties...**) to bring up the corresponding dialog window.
- **Entry of Ground Data:** This component (called the Ground Plane), usually located near the bottom of the Editor window, represents the transmission line ground return path. The properties of this component may be edited by either a left double-click on the component (or right-click and select **Properties...**) to bring up the corresponding dialog window.

A fourth object required is the definition of the transmission line itself. This definition can be a geometrical cross-section of the Transmission Line Tower Component (or towers), or can be a Manual

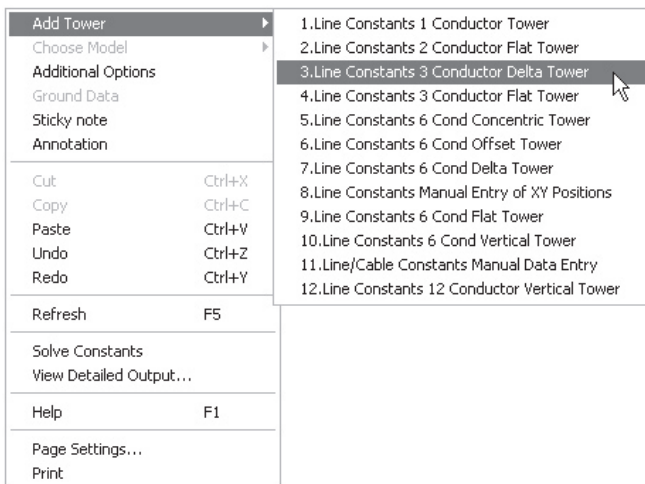
Entry of Admittance/Impedance Data (Bergeron model only). In either case, this is left for the user to add manually.



Transmission Line Tower Components

Adding a Tower Component

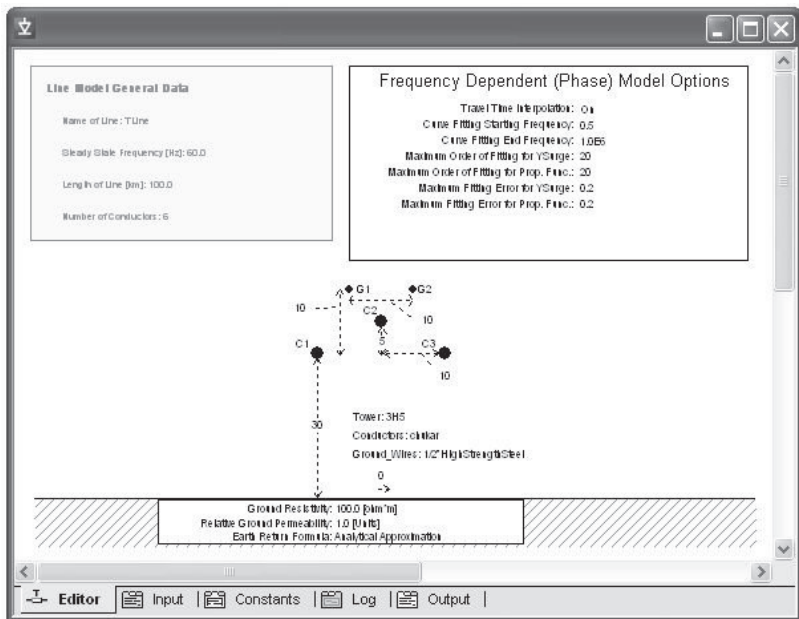
Overhead Line Tower components can be added to the T-Line/Cable Configuration Editor in one of two ways. The most straightforward method is to use the right-click pop-up menu: In the Editor section of the T-Line/Cable Configuration Editor, move the mouse pointer over a blank area of the window. Right-click and select **Add Tower**. A sub-menu will appear containing a list of all transmission line tower components available in the Master Library. Select a tower and it will be automatically added.



Chapter 8: Transmission Lines and Cables

You can also copy and paste tower components directly from the Master Library. Open the Master Library in Circuit view and then open the 'Tlines' module on the main page. Select an Overhead Line Tower (at the bottom of the page), right-click on the component and select **Copy** (or press **Ctrl + C**). Open the Editor section of the T-Line/Cable Configuration Editor, right-click over a blank area and select **Paste** (or press **Ctrl + V**).

When finished, you should have something similar to that shown below in the Editor section of the T-Line/Cable Configuration Editor:



NOTE: The graphical location of the tower component does not affect the results. However, the tower (or towers) should be positioned to allow for ease in readability. That is, directly on top of the Ground Plane.

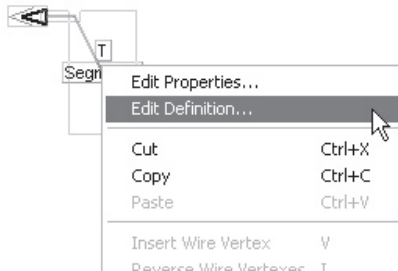
Multiple towers may be added to a single configuration. Just remember to ensure that the conductors are numbered properly within each of the tower components, and that the X positions of the new towers in the corridor are adjusted. Also, any conductors added by the additional towers must be reflected in the corresponding Transmission Line Interface components (if in Remote Ends mode).

Editing Tower Properties

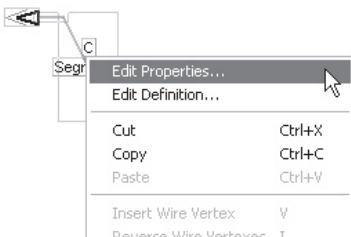
Tower properties can be edited through the corresponding tower properties dialog window. Right-click over the Overhead Line Tower (without selecting it) and select **Properties...** to access this dialog.

Editing a Cable Segment Cross-Section

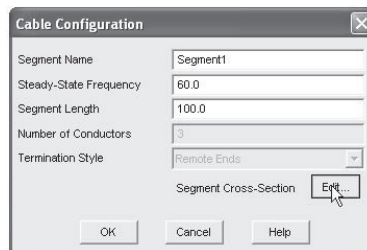
To invoke the T-Line / Cable Properties Editor, right-click over the Cable Configuration component (without selecting it) and select **Edit Definition**, as shown below:



Alternatively, right-click over the Cable Configuration component and select **Edit Properties...**. The editor can then be accessed from within the Configuration dialog window by pressing the **Edit...** button, as shown below:

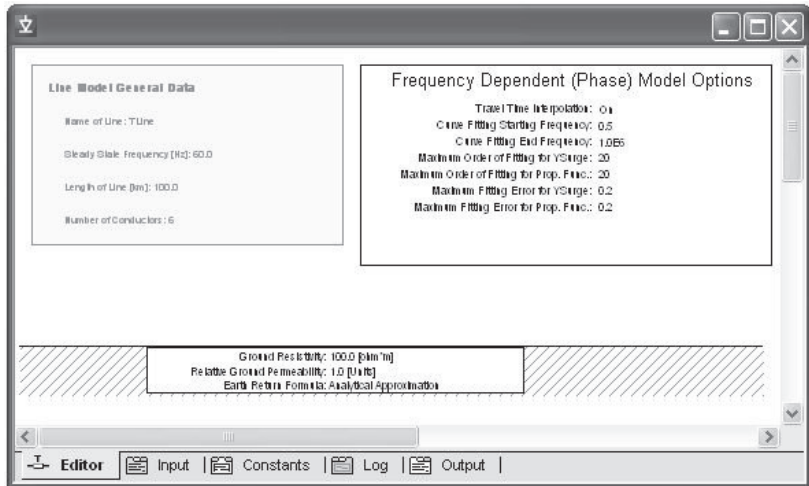


Edit the Properties of the Cable Configuration component



Press the Edit Button

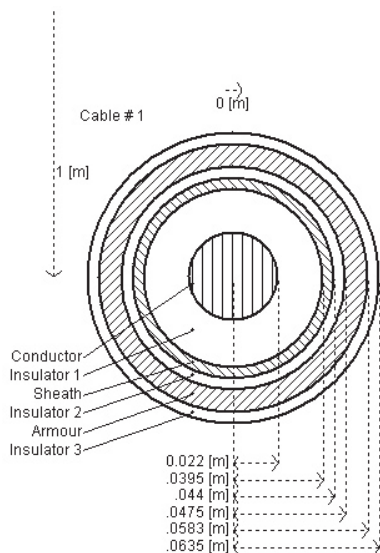
In either case, the T-Line/Cable Configuration Editor will then open overtop of the Design Editor. As shown below, the default view is the Editor section, where the cable system is graphically defined.



By default, the Editor section will contain three graphical objects:

- **Line Model General Data:** Located in the top-left corner, this object simply displays what has been entered into the corresponding Cable Configuration dialog. This object is for display only and cannot be edited from the T-Line/Cable Configuration Editor.
- **Frequency Dependent (Phase) Model Options:** This component represents the transmission line model being used, and by default is the Frequency Dependent (Phase) model component. The properties of this component may be edited by either a left double-click on the component (or right-click and select **Properties...**) to bring up the corresponding dialog window.
- **Entry of Ground Data:** This component, usually located near the bottom of the Editor window, represents the transmission line ground return path. The properties of this component may be edited by either a left double-click on the component (or right-click and select **Properties...**) to bring up the corresponding dialog window.

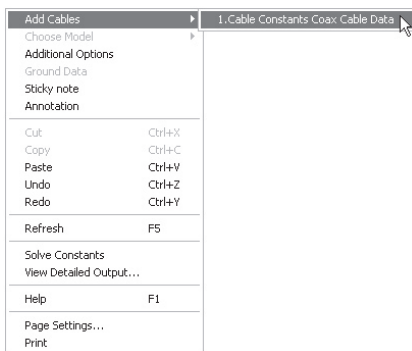
A fourth object required is the definition of the cable itself. This definition is a geometrical cross-section of the cable called a Cable Cross-Section. The user must add this manually.



Cable Cross-Section Component

Adding a Cable Cross-Section Component

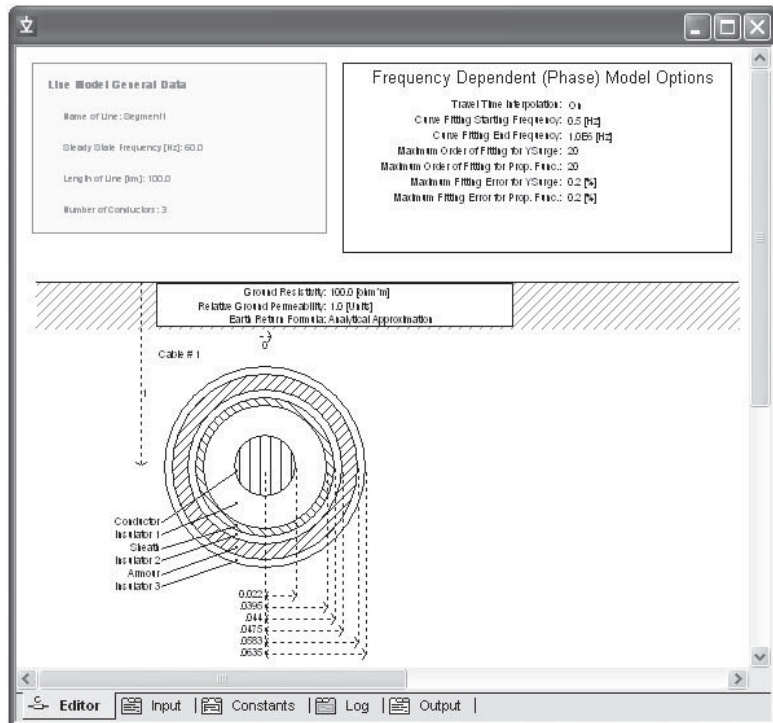
Cable Cross-Section components can be added to the T-Line/Cable Configuration Editor in one of two ways. The most straightforward method is to use the right-click pop-up menu: In the Editor section of the Editor, move the mouse pointer over a blank area of the window. Right-click and select **Add Cable**. A sub-menu will appear containing a list of all cross-section components available in the Master Library. Select a cross-section and it will be automatically added.



Chapter 8: Transmission Lines and Cables

You can also copy and paste cable cross-section components directly from the Master Library. Open the Master Library in Circuit view and then open the 'Cables' module on the main page. Select a cross-section component (at the bottom of the page), right-click on the component and select **Copy** (or press **Ctrl + C**). Open the Editor section of the T-Line/Cable Configuration Editor, right-click over a blank area and select **Paste** (or press **Ctrl + V**).

When finished, you should have something similar to that shown below in the Editor section of the T-Line/Cable Configuration Editor:



NOTE: The location of the Cable Cross-Section component does not affect the results. However, the cross-section (or cross-sections) should be positioned to allow for ease in readability. That is, directly below the Ground Plane.

Multiple cables may be added to a single configuration. Just remember to ensure that the conductors are numbered properly within each

of the cross-section components, and that the X positions of the new cross-sections in the corridor are adjusted. Also, any conductors added by the additional cross-sections must be reflected in the corresponding Cable Interface components.

Editing Cross-Section Properties

Cable cross-section properties can be edited through the corresponding cross-section properties dialog window. Right-click over the cross-section component (without selecting it) and select **Edit Parameters...** to access this dialog.

Selecting the Proper Line Model

There are three types of distributed (i.e. travelling wave) transmission models that may be selected to represent your transmission corridor: The Bergeron model, the Frequency-Dependent (Mode) model, and the Frequency-Dependent (Phase) model. These models exist as components in the Master Library, and each include adjustable properties. The requirements for your study will determine which of the three models is suitable.

Bergeron Model Options	
Travel Time Interpolation:	On
Reflectionless Line (ie Infinite Length):	No

Frequency Dependent (Phase) Model Options	
Travel Time Interpolation:	On
Curve Fitting Starting Frequency:	0.5 [Hz]
Curve Fitting End Frequency:	1.0E6 [Hz]
Maximum Order of Fitting for YSurge:	20
Maximum Order of Fitting for Prop. Func.:	20
Maximum Fitting Error for YSurge:	2 [%]
Maximum Fitting Error for Prop. Func.:	2 [%]

Frequency Dependent (Mode) Model Options	
Travel Time Interpolation:	On
Curve Fitting Starting Frequency:	0.5 [Hz]
Curve Fitting End Frequency:	1.0E6 [Hz]
Maximum Order of Fitting for ZSurge:	20
Maximum Order of Fitting for Prop. Func.:	20
Maximum Fitting Error for ZSurge:	2 [%]
Maximum Fitting Error for Prop. Func.:	2 [%]

Chapter 8: Transmission Lines and Cables

The Bergeron Model

The Bergeron model represents the L and C elements of a PI section in a distributed manner (not using lumped parameters like PI sections). It is accurate at the specified frequency and is suitable for studies where the specified frequency load-flow is most important (e.g. relay studies).

When using the Bergeron model, it is not always necessary to use a tower component to represent a transmission line. If you are modeling a three-phase system, then you can enter the line data, in admittance or impedance format, directly by substituting the Manual Entry of Y, Z component.

Manual Entry of Y,Z

```
+ve Sequence R: .6861e-7 [pu/m]
+ve Sequence XL: .951e-6 [pu/m]
+ve Sequence XC: .571e6 [pu*m]
0 Sequence R: .7175e-6 [pu/m]
0 Sequence XL: .251e-5 [pu/m]
0 Sequence XC: .793e6 [pu*m]
```

This component can be added just as you were adding a tower component. Keep in mind that this component substitutes for the tower component, and you still need the Bergeron model present in your transmission line configuration.

The Frequency-Dependent (Mode) Model

The Frequency-Dependent (Mode) model represents the frequency dependence of all parameters (not just at the specified frequency as in the Bergeron model). This model uses modal techniques to solve the line constants and assumes a constant transformation. It is therefore only accurate for systems of ideally transposed conductors (or two conductor horizontal configurations) or single conductors.

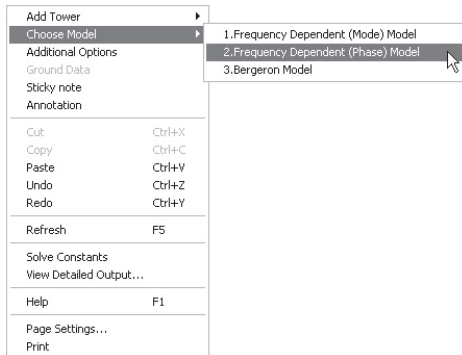
The Frequency-Dependent (Phase) Model

The Frequency-Dependent (Phase) model also represents the frequency dependence of all parameters as in the 'Mode' model above. However, the Frequency Dependent (Phase) model circumvents the constant transformation problem by direct formulation in the phase domain. It is therefore accurate for all transmission configurations, including unbalanced line geometry.

The Frequency-Dependent (Phase) model should always be the model of choice, unless another model is chosen for a specific reason. This model is the most advanced and accurate time domain line model in the world!

Adding a Line Model

Line model components can be added to the T-Line/Cable Configuration Editor in one of two ways. The most straightforward method is to use the right-click pop-up menu: In the Editor section of the Editor, move the mouse pointer over a blank area of the window. Right-click and select **Choose Model**. A sub-menu will appear containing a list of all model components available in the Master Library. Select a model and it will be automatically added.



NOTE: The line configuration cannot contain more than one model component. This menu item will be disabled if a model already exists in the Editor.

You can also copy and paste model components directly from the Master Library. Open the Master Library in Circuit view and then open either the 'TLines' module or the 'Cables' module on the main page. Select a line model component (middle of the page), right-click on the component and select **Copy** (or press **Ctrl + C**). Open the Editor section of the T-Line/Cable Configuration Editor, right-click over a blank area and select **Paste** (or press **Ctrl + V**).

Editing Line Model Parameters

Line model properties can be edited through the corresponding model properties dialog window. Right-click over the line model component (without selecting it) and select **Edit Parameters...** to access this dialog.

LINE CONSTANTS FILES

Whenever a project containing a transmission line or a cable system is compiled (or the line constants are solved manually), text files involved in the calculation of the line constants are created. PSCAD uses information extracted from the various components involved with the transmission system, and constructs a line constants input (*.tli) file. This file is then used as input by the T-Line/Cable Constants program - a separate executable supplied with your PSCAD software.

Depending on the transmission system (i.e. overhead line or underground cable) and the actual line model used, the T-Line/Cable Constants program creates up to three main output files. These are:

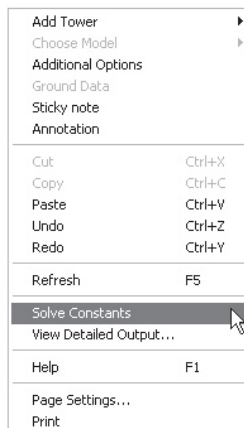
- Constants File (*.tlo/* .clo)
- Log File (*.log)
- Output File (*.out)

These files are located in the project temporary (*.emt) directory and can may be viewed directly from within the The T-Line/Cable Properties Editor.

Solving the T-Line/Cable Constants Manually

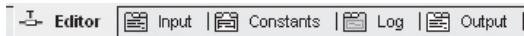
To manually solve a specific transmission line or cable segment in your project, first open the The T-Line/Cable Properties Editor.

Right-click over a blank area of the Editor page and select **Solve Constants**.



Viewing Line Constants Files

There are a few files involved with the T-Line/Cable Constants program. These are the Input, Constants, Log and Output files. To view any of these files, first execute the T-Line/Cable Constants program. Then, simply click the corresponding tab in the T-Line Properties Editor tab bar, located near the bottom of the Design Editor window.



The tab window viewers specified in this bar correspond as follows:

- **Editor:** T-Line (or Cable) Properties Editor
- **Input:** Input (*.tli/*.cli) File
- **Constants:** Constants (*.tlo/*.clo) File
- **Log:** Log (*.log) File
- **Output:** Output (*.out) File

Input (*.tli/*.cli) File

The line constants input file is automatically constructed by PSCAD when the project is compiled, or when the line constants are solved manually (see Solving the T-Line/Cable Constants Manually).

This file, which has extension *.tli, is used as an input file to the T-Line/Cable Constants program, which loop reads through the file, controlled by key statements. It is for this reason that the Input file cannot be manually adjusted. All quantities specified in this file become global variables within the T-Line/Cable Constants program.

The file is pieced together using information taken from:

1. **Configuration Component:** Depending on whether you are modeling a transmission line or a cable corridor, the first part of this file is composed of parameters taken directly from the Transmission Line or Cable Configuration component, as shown below for a transmission line:

```
Line Summary:
{
Line Name = FLAT230
Line Length = 100.0
Steady State Frequency = 60.0
Number of Conductors = 3
}
```

Chapter 8: Transmission Lines and Cables

- 2. Tower or Cross Section Component:** Important parameters, such as conductor data and ground data, dimensions, etc. are extracted from the Model-Data segments of whatever t-line tower or cable cross-section components exist in the T-Line/Cable Configuration Editor. These are placed in the input file as well and represent the bulk of the file:

```
Line Constants Tower:
{
  Name = H-Frame-3H4
  Circuit = 1
  {
    Transposed = 0
    Conductors = 3
    Conductor Phase Information = 1 2 3
    Radius = 0.0203454
    DCResistance = 0.03206
    ShuntConductance = 1.0e-011
    P1 = -10.0 30.0
    P2 = 0.0 30.0
    P3 = 10.0 30.0
    Sag = 10.0
    Sub-ConductorsPerBundle = 2
    {
      BundleSpacing = 0.4572
    }
  }
  GroundWires = 2
  {
    Eliminate Ground Wires = 1
    Radius = 0.0055245
    DCResistance = 2.8645
    P1 = -5.0 35.0
    P2 = 5.0 35.0
    Sag = 10.0
  }
}
Line Constants Ground Data:
{
  GroundResistivity = 100.0
  GroundPermeability = 1.0
  EarthImpedanceFormula = 0
}
```

- 3. Model Component:** Some important information will appear in the input file from whatever model is being used. In this case, it is the Frequency Dependent (Phase) model:

```
Frequency Dep. (Phase) Model Options:
{
  Interpolate Travel Times = 1
  Infinite Line Length = 0
  Curve Fitting Start Frequency = 0.5
  Curve Fitting End Frequency = 1000000.0
  Maximum # of Poles for Surge Admittance Fit = 20
  Maximum # of Poles for Attenuation Constant Fit = 20
  Maximum Fitting Error (%) for Surge Admittance = 0.2
  Maximum Fitting Error (%) for Attenuation Constant = 2.0
  Weighting Factor 1 = 100.0
  Weighting Factor 2 = 1000.0
  Weighting Factor 3 = 1.0
  Write Detailed Output Files = 0
}
```

Constants (*.tlo/*.clo) File

The constants file is a T-Line/Cable Constants program output file, which is used as an input file for EMTDC. This file contains all the information necessary for EMTDC to represent the transmission system in the time domain, and is for the most part, of no concern to the user.

Log (*.log) File

The log file is a collection of messages sequentially output by the T-Line/Cable Constants program while it runs its course.

This is an important file when initially debugging and tuning a transmission line or cable segment. It is recommended that the user scan through this file to ensure that no problems occurred while the line constants were being solved.

Output File

Output files are created by the T-Line/Cable Constants program and are used to display important transmission system data in a convenient format for the user. This can include impedance and admittance matrix information, sequence data and travel times.

The format of the output file will change slightly depending on the model used. See the Additional Options component for details on formatting data in this file.

Chapter 8: Transmission Lines and Cables

Phase Data

The phase data section displays relevant line parameters at a specific frequency, where all quantities are in the phase domain. The term 'phase domain' refers to quantities that have not been transformed into the modal domain. Phase data represents the 'real life' characteristics of the line.

Series Impedance Matrix (Z):

This data represents the system series impedance matrix \mathbf{Z} per-unit length (Ω/m). The diagonal terms represent the self impedances of each conductor, whereas the off-diagonals are the mutual impedances between the respective conductors. The dimension of the matrix depends on the final number of equivalent conductors/ground wires in the system N :

$$\mathbf{Z} = \begin{bmatrix} Z_{11} & Z_{12} & \cdots & Z_{1N} \\ Z_{21} & Z_{22} & \cdots & Z_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{N1} & Z_{N2} & \cdots & Z_{NN} \end{bmatrix} [\Omega/\text{m}]$$

Series Impedance Matrix

All elements in this matrix are complex and are given in Cartesian format:

$$R_{ij}, X_{ij} \rightarrow Z_{ij} = R_{ij} + j \cdot X_{ij}$$

The positions of elements in this matrix are dependent on the manner in which the conductors/ground wires have been numbered. The type of ideal transposition that has been selected will also affect this matrix.

Shunt Admittance Matrix (Y):

This data represents the system shunt admittance matrix \mathbf{Y} per-unit length (S/m). The diagonal terms represent the self admittances of each conductor, whereas the off-diagonals are the mutual admittances between the respective conductors. The dimension of the matrix depends on the final number of equivalent conductors/ground wires in the system N :

$$Y = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1N} \\ Y_{21} & Y_{22} & \cdots & Y_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{N1} & Y_{N2} & \cdots & Y_{NN} \end{bmatrix} [S/m]$$

Shunt Admittance Matrix

All elements in this matrix are complex and are given in Cartesian format:

$$G_{ij}, B_{ij} \rightarrow Y_{ij} = G_{ij} + j \cdot B_{ij}$$

The positions of elements in this matrix are dependent on the manner in which the conductors/ground wires have been numbered. The type of ideal transposition that has been selected will also affect this matrix.

Long-Line Corrected Series Impedance Matrix (ZLL):

This data represents the long-line corrected, series impedance matrix \mathbf{Z} (Ω). This matrix is the impedance of the entire line length, where all quantities have been passed through a correction algorithm to account for the electrical effects of long line distances.

The long-line corrected quantities have a specific use: They should be used whenever a single PI-section equivalent is being derived to represent the entire line length at a specific frequency. This data should not be used to define time domain traveling wave models.

Long-Line Corrected Shunt Admittance Matrix (YLL):

This data represents the long-line corrected, shunt admittance matrix \mathbf{Y} (S). This matrix is the admittance of the entire line length, where all quantities have been passed through a correction algorithm to account for the electrical effects of long line distances.

The long-line corrected quantities have a specific use: They should be used whenever a single PI-section equivalent is being derived to represent the entire line length at a specific frequency. This data should not be used to define time domain traveling wave models.

Sequence Data

The sequence data section displays relevant line parameters at a specific frequency, where all quantities are sequence quantities. The

Chapter 8: Transmission Lines and Cables

sequence data is calculated directly, through the use of a sequence transform matrix T :

$$T = \frac{1}{2\sqrt{3}} \begin{bmatrix} 2 & 2 & 2 \\ 2 & -1-j\sqrt{3} & -1+j\sqrt{3} \\ 2 & -1+j\sqrt{3} & -1-j\sqrt{3} \end{bmatrix}$$

Transformation Matrix

Sequence Impedance Matrix (Z_{sq}):

This data represents the system sequence impedance matrix Z_{sq} per-unit length (Ω/m). Z_{sq} is derived directly from the series impedance matrix Z and the sequence transform matrix T (both described above) as follows:

$$Z_{sq} = T^{-1} \cdot Z \cdot T$$

If all 3-phase circuits in the Z matrix are ideally transposed, then the sequence impedance matrix Z_{sq} will be a diagonal matrix, where the diagonal terms are the equivalent zero, positive and negative sequence components.

For a single 3-phase, ideally transposed circuit, the sequence impedance matrix appears as follows:

$$Z_{sq} = \begin{bmatrix} Z_0 & 0 & 0 \\ 0 & Z_+ & 0 \\ 0 & 0 & Z_- \end{bmatrix}$$

Sequence Impedance Matrix (Single, Balanced 3-Phase Circuit)

In the case of two, ideally transposed 3-phase circuits, the sequence impedance matrix will appear as shown below:

$$Z_{sq} = \begin{bmatrix} Z_{0f} & 0 & 0 & Z_{0M} & 0 & 0 \\ 0 & Z_{+f} & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{-f} & 0 & 0 & 0 \\ Z_{0M} & 0 & 0 & Z_{02} & 0 & 0 \\ 0 & 0 & 0 & 0 & Z_{+2} & 0 \\ 0 & 0 & 0 & 0 & 0 & Z_{-2} \end{bmatrix}$$

Sequence Impedance Matrix (Two, Balanced 3-Phase Circuits)

Where,

- Z_{0n} = Zero sequence impedance of the nth circuit [Ω/m]
- Z_{+n} = Positive sequence impedance of the nth circuit [Ω/m]
- Z_{-n} = Negative sequence impedance of the nth circuit [Ω/m]
- Z_{0M} = Zero sequence mutual impedance [Ω/m]

Sequence data, of course, only makes sense when 3-phase circuits are considered. The format of this matrix depends heavily on the manner in which the individual circuits in the system are transposed (if at all). For example, if a double-circuit line is transposed so that all six conductors are included in the transposition, a sequence matrix will not be provided.

The positions of elements in this matrix are dependent on the manner in which the conductors/ground wires have been numbered. The type of ideal transposition that has been selected will also affect this matrix.

Sequence Admittance Matrix (Y_{sq}):

This data represents the system sequence admittance matrix \mathbf{Y}_{sq} per-unit length (S/m). \mathbf{Y}_{sq} is derived directly from the shunt admittance matrix \mathbf{Y} and the sequence transform matrix \mathbf{T} (both described above) as follows:

$$\mathbf{Y}_{sq} = \mathbf{T}^{-1} \cdot \mathbf{Y} \cdot \mathbf{T}$$

The same concepts described for the sequence impedance matrix \mathbf{Z}_{sq} , apply to \mathbf{Y}_{sq} .

Load Flow RXB Formatted Data

This data is simply the data given in the Sequence Data section, but reformatted for easy reading. The following diagram shows from where the data is taken:

PSCAD

The detailed output data files are written as columnar formatted ASCII text files, which are placed to the Project Temporary Directory when the line constants are solved.

Creating Detailed Output

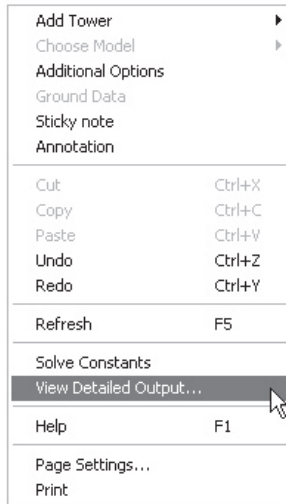
Before any detailed output can be viewed, the transmission line or cable system must be solved by either using the Frequency Dependent (Phase) or Frequency Dependent (Mode) model. Note that in either case, the input parameter **Output Detailed Output Files?** must be selected as **Yes** before the constants are solved.



For more on solving the constants, see Solving the T-Line/Cable Constants Manually.

Invoking the Detailed Output Viewer

The Detailed Output Viewer can be opened from within The T-Line/Cable Properties Editor only. Once in the Editor, right-click on a blank part of the Editor canvas and select **View Detailed Output...** from the menu.



Note that either the Frequency Dependent (Phase) or Frequency Dependent (Mode) model must be present in the Editor. Detailed

Chapter 8: Transmission Lines and Cables

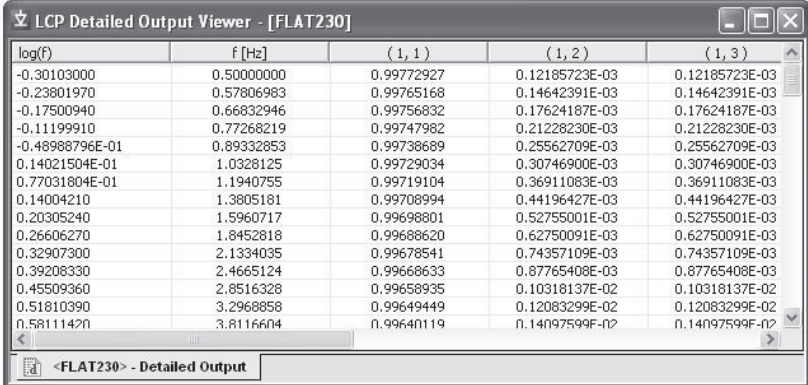
output is not available for the Bergeron model, as this model solves at only one frequency. See the Output (*.out) File for more on viewing Bergeron model output.

The Viewer Environment

The following sections provide a quick overview of the main parts of the Detailed Output Viewer utility.

Spreadsheet Data Viewer

The Detailed Output viewer main page is essentially a navigatable, spreadsheet-like viewer, which is opened as a separate window within the PSCAD environment when the viewer is invoked.



The screenshot shows a window titled "LCP Detailed Output Viewer - [FLAT230]". It contains a spreadsheet with the following data:

log(f)	f [Hz]	(1, 1)	(1, 2)	(1, 3)
-0.30103000	0.50000000	0.99772927	0.12185723E-03	0.12185723E-03
-0.23801970	0.57806983	0.99765168	0.14642391E-03	0.14642391E-03
-0.17500940	0.66832946	0.99756832	0.17624187E-03	0.17624187E-03
-0.11199910	0.77268219	0.99747982	0.21228230E-03	0.21228230E-03
-0.48988796E-01	0.89332853	0.99738689	0.25562709E-03	0.25562709E-03
0.14021504E-01	1.0328125	0.99729034	0.30746900E-03	0.30746900E-03
0.77031804E-01	1.1940755	0.99719104	0.36911083E-03	0.36911083E-03
0.14004210	1.3805181	0.99708994	0.44196427E-03	0.44196427E-03
0.20305240	1.5960717	0.99698801	0.52755001E-03	0.52755001E-03
0.26606270	1.8452818	0.99688620	0.62750091E-03	0.62750091E-03
0.32907300	2.1334035	0.99678541	0.74357109E-03	0.74357109E-03
0.39208330	2.4665124	0.99668633	0.87765408E-03	0.87765408E-03
0.45509360	2.8516328	0.99658935	0.10318137E-02	0.10318137E-02
0.51810390	3.2968858	0.99649449	0.12083299E-02	0.12083299E-02
0.58111420	3.8116604	0.99640119	0.14097599E-02	0.14097599E-02

Each column in the spreadsheet represents an individual quantity (usually a single matrix or vector element) and each row of data represents a calculation at a particular frequency.

Column Organization

Depending on the output parameter being viewed, the data may exist as either matrix or array, or whether curve fitting results are included.

Matrix:

log(f)	f [Hz]	(1, 1)	(1, 2)	(1, 3)	(2, 1)	(2, etc...
--------	--------	--------	--------	--------	--------	------------

Matrix parameters, such as the Series Impedance matrix (Z), are listed in order from left-to-right and top-to-bottom. The column headers are labelled in (row,column) format.

PSCAD

Array:

log(f)	f [Hz]	(1)	(2)	(3)	etc...
--------	--------	-----	-----	-----	--------

Array quantities are simply listed in order.

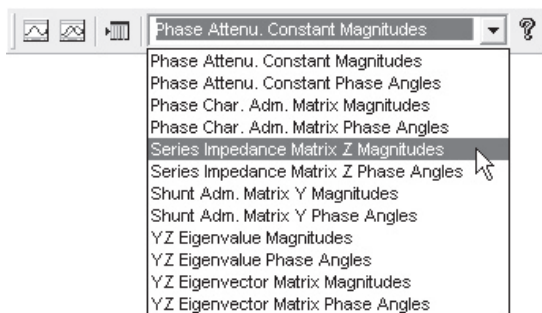
Calculated/Fitted Quantities:

log(f)	f [Hz]	(1)C	(1)F	(2)C	(2)F	(3)C	(3)F	etc...
--------	--------	------	------	------	------	------	------	--------

When curve fitting results are included, the calculated and fitted quantities are staggered, as indicated by the 'C' and 'F' respectively.

Switching between Spreadsheets

Only one parameter may be viewed at a time in spreadsheet form. To switch to another available parameter, click on the parameter selection drop list in the Detailed Output Toolbar:



Copying Data Rows

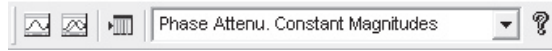
One or more rows of data may be copied to the Windows clipboard. Simply click on a single row, or hold down the **Ctrl** key and select multiple rows. Right-click on the selection and select **Copy**.

log(f)	f [Hz]	(1, 1)	(1, 2)
-0.30103000	0.50000000	0.99772927	0.121
-0.23801970	0.57806983	0.99765168	0.146
-0.17500940	0.66832946	0.99756832	0.176
-0.11199910	0.77268219	0.99747982	0.212
-0.48988796E-01	0.89332853	0.99738689	0.255
0.14021504E-01	1.0328125	0.99729034	0.307
0.77031804E-01	1.1940755	0.99719104	0.369
0.14004210	1.3805181	0.99708994	0.441
0.20305240	1.5960717	0.99698801	0.527
0.26606270	1.8452818	0.99688620	0.627
0.32907300	2.1334035	0.99678541	0.743
0.39208330	2.4615252	0.99668653	0.877
0.45509360	2.8396469	0.99658935	0.103
0.51810390	3.2677686	0.99649449	0.120
0.58111420	3.7458903	0.99640110	0.140




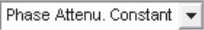

Chapter 8: Transmission Lines and Cables

Detailed Output Toolbar

The Detailed Output Viewer is accompanied by its own toolbar:



The individual toolbar buttons are listed below with a short description:

Button	Description
	View only curves from presently viewed spreadsheet
	View all curves at once
	X-axis settings (i.e. plot vs. frequency or log(f))
	Drop list to select data viewed in spreadsheet
	Access help

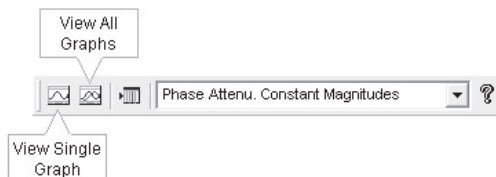
Curve Viewer

The Curve Viewer is a simple utility used for viewing the line constants detailed output data graphically. The utility uses the same online plotting facilities that are provided in the main PSCAD environment, tailored to this specific use. There are two Curve View utilities: View Single Graph and View All Graphs.

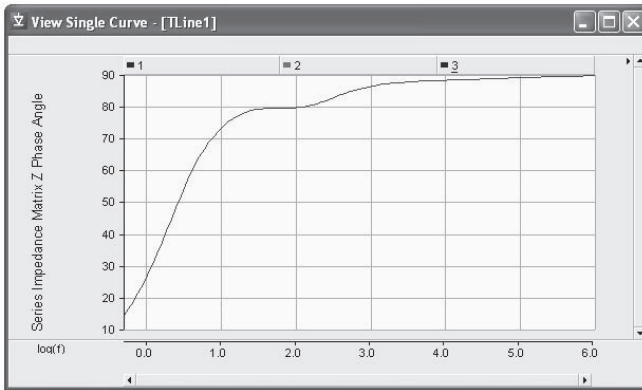
NOTE: For detailed information on using these plotting tools, see Online Plotting.

Invoking the Curve Viewer

To invoke the Curve Viewers, press either the **View Single Graph** or **View All Graphs** button in the Detailed Output toolbar:



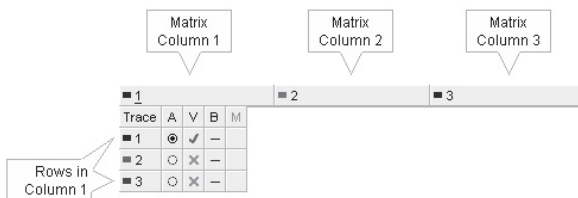
This will bring up the Curve Viewer as a floating window, as shown below for the Single Graph Viewer:



When viewing matrix quantities like the Series Impedance (Z) above, only the diagonal elements of the matrix are initially enabled. This is mainly to reduce the number of possible Traces.

Element Identification

The Curve Viewer makes use of the multi-trace Curve feature available in the PSCAD online plotting tools, in order to allow plotting of 2-dimensional parameters, such as matrices. For example, if plotting a 3x3 matrix, such as the Y or Z matrix magnitudes, the matrix elements are organized according to Curve and Trace, where each multi-trace Curve represents an entire matrix column, and the respective Traces represent the elements of that column.

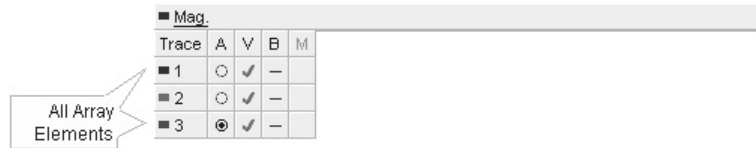


A 3x3 Matrix

In the diagram above, the column 1 Curve has been expanded to reveal the three Traces representing the 3-elements of the column. That is, elements (1,1), (2,1) and (3,1). For more on multi-trace Curves, see Curves and Traces.

Chapter 8: Transmission Lines and Cables

Array items, such as the eigenvalue array, are organized such that all elements are included within a single Curve. For example, the magnitudes of a 3-element array would appear as follows:



■ Mag.				
Trace	A	V	B	M
■ 1	○	✓	—	
■ 2	○	✓	—	
■ 3	⊙	✓	—	

A 3-Element Array

Designing Components

One of the features that makes PSCAD such a powerful simulation tool is its allowance for the design of custom models. Users can develop models from the very simple, to the very complex, limited only by their skills and knowledge of the subject.

In order for a custom model to be included in either the System Dynamics or the Electric Network Solution in EMTDC, the model must first be added as a component in PSCAD. Components act as a graphical representation of models, allowing the user to supply input parameters, perform pre-calculations on input data, and change the component appearance.

This topic discusses the various features and tools available to the user for the design of custom components in PSCAD. The information here is closely linked with the following topic entitled PSCAD Script. It is suggested that you become familiar with both chapters before proceeding with your component design.

The last section of this topic includes a tutorial called Creating a New Component, which goes through many of the features described below.

NOTE: See 'Creating a New Component or Module' in Chapter 5 for details on creating a new component or module, using the Component Wizard.

THE DESIGN EDITOR

In PSCAD V2, user components were designed and edited by way of a text file. In PSCAD V3, component design was accomplished by using a utility entitled the Component Workshop (or CWS). The Component Workshop was invoked by editing the Component Definition and contained different sections for the design of graphics, dialog windows and code sections. In PSCAD V4, components are designed in a very similar fashion to V3, but without the use of

Chapter 9: Designing Components

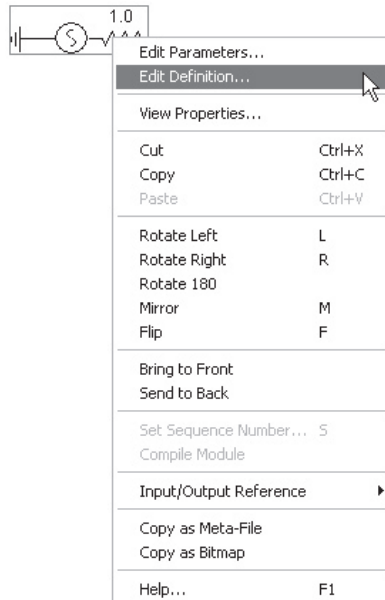
the Component Workshop as a separate utility. Now, the user can conveniently edit Component Definitions directly within the Design Editor.

The Design Editor includes a tab bar at the bottom of its window, which includes tabs for the convenient access of information pertinent to the project at hand. Three of the tabs represent the three main sections of the Component Definition, and will become enabled when the user decides to edit a particular Component Definition. These are:

- Graphic
- Parameters
- Script

EDITING A COMPONENT OR MODULE DEFINITION

To edit the definition of any component or module (i.e. enable the relevant tabs), right-click over the component and select **Edit Definition...** from the pop-up menu. As an alternative, press **Ctrl + double left click**, while the mouse pointer is over the component.



NOTE: You cannot edit the definition of Master Library components, unless you rename the 'master.psl' file.

RETURNING TO CIRCUIT VIEW

To exit any section of the Component Definition, click the **Return to Circuit View** button in the toolbar corresponding to that section. For example, use the Graphic Palette while in the Graphic section. While in the Graphic section only, pressing the **Backspace** key will also exit back to Circuit view.



THE GRAPHIC SECTION

The component graphic is important as it represents your model visually to you and other users in PSCAD. The Graphic section is the design environment, in which component Graphic Objects, Text Labels and Connections can be added, removed and manipulated. Also, the general appearance and functionality of the component can be altered according to specified conditional statements.

The Graphic canvas is always the default window to open upon entering the Component Definition for editing. If you are not sure where you are in the Design Editor, simply look at the horizontal tab bar at the bottom of the Design Editor window for clarification. If the Graphic section is enabled, then the tab bar should appear as below:



Navigating and Zooming

There are many navigational features available to help you efficiently navigate about the Graphic canvas.

Chapter 9: Designing Components

Scroll Bars

Standard vertical and horizontal scroll bars are available in all Design Editor windows. These are located at the right-most and bottom-most edges of the open window respectively.



Arrow Keys

You can use the arrow buttons on your keyboard to scroll both horizontally and vertically while in the Graphics section.

Panning (Dynamic Scroll) Mode

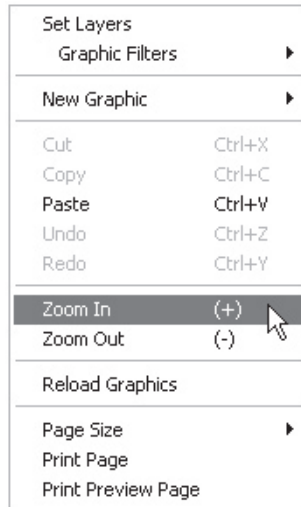
The panning or dynamic scroll feature allows you to scroll through the Graphic section in a fluid motion. You can invoke panning mode by one of the following methods:

- On a blank portion of the page, press the **Ctrl** and **Shift** keys, then click and hold the left mouse button (**Ctrl + Shift + left mouse hold**). Moving the mouse will then allow panning through the page.
- Press the **Pan** button in the Main Toolbar to invoke panning mode. To indicate that you are in panning mode, the mouse pointer will change into a hand shape, as shown to the right. Press Esc to cancel Pan Mode.

Zooming

Zoom features are available when working in the Graphic section. There are a few different methods for zooming available:

- From the Main Menu, select **View | Zoom**. You then have a choice to select either **In**, **Out** or a specified zoom percentage.
- From the Main Toolbar, select either the **Zoom In** or **Zoom Out** buttons, or select a percentage zoom directly from the **Zoom In/Out** drop down list.
- Right-click on a blank part of the page and select **Zoom In** or **Zoom Out**.



Graphic Objects

There are several types of graphic objects available to help design a visually appealing component. These include:

- $\frac{1}{4}$ Arc
- $\frac{1}{2}$ Arc
- Ellipse
- Line
- Rectangle



Object properties, such as colour, fill and size can be changed to suit the component graphical needs. A good graphical design can enhance user understanding of the purpose of the component and what its primary function is.

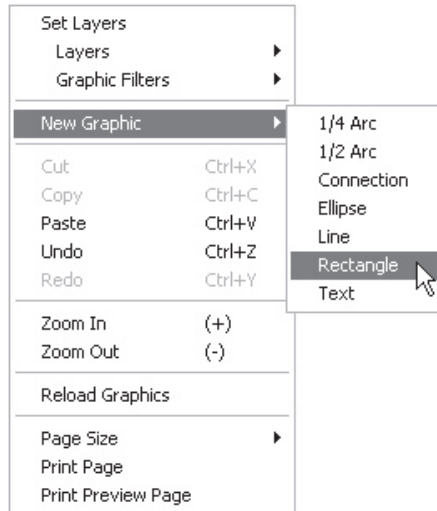
Adding Graphic Objects

To add a graphic object to your component definition, the most straightforward method is to use the Graphic Palette:



Chapter 9: Designing Components

Another method is to use the right-click menu: Move the mouse pointer over a blank area of the Graphic window. Right-click and select **New Graphic**.



Depending on the object selected, a new object should appear attached to your mouse pointer. With your mouse, move the object to the desired location within the Design Window and left-click to place the object.

Rotate, Flip, Mirror and Resize Graphic Objects

Graphic objects can be rotated, flipped, mirrored or resized once they have been placed in the Graphic window. To resize, left-click on the object so that Grips appear.



Place the mouse pointer over a selected Grip, press and hold the left mouse button, and move the mouse. Note that the corner Grips will

PSCAD

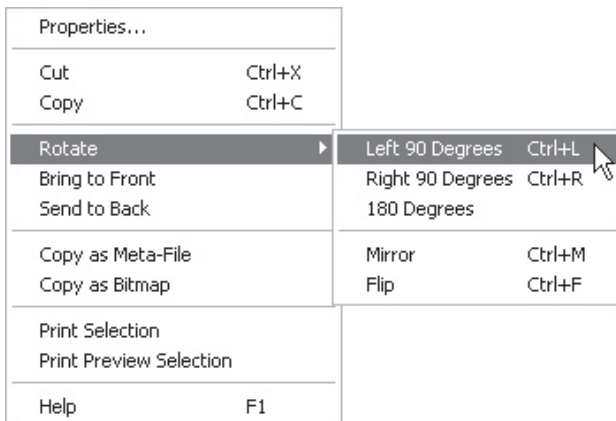
allow resizing in both directions, while the mid-point grips will only allow movement in either the horizontal or vertical directions.

Rotate, flip or mirror can be accomplished in one of three ways. The first is to use the speed buttons on the Rotation Bar.



If you cannot see this toolbar, go to the Main Menu bar and select **View | Rotation Bar**. Select the object with a left-click of your mouse. Then, click one of the four buttons shown above.

You can also use the right-click pop-up menus: Right-click over the object and select **Rotate**. Choose one of the options given.

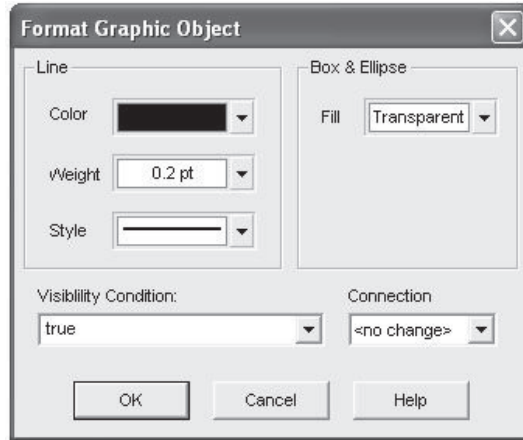


As an alternative to the right-click menu and toolbar, you can also use keyboard shortcuts **Ctrl + r**, **Ctrl + f** and **Ctrl + m** (or simply **r**, **f** and **m**) for rotate, flip and mirror respectively. Make sure that the mouse pointer is over top of the object before using these keys. If you select the object first, then the object will rotate, flip and mirror centred on the object itself.

Changing Graphic Object Properties

Graphic Object properties (except the Arc) can be adjusted through the **Format Graphic Object** dialog window. To change the object properties, either **left double-click** or right-click over the object and select **Properties....**

Chapter 9: Designing Components



As shown above, graphical features, such as styles and colours, can be adjusted.

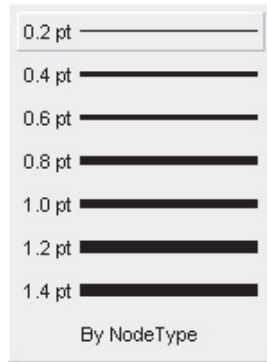
Line:

- **Colour:** Select the down arrow to bring up the colour palette shown below. Left-click on a colour button to change the colour of the object line. If the object is a Rectangle or Ellipse, this will change the border colour. If **By Node Type** is selected, the type of node to which the object is associated with will determine the line colour. See the Connection input below for more details.

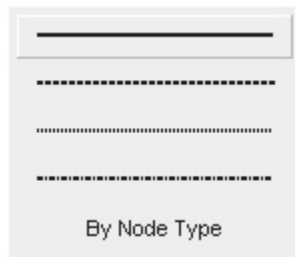


- **Weight:** Select the down arrow to bring up the line weight palette shown below. Left-click on a weight button to change the weight or thickness of the object line. If the object is a Rectangle or Ellipse, this will change the border weight. If **By Node Type** is selected, the type of node to which the

object is associated with will determine the line weight. See the Connection input below for more details.



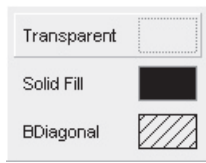
- **Style:** Select the down arrow to bring up the line style palette shown below. Left-click on a style button to change the style of the object line. If the object is a Rectangle or Ellipse, this will change the border style. If **By Node Type** is selected, the type of node to which the object is associated with will determine the line style. See the Connection input below for more details.



Box & Ellipse:

- **Fill:** Select the down arrow to bring up the fill palette shown below. With the mouse pointer over either the **Solid Fill** or **BDiagonal** buttons, left click the down arrow to bring up the respective sub-palettes shown below. This input is disabled for Line objects!

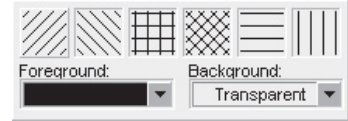
Chapter 9: Designing Components



Fill Palette



Solid Fill Sub-Palette



BDiagonal Sub-Palette

Other:

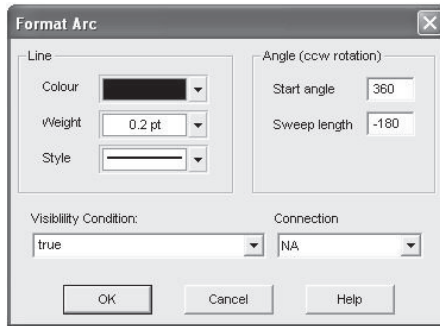
- **Visibility Condition:** Enter a conditional statement to determine under what input conditions the object is to be visible. See 'Conditional Statements, Layers & Filters' in this chapter for more details.
- **Connection:** Enter the name of a Connection to which this object is to be associated. This input is used to associate graphical properties, such as colour, thickness and style, with a particular type of Connection, to which it is connected. For example, if the node indicated in this field happens to be a single-line, electrical node (i.e. dimension of 3), then the line weight will become greater when the project is compiled. This method is used extensively in the Master Library models.

In addition to the Format Graphic Object dialog, you can change the line/border colour, weight and style, as well as fill properties directly from the Graphic Palette. First, select the object with a left-click (Grips appear). Then adjust any of the above properties by using the appropriate buttons.



Changing Arc Object Properties

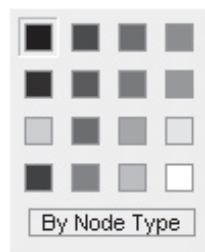
The Arc is a special type of graphical object. Arc object properties can be adjusted through the **Format Arc** dialog window. To change the Arc properties: Right-click over the Arc object (without selecting it) and select **Properties....**



As shown above, graphical features, such as styles and colours, can be adjusted.

Line:

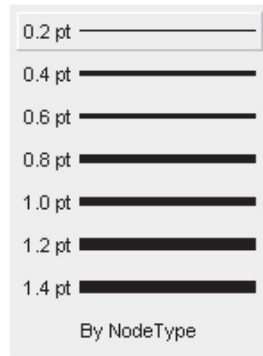
- **Colour:** Select the down arrow to bring up the colour palette shown below. Left-click on a Colour button to change the colour of the arc line. If **By Node Type** is selected, the type of node to which the arc is associated with will determine the line colour. See the Connection input below for more details.



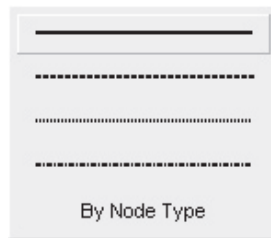
NOTE: For convenience, there is a quarter arc and half arc available in the Graphic Palette and the right-click menu.

- **Weight:** Select the down arrow to bring up the line weight palette shown below. Left-click on a weight button to change the weight or thickness of the arc line. If **By Node Type** is selected, the type of node to which the arc is associated with will determine the line weight. See the Connection input below for more details.

Chapter 9: Designing Components



- **Style:** Select the down arrow to bring up the line style palette shown below. Left-click on a style button to change the style of the arc line. If **By Node Type** is selected, the type of node to which the arc is associated with will determine the line style. See the Connection input below for more details.



Angle (ccw rotation):

- **Start Angle:** Enter the angle in degrees at which the arc has its starting point. This angle is based on a standard, four-quadrant system.
- **Sweep Length:** Enter the angular distance through which the arc is to sweep. Sweep distance is always in the counter-clockwise direction.

Other:

- **Visibility Connection:** Enter a conditional statement to determine under what input conditions the arc is to be

visible. See 'Conditional Statements, Layers & Filters' in this chapter for more details.

- **Connection:** Enter the name of a Connection to which this arc is to be associated. This input is used to associate graphical properties, such as colour, thickness and style, with a particular type of Connection, to which it is connected. For example, if the node indicated in this field happens to be a single-line, electrical node (i.e. dimension of 3), then the arc line weight will become greater when the project is compiled. This method is used extensively in the Master Library models.

In addition to the Format Arc Object dialog, you can change the line/border colour, weight and style from the Graphic Palette. First, select the arc with a left-click (Grips appear). Then adjust any of the above properties by using the appropriate buttons.



Cut, Copy, Paste and Delete Objects

You can cut, copy or paste any graphic object by using one of three methods:

- Select on the object with a left mouse click so that Grips appear. Click the **Cut** or **Copy** buttons from the Main Toolbar. Click the **Paste** button Main Toolbar to paste.
- Right-click on the object and select **Cut** or **Copy** from the pop-up menu. Right-click over a blank area of the Graphic window and select **Paste** from the pop-up menu.
- Select on the object with a left mouse click so that Grips appear. Press **Ctrl + x** or **Ctrl + c** to cut or copy the object respectively. Press **Ctrl + v** to paste.

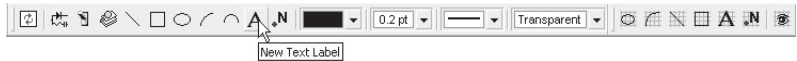
Text Labels

Text can be displayed on component graphics by using a Text Label object. Text labels can be one line of text only, and the user may select alignment and size.

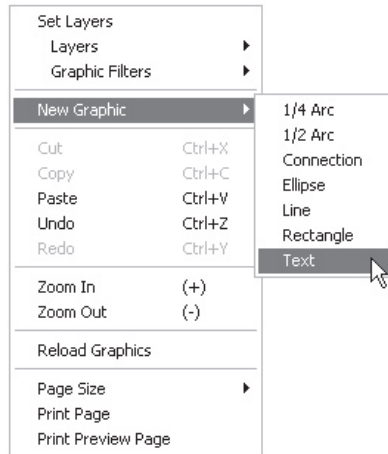
Adding Text Labels

To add a Text Label object to your component definition, the most straightforward method is to use the Graphic Palette:

Chapter 9: Designing Components



Simply left-click the **New Text Label** button on the toolbar, drag the label to where you want it placed and left-click again. Another method is to use the right-click menu: Move the mouse pointer over a blank area of the Graphic window. Right-click and select **New Graphic | Text**.



A Text Label should appear attached to your mouse pointer. With your mouse, move the label to the desired location within the Graphic window and left-click to place the object.

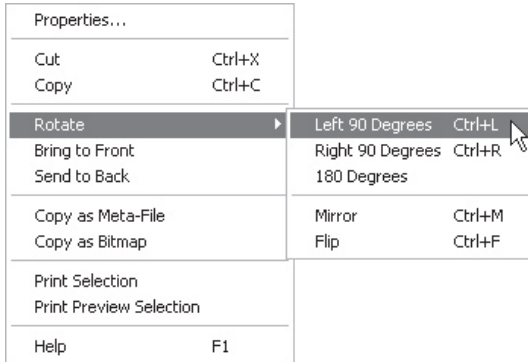
Rotate, Flip or Mirror Text Labels

Text labels can be rotated, flipped or mirrored once they have been placed in the Graphic window. This can be accomplished in one of three ways: The first is to use the speed buttons on the Rotation Bar:



If you cannot see this toolbar, go to the Main Menu bar and select **View | Rotation Bar**. Select the object with a left-click of your mouse. Then, click one of the four buttons shown above.

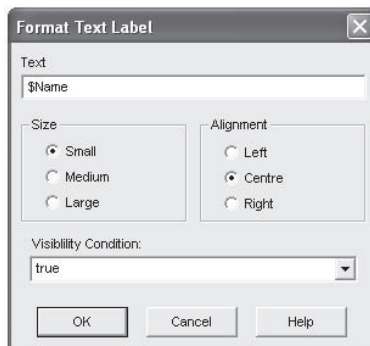
You can also use the right-click menus: Select the label with a left-click of your mouse. Right-click over the selected object and select **Rotate**. Choose one of the options given.



As an alternative to the right-click menu and toolbar, you can also use keyboard shortcuts **r**, **f** and **m** for rotate, flip and mirror respectively. Make sure that the mouse pointer is over top of the label before using these keys. If you select the label first, then the label will rotate, flip and mirror centred on the label itself.

Changing Text Label Properties

Text label properties can be adjusted through the **Format Text Label** dialog window. To change the Text Label properties, either **left double-click** or right-click over the Text Label and select **Properties...**



As shown above, text features, such as size and style, can be adjusted.

Chapter 9: Designing Components

- **Text:** Enter the text you wish to appear on the label.
- **Size:** Select **Small**, **Medium** or **Large** sizes for your text.
- **Alignment:** Select **Left**, **Centre** or **Right** justification for your text.
- **Visibility Condition:** Enter a conditional statement to determine under what input conditions the object is to be visible. See 'Conditional Statements, Layers & Filters' in this chapter for more details.

Cut, Copy, Paste and Delete Labels

You can cut, copy or paste a Text Label by using one of three methods:

- Select on the Label with a left mouse click. Click the **Cut** or **Copy** buttons from the Main Toolbar. Click the **Paste** button Main Toolbar to paste.
- Right-click on the Label and select **Cut** or **Copy** from the pop-up menu. Right-click over a blank area of the Graphic window and select **Paste** from the pop-up menu.
- Select on the Text Label with a left mouse click so that Grips appear. Press **Ctrl + x** or **Ctrl + c** to cut or copy the label respectively. Press **Ctrl + v** to paste.

Linking a Text Label to an Input Field

It is possible to link a Text Label text to a specific input parameter within the same definition. Once linked, the Text Label can be used to graphically display the parameter value (and unit). For example, you can display the MVA rating of a transformer model according to what is entered in its 'Rated MVA' input field.

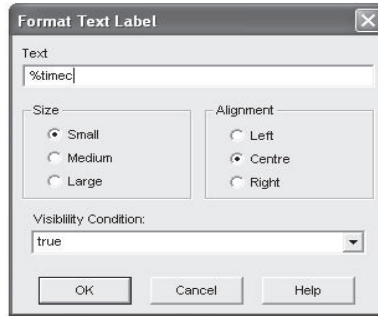
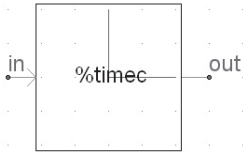
Linking is accomplished by simply entering the 'Symbol' name of an associated input parameter into the text label, preceded by either a percent (%) or a dollar (\$) symbol. If the \$ prefix is used, just the value of the linked input field will be displayed. If the % prefix is used, both the value and the specified unit will be displayed. The following example illustrates how to link to an input field. See 'The Parameters Section' in this chapter for more details on parameter fields.

EXAMPLE 9-1:

A user wants to display the value of an Input Field with Symbol name 'timec,' which represents the component time constant. The user

PSCAD

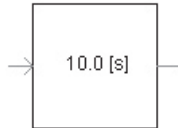
adds a Text Label to the component graphic and adds the following to the **Text** input field in the **Format Text Label** dialog window:



Component Graphic section

Text Label Property Settings

If the 'timec' input field has a value of say **10.0 [s]**, then the resulting display on the Circuit canvas would be similar to:



NOTE: If the % prefix above is replaced by the \$ prefix, the units will not be displayed.

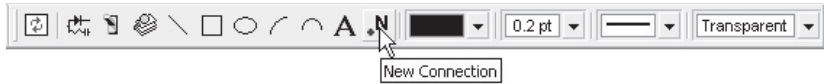
Connections

Connections are used to provide an interface to or from your component and the external system (as drawn on the Circuit canvas). They provide a means to either read data from, or output data to the external system each simulation time step. These nodes play an essential part in the construction of circuits in PSCAD, and are the graphical signal communication avenue between models.

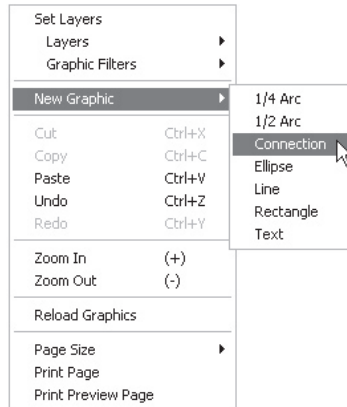
Adding Connections

To add a Connection object to your component definition, the most straightforward method is to use the Graphic Palette:

Chapter 9: Designing Components



Simply left-click the **New Connection** button on the toolbar, drag the connection to where you want it placed and left-click again. Another method is to use the right-click menu: Move the mouse pointer over a blank area of the Graphic window. Right-click and select **New Graphic | Connection**.

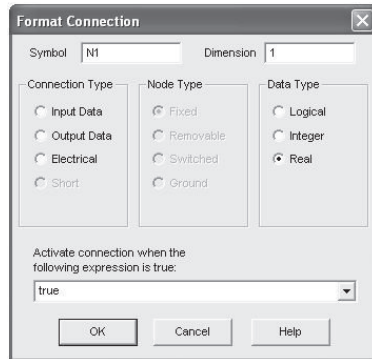


NOTE: You may notice that Connections are always snapped to the Graphic canvas drawing grid. This is to ensure that when other components are connected to yours, their respective connection points will overlap.

A Connection should appear attached to your mouse pointer. With your mouse, move the node to the desired location within the Graphic window and left-click to place the object.

Changing Connection Properties

Connection properties can be adjusted through the **Format Connection** dialog window. To change the Connection properties: Either **left double-click** or right-click over the Connection and select **Properties....**



As shown above, Connection features, such as name and type, can be adjusted.

- **Symbol:** Enter a name for the Connection. Note that this name must be compatible with standard Fortran naming conventions (i.e. it must begin with a non-numeric character, do not include spaces, etc.).
- **Dimension:** If this Connection is to carry a signal that is defined as a standard Fortran array, then this is the dimension of the array signal. For example, if Connection 'N1' is to be defined as REAL N1(3), then this field should be specified as 3. You may also substitute the Symbol Name of an integer Input Field or Choice Box in this field. The connection will assume a dimension equal to this value upon compilation.
- **Connection Type:** Select **Input Data**, **Output Data** or **Electrical Node**. If this connection is to be part of the EMTDC System Dynamics (i.e. a control signal), then you must either choose Input or Output Data (an input data signal is that which is coming into or being accepted by your component). Only select Electrical Node if this connection is to be part of an electrical circuit.
- **Node Type:** Select **Fixed**, **Removable**, **Switched** or **Ground**. This parameter is only enabled if the Connection Type is an Electrical Node (see the following Electrical Node Types section).
- **Data Type:** Select **Integer** or **Real**. This input defines the type of data signal that will be passing through this connection and is based on the standard Fortran INTEGER

Chapter 9: Designing Components

or REAL declarations. It is only enabled if Connection Type is Input or Output Data.

- **Activate connection when the following expression is true:** Enter a conditional statement to determine under what input conditions the connection is to be enabled. See ‘Conditional Statements, Layers & Filters’ in this chapter for more details.

Cut, Copy, Paste and Delete Connections

You can cut, copy or paste a Connection by using one of three methods:

- Select on the Connection with a left mouse click. Click the **Cut** or **Copy** buttons from the Main Toolbar. Click the **Paste** button Main Toolbar to paste.
- Right-click on the Connection and select **Cut** or **Copy** from the pop-up menu. Right-click over a blank area of the Graphic window and select **Paste** from the pop-up menu.
- Select on the Connection with a left mouse click so that Grips appear. Press **Ctrl + x** or **Ctrl + c** to cut or copy the label respectively. Press **Ctrl + v** to paste.

Electrical Node Types

When the connection is selected as an electrical node, there are four electrical node types available to the user. These are described below:

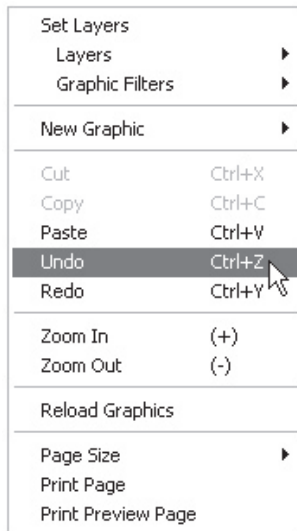
- **Fixed:** A fixed node is the most common type of electrical node and should be the default chosen when in doubt. It represents a simple electrical node.
- **Removable:** A removable node is that which may be ‘removed’ by PSCAD, if it is to be part of a collapsible branch. For example, a branch with separate series RLC elements can be collapsed by PSCAD into an equivalent, single element impedance branch (Z) (effectively removing two extra nodes). Select **Removable** if you would like to take advantage of this feature.
- **Switched:** If your node is to be part of a frequently switching branch, that is a branch whose equivalent conductance is changing many times during a simulation (thyristor, GTO, etc.), then this option should be chosen. Switched nodes are included in the Optimal Node Ordering algorithm, which makes matrix decomposition more efficient, and thereby speeds up the simulation.

PSCAD

- **Ground:** Select this option if your node is to be a ground node.

Undo and Redo

To undo or redo any changes in the Graphics section, select the **Undo** or **Redo** buttons in the Main Toolbar, press **Ctrl + z** or **Ctrl + y**, or right-click over a blank part of the Graphics window and select **Undo** or **Redo**.



Saving and Reloading Graphics

Any changes made to your component graphics should be saved regularly. To save changes, simply exit the component definition (i.e. go back to the Circuit window). To do this:

1. Either **left double-click** the project name in the Workspace window, or click the **Return to Circuit View** button in the Graphic Palette.



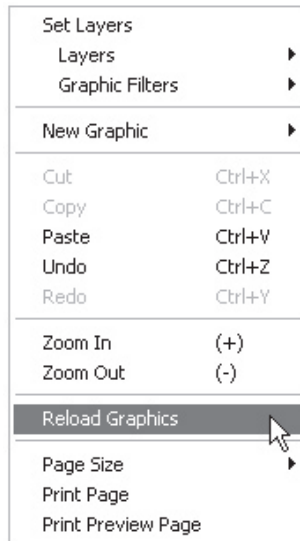
2. Save the Project by right-clicking on the project name in the Workspace and select **Save**.

Chapter 9: Designing Components

If desired, you can at any time reload the graphics from the time at which they were last saved, by pressing the **Reload Graphics** button.



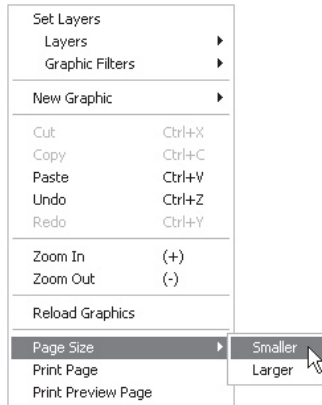
Another method is to use the right-click menu: Move the mouse pointer over a blank area of the Graphics window. Right-click and select **Reload Graphics**.



Caution should be exercised with this function, as once the graphics are reloaded, **all changes will be lost!**

Adjusting Graphic Page Size

In order to accommodate very large component graphics (usually imported from PSCAD V2 or V3), an option to change the Graphic canvas size is given. Move the mouse pointer over a blank area of the Graphic window. Right-click and select **Page Size**.



The default canvas size is **Smaller**.

THE PARAMETERS SECTION

The Parameters section is used to design the primary interface between the component and the user. This is accomplished through the use of programmable dialog windows known as Categories, where users may input everything from model parameters to model conditions and appearance. Category windows are the only sections accessible to users once the component design is completed.

If you are not sure where you are in the Design Editor, simply look at the horizontal tab bar below for an indication. If the Parameters section is not enabled, click on the 'Parameters' tab as shown below:



Categories

Within the Parameters section, Categories are ordered similar to pages in a book. You may include several pages, each containing inputs pertaining to a similar function, or you may place all inputs in a single Category page. There are three different types of input fields that may be added to Category pages. These include:

- Text Field
- Input Field
- Choice Box

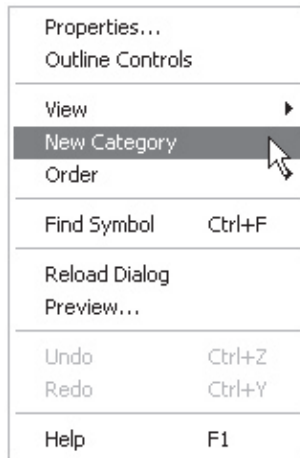
Chapter 9: Designing Components

Adding a New Category

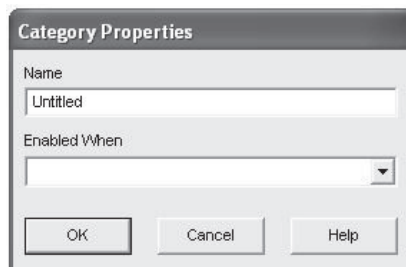
To add a new Category page to your Component Definition, the most straightforward method is to use the Parameters Bar:



Left click the **New Category** button. Another method is to use the right-click menu: Move the mouse pointer over a blank area (white part) of the Parameters window. Right-click and select **New Category**.



In either case, the Category Properties dialog should appear.



Select a descriptive name for the Category page in the **Name** field. If desired, you can add a conditional statement in the **Enabled**

PSCAD

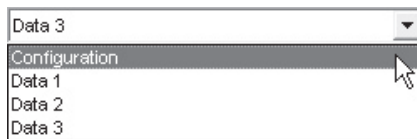
When input. Click the **OK** button. A new Category page should then appear within the Parameters section viewing window.



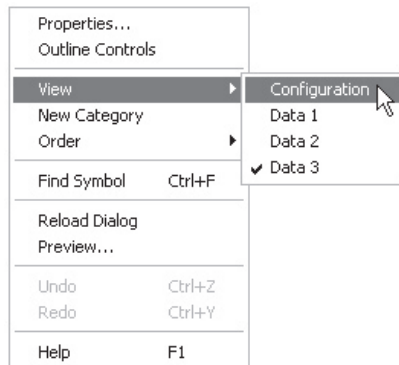
You can edit the parameters of the Category Properties dialog at any time. See 'Changing Category Properties' below for more details.

Viewing Categories

As mentioned above, categories are organized similar to pages in a book. Once you have added more than one category, there may be a need to navigate through these pages. The most straightforward method is to use the Parameters Bar. Simply left-click the down arrow on the **View Category** drop list as shown below:



Another method is to use the right-click menu: Move the mouse pointer over a blank area (white part) of the Parameters window. Right-click and select **View Category**.



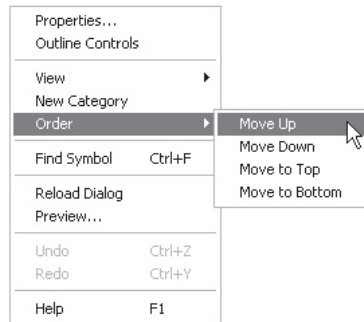
Chapter 9: Designing Components

Ordering Categories

Once you have added more than one category, there may be a need to re-order the sequence of the category pages. The most straightforward method is to use the Parameters Bar. Simply left-click either the **Move Category Up** or the **Move Category Down** buttons. To move a certain Category directly to the top or bottom of the list, select either the **Move Category To Top** or the **Move Category To Bottom** buttons.



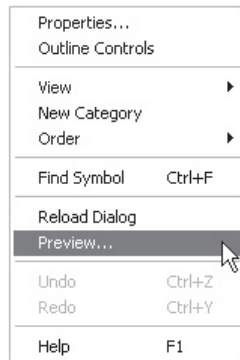
Another method is to use the right-click menu: Move the mouse pointer over a blank area (white part) of the Parameters window. Right-click and select **Order**. Choose one of the given options from the sub-menu.



Previewing Categories

For convenience, a preview function exists so that look and feel of the category pages can be viewed, as they would appear to someone using the component - without leaving the Component Definition. The most straightforward method is to use the **Parameters Bar**. Simply left click **Preview Dialog** button.

Another method is to use the right-click menu: Move the mouse pointer over a blank area (white part) of the Parameters window. Right-click and select **Preview....**

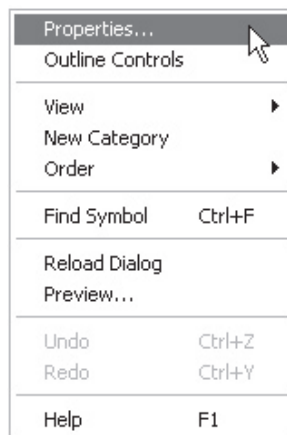


Changing Category Properties

Category properties can be adjusted through the **Category Properties** dialog window. The most straightforward method is to use the **Parameters Bar**. Simply left click **Category Properties** button.

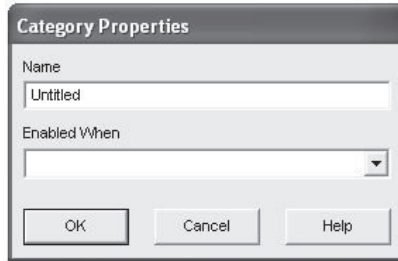


Another method is to use the right-click menu: Move the mouse pointer over a blank area (white part) of the Parameters window. Right-click and select **Properties....**



Chapter 9: Designing Components

Either of these operations should bring up the Category Properties dialog:



- **Name:** Enter a descriptive name for display on the Category page.
- **Enabled When:** Enter a conditional statement to determine under what input conditions the Category page is to be enabled. If a Category is disabled, the user will still be able to view the contents, but all input fields will be disabled. See 'Conditional Statements, Layers & Filters' in this chapter for more details.

Duplicating a Category

The ability to duplicate a Category (including all of its contents) is essential when designing components. A direct duplication will ensure that all Input Fields, Text Fields and Choice Boxes are aligned in the same manner as the copied Category. This saves time in designing the Category layout. The most straightforward method to duplicate a Category is to use the Parameters Bar.



Left click the **Duplicate Category** button. A new Category named '<CategoryName_1>' will appear at the bottom of the **View Category** drop list.

Deleting a Category

To delete a particular Category page, you must first ensure that you are in the Parameters section of the Design Editor and have the Segment open in the Script window. The most straightforward method is to use the Parameters Bar.



Left click the **Delete Category** button.

Text Field

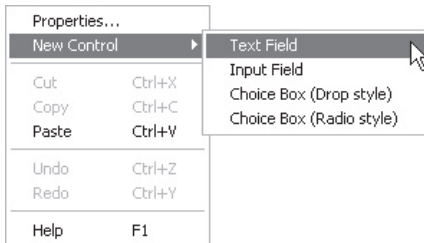
Text fields are used primarily to add descriptive comments as input, or to define signals to be used as internal output variables within the component. See Internal Output Variables for more details.

Adding Text Fields

To add a Text Field to your component definition, the most straightforward method is to use the Parameters Bar:



Left click the **New Text Field** button - a Text Field should appear within the Category page. Another method is to use the right-click menu: Move the mouse pointer over a blank area of the Category page. Right-click and select **New Control | Text Field**.



The new Text Field should appear somewhere within the Category page.

Moving and Resizing Text Fields

To move the new Text Field, left-click over top of the field to select it (Grips appear).



Chapter 9: Designing Components

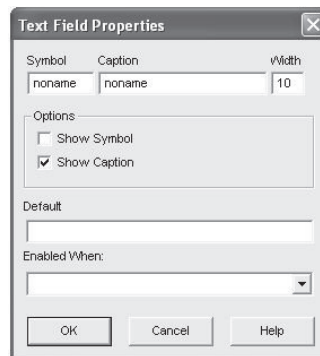
The Text Field can now be moved in one of two ways:

1. Click again and **hold** down the **left mouse button**. Move your mouse around.
2. Use the arrow keys on your keyboard. Press and hold an arrow key to move the field in large increments. Tap the arrow key to move the field in small increments.

To resize the new Text Field, left-click over top of the field to select it (Grips appear). Select a Grip, hold down the left button and move in an outwards direction.

Changing Text Field Properties

Text Field properties can be adjusted through the **Format Text Label** dialog window. To change the Text Field properties, right-click over the Text Field and select **Properties....**



As shown above, features, such as caption and width, can be adjusted.

- **Symbol:** Enter a name for the Text Field. Note that this name must be compatible with standard Fortran naming conventions (i.e. it must begin with a non-numeric character, do not include spaces, etc.).
- **Caption:** Enter text to describe the Text Field. What is entered here will be displayed in the Category page directly to the left of the Text Field box.
- **Width:** Select the character width of the Text Field box itself (i.e. white box).
- **Default:** Use this field to add text, which will show up as default text in the Text Field box.

PSCAD

- **Enabled When:** Enter a conditional statement to determine under what input conditions the Text Field is to be enabled. See ‘Conditional Statements, Layers & Filters’ in this chapter for more details.

Options:

- **Show Symbol:** Select this to display the contents of **Symbol** directly to the left of the Text Field box.
- **Show Caption:** Select this to display the contents of **Caption** directly to the left of the Text Field box.

Cut, Copy, Paste and Delete Text Fields

You can cut, copy or paste a Text Field by using one of three methods:

- Select on the Text Field with a left mouse click so that Grips appear. Click the **Cut** or **Copy** buttons from the Main Toolbar. Click the **Paste** button Main Toolbar to paste.
- Right-click on the Text Field and select **Cut** or **Copy** from the pop-up menu. Right-click over a blank area of the Category page and select **Paste** from the pop-up menu.
- Select on the Text Field with a left mouse click so that Grips appear. Press **Ctrl + x** or **Ctrl + c** to cut or copy the label respectively. Press **Ctrl + v** to paste.

Input Field

Input Fields allow the user to add numerical input to the component. They can also be used (instead of an external input connection) to import a defined signal from elsewhere in the system directly as an online input to the component (i.e. for signal feedback).

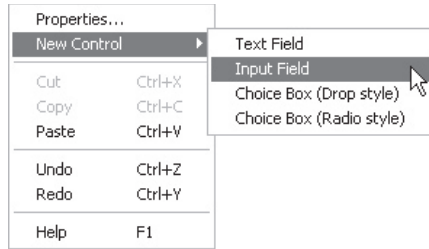
Adding Input Fields

To add an Input Field to your component definition, the most straightforward method is to use the Parameters Bar:



Left click the **New Input Field** button - an Input Field should appear within the Category page. Another method is to use the right-click menu: Move the mouse pointer over a blank area of the Category page. Right-click and select **New Control | Input Field**.

Chapter 9: Designing Components



The new Input Field should appear somewhere within the Category page.

Moving and Resizing Input Fields

To move the new Input Field, left-click over top of the field to select it (Grips appear).



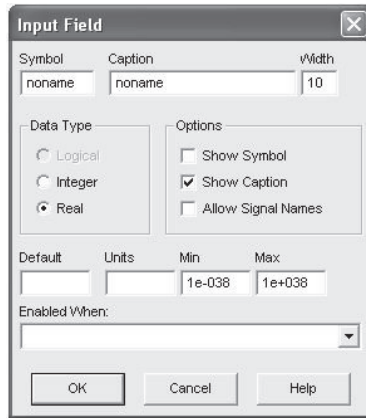
The Input Field can now be moved in one of two ways:

1. Click again and **hold** down the **left mouse button**. Move your mouse around.
2. Use the arrow keys on your keyboard. Press and hold an arrow key to move the field in large increments. Tap the arrow key to move the field in small increments.

To resize the new Input Field, left-click over top of the field to select it (Grips appear). Select a Grip, hold down the left button and move in an outwards direction.

Changing Input Field Properties

Input Field properties can be adjusted through the **Format Input Field** dialog window. To change the Input Field properties, right-click over the Input Field and select **Properties...**



As shown above, features, such as name and width, can be adjusted.

- **Symbol:** Enter a name for the Input Field. Note that this name must be compatible with standard Fortran naming conventions (i.e. it must begin with a non-numeric character, do not include spaces, etc.).
- **Caption:** Enter text to describe the Input Field. What is entered here will be displayed in the Category page directly to the left of the Input Field box.
- **Width:** Select the character width of the Input Field box itself (i.e. white box).
- **Data Type:** Select **Integer** or **Real**. This input defines the type of input variable and is based on the standard Fortran INTEGER or REAL declarations.
- **Default:** Use this field to add a value, which will show up as the default value in the Input Field box. Assume that a user who is not familiar with this component might simply accept your default value for lack of a better value. So, do your best to enter an appropriate value.
- **Units:** Add a unit (if any) to represent the Target Unit related to this Field. See 'Unit System' in Chapter 5 for more details.
- **Min / Max:** Enter the maximum and minimum value limits. If you are unsure of what these limits should be, or limits are not required, simply enter $-1e-038$ and $1e+038$ respectively. If a user enters a value outside this range, PSCAD will provide a warning in the Output Window.
- **Enabled When:** Enter a conditional statement to determine under what input conditions the Input Field is to be

Chapter 9: Designing Components

enabled. See 'Conditional Statements, Layers & Filters' in this chapter for more details.

Options:

- **Show Symbol:** Select this to display the contents of **Symbol** directly to the left of the Input Field box.
- **Show Caption:** Select this to display the contents of **Caption** directly to the left of the Input Field box.
- **Allow Variables:** If this option is selected, then this input will accept non-numerical user input. This option is used when the Input Field is to be controlled by an externally defined signal (either constant or variable). See Allowing Signal Names for more details.

Cut, Copy, Paste and Delete Input Fields

You can cut, copy or paste a Input Field by using one of three methods:

- Select on the Input Field with a left mouse click so that Grips appear. Click the **Cut** or **Copy** buttons from the Main Toolbar. Click the **Paste** button Main Toolbar to paste.
- Right-click on the Input Field and select **Cut** or **Copy** from the pop-up menu. Right-click over a blank area of the Category page and select **Paste** from the pop-up menu.
- Select on the Input Field with a left mouse click so that Grips appear. Press **Ctrl + x** or **Ctrl + c** to cut or copy the label respectively. Press **Ctrl + v** to paste.

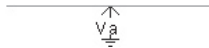
Allowing Signal Names

The allowance of variable signals can be a useful feature for reducing the amount of external connections in the component graphic, as well as adding flexibility to the component input parameters.

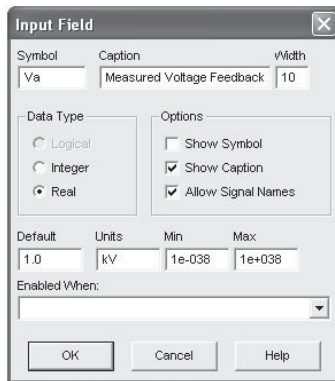
By selecting the **Allow Signal Names** feature, the Input Field will accept either numerical or non-numerical user input. In other words, you can either enter a numerical constant value, or you can enter the symbol name of a signal that has been defined elsewhere in the project. In this way, Input Fields can be used in the same manner as external connections.

EXAMPLE 9-2:

A user needs to use an instantaneous measured voltage, from a remote location in the system, as feedback to a user-defined source component. The first step is to ensure that the measured voltage signal is actually defined somewhere in the system. That is, it is being generated by a Voltmeter component similar to that shown below:



The second step is to define an Input Field that will accept the measured signal 'Va' as input: In the Parameters section of the Component Definition, add a new Input Field on an existing category page. Change the properties of the Input Field as follows:



Note that the **Allow Variables** option has been selected.

The **Allow Variables** option should remain de-selected if the particular Input Field is intended to be a constant value only.

Choice Box

Choice Boxes are used to allow the user to set conditions according to a selected choice. Choice Boxes can be included in a category

Chapter 9: Designing Components

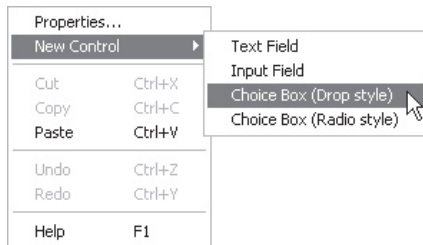
page as either a **Drop Style**, or **Radio Style**. In each case, the choice selected will possess an associated integer number, for use internally for conditional statements.

Adding Choice Boxes

To add a Choice Box to your component definition, the most straightforward method is to use the Parameters Bar:



Left click the **New Choice Box (Radio Style)** or the **New Choice Box (Drop Style)** buttons - a new Drop List or set of Radio Buttons should appear within the Category page. Another method is to use the right-click menu: Move the mouse pointer over a blank area of the Category page. Right-click and select **New Control | Choice Box (Radio Style)** or **New Control | Choice Box (Drop Style)**.



The new Choice Box should appear somewhere within the Category page.

Moving and Resizing Choice Boxes

To move the new Choice Box, left-click over top of the field to select it (Grips appear).



The Choice Box can now be moved in one of two ways:

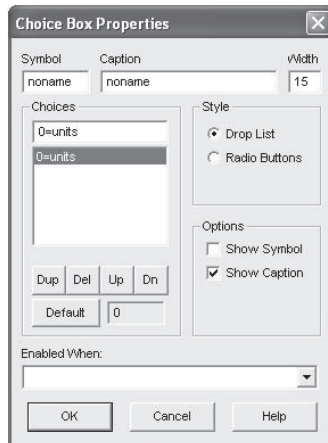
1. Click again and **hold** down the **left mouse button**. Move your mouse around.

2. Use the arrow keys on your keyboard. Press and hold an arrow key to move the box in large increments. Tap the arrow key to move the box in small increments.

To resize the new Choice Box, left-click over top of the field to select it (Grips appear). Select a Grip, hold down the left button and move in an outwards direction.

Changing Choice Box Properties

Choice Box properties can be adjusted through the **Choice Box Properties** dialog window. To change the Choice Box properties, right-click over the Choice Box and select **Properties...**



As shown above, features, such as caption and width, can be adjusted.

- **Symbol:** Enter a name for the Choice Box. Note that this name must be compatible with standard Fortran naming conventions (i.e. it must begin with a non-numeric character, do not include spaces, etc.).
- **Caption:** Enter text to describe the Choice Box. What is entered here will be displayed in the Category page directly to the left of the Choice Box.
- **Width:** Select the character width of the Choice Box drop list itself (i.e. white box /w arrow).
- **Choices:** Configure the choice list as described in Adding Choices to a Choice Box below.

Chapter 9: Designing Components

- **Style:** Select **Drop List** or **Radio Buttons** to toggle between Drop Style and Radio Style.
- **Enabled When:** Enter a conditional statement to determine under what input conditions the Choice Box is to be enabled. See 'Conditional Statements, Layers & Filters' in this chapter for more details.

Options:

- **Show Symbol:** Select this to display the contents of **Symbol** directly to the left of the Choice Box.
- **Show Caption:** Select this to display the contents of **Caption** directly to the left of the Choice Box.

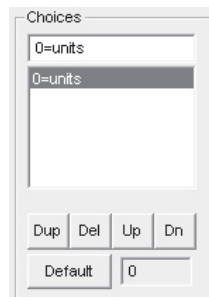
Cut, Copy, Paste and Delete Choice Boxes

You can cut, copy or paste a Choice Box by using one of three methods:

- Select on the Choice Box with a left mouse click so that Grips appear. Click the **Cut** or **Copy** buttons from the Main Toolbar. Click the **Paste** button Main Toolbar to paste.
- Right-click on the Choice Box and select **Cut** or **Copy** from the pop-up menu. Right-click over a blank area of the Category page and select **Paste** from the pop-up menu.
- Select on the Choice Box with a left mouse click so that Grips appear. Press **Ctrl + x** or **Ctrl + c** to cut or copy the label respectively. Press **Ctrl + v** to paste.

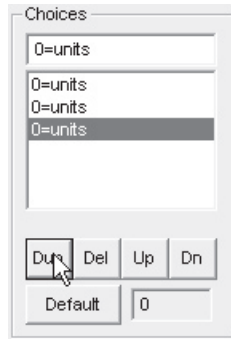
Adding Choices to a Choice Box

The choice input area can be broken down into three main parts: Choice editing field, the choice list and the control buttons. When a new Choice Box is created, one choice is added by default.

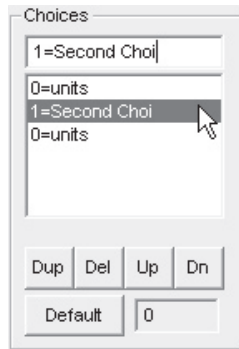


PSCAD

To add more choices to the list, click the **Dup** button.



To edit the contents of a choice: Left-click on the choice name within the list to select it, or use the **Up** or **Dn** buttons to scroll up or down the choice list respectively. The contents of the selected choice will appear in the choice-editing field at the top. Left-click within the edit field so that a flashing cursor appears, and then edit the contents.



Note that when adding choices to the Choice Box, the most important thing to remember is that a unique number must be assigned to each entry in the choice list. This is accomplished by simply adding a prefix to each entry in the form of:

<Assigned Number>=

It is this assigned number that will be associated with the Symbol name of this Choice Box.



Chapter 9: Designing Components

You can set the choice that will appear as the default selection, as follows: Left-click on a choice name within the list to select it, or use the **Up** or **Dn** buttons to scroll up or down the choice list respectively. Once selected, left-click the Default button – the associated integer will appear in the field directly to the right of the **Default** button.

To delete a choice in the choice list, left-click on a choice name within the list to select it, or use the **Up** or **Dn** buttons to scroll up or down the choice list respectively. Left-click the **Del** button.

Field Alignment

It is important to ensure that the Fields placed within a Category page are neat and tidy. There are two functions in the Parameters toolbar to help you organize all Fields both horizontally and vertically. These are:

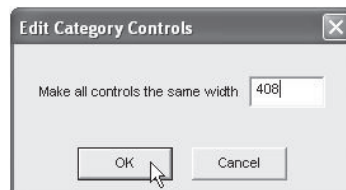
-  Make Same Width
-  Space Evenly

Make Same Width

This function makes all fields in a Category the same width. First define all Fields within the Category (see Text Field, Input Field, and/or Choice Box). Press the **Make Same Width** button on the Parameters toolbar:



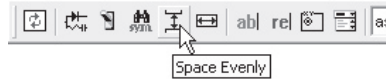
A dialog will appear asking to enter the width in pixels for all fields:



Enter the width and press **OK**.

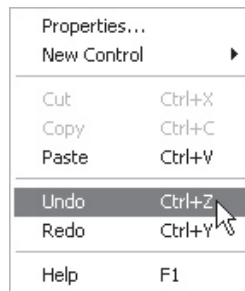
Space Evenly

This function spaces all fields in a Category evenly in the vertical direction. First define all Fields within the Category (see Text Field, Input Field, and/or Choice Box). Press the **Space Evenly** button on the Parameters toolbar:



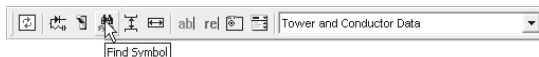
Undo and Redo

To undo or redo any changes in the Parameters section, select the **Undo** or **Redo** buttons in the Main Toolbar, press **Ctrl + Z** or **Ctrl + Y**, or right-click over a blank part of the Parameters window and select **Undo** or **Redo**.



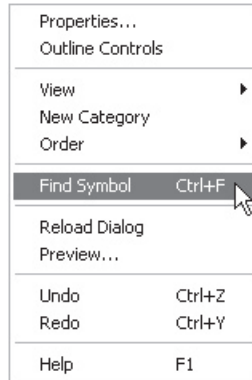
Find Symbol

The Parameters section includes a search feature for finding parameter Symbol Names in all Category pages that exist. To invoke this feature, the most straightforward method is to use the Parameters Bar.

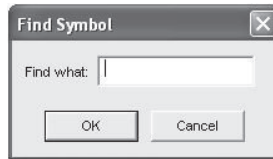


Left click the **Find Symbol** button. Another method is to use the right-click menu: Move the mouse pointer over a blank area (white part) of the Parameters window. Right-click and select **Find Symbol**.

Chapter 9: Designing Components



In either case, the Find Symbol dialog should appear.



Enter the Symbol Name of the Text Field, Input Field, or Choice Box you are looking for (do not enter the Caption or any other text) and click **OK**. If the Symbol Name exists in any of the Category pages, you will be automatically brought to that Category, and the corresponding field will appear selected (i.e. with Grips).

Saving and Reloading Parameters

Any changes made in the Parameters section should be saved regularly. To save changes, simply exit the component definition (i.e. go back to the Circuit window). To do this:

1. Either **left double-click** the project name in the Workspace window, or click the **Return to Circuit View** button in the Parameters Bar:



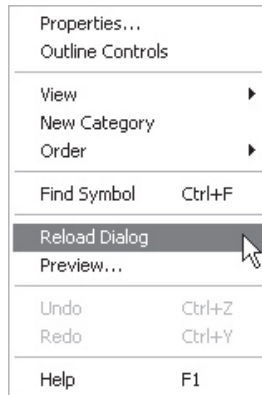
2. Save the Project by right-clicking on the project name in the Workspace and select **Save**.

PSCAD

If desired, you can at any time reload the graphics from the time at which they were last saved, by pressing the **Reload Dialog** button.



Another method is to use the right-click menu: Move the mouse pointer over a blank area (white part) of the Parameters window. Right-click and select **Reload Dialog**.



Caution should be exercised with this function, as once the dialogs are reloaded, **all changes will be lost**.

CONDITIONAL STATEMENTS, LAYERS AND FILTERS

All objects and fields associated with either the Graphic or the Parameters sections will possess an input field labelled either **Enabled When** or **Visibility Condition**. Within these fields, the user may add statements to either enable or disable the object, or make the object visible or invisible according to logical conditions.

Conditional Statements

The primary purpose of conditional statements is to either enable or disable an object (such as an Input Field), or to make an object visible or invisible (such as a Graphic Object). Conditional statements can be entered as input parameters in either a Graphic Object, a Category or an Input Field properties dialog.

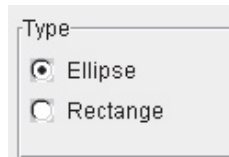
Chapter 9: Designing Components

The user can construct conditional statements by using both Arithmetic and Logical Operators, where the condition values themselves are always based on the value of a Choice Box. This feature adds another dimension to the component design, as through conditional statements the operation and appearance of the component can be controlled by the user.

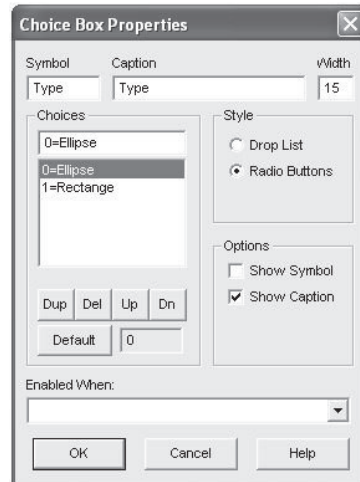
EXAMPLE 9-3:

A component designer wants to change the appearance of a component according to user-selected input. The component graphic is to be either an ellipse or a rectangle, depending on a Choice Box with Symbol Name **Type**. The Choice Box specifies two choices: Ellipse, which is given the value '0,' and Rectangle, which is given the value '1.'

In the component Parameters section, the designer adds a single Choice Box:

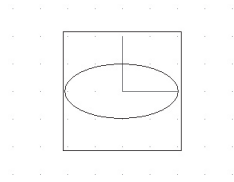


Choice Box Displayed
on Category Page

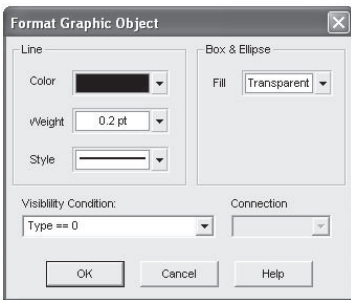


Choice Box Property Settings

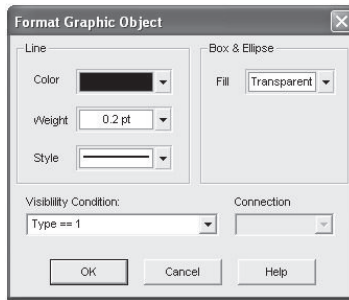
In the component Graphic section, the designer draws a simple ellipse and a simple rectangle:



In the properties dialog for the Graphic Objects, the following conditional statements are added to the respective **Visibility Condition** input fields:



Ellipse Property Settings



Rectangle Property Settings

Now, if a user selects 'Ellipse' in the Choice Box, only the ellipse will appear in the component graphic. If 'Rectangle' is chosen, only the rectangle will appear.

Conditional statements are not limited to a single logical truth. It is possible to use several logical conditions in one statement. For example, say there was another Choice Box added to the component of EXAMPLE 9-3 with Symbol Name **Type2**. Now assume that in order for the ellipse to be visible on the component graphic, it is required that **Type2** have a value of '3' (in addition to **Type** having a value of '0'). Then the following conditional statement would need to appear in **Visibility Condition** input field of the ellipse object properties dialog:

(Type == 0) && (Type2 == 3)

Chapter 9: Designing Components

The above states: If **Type** equals 0 and **Type2** equals 3, then make the ellipse visible.

Arithmetic Operators may also be included in conditional statements. For example, the following statement is also valid:

(Type + Type2 == 3)

The above states: If the sum of **Type** and **Type2** equals 3, then make the ellipse visible.

NOTE: Care must be exercised when using the divide (/) function, as a 'divide by zero' error can occur.

Layers

As component graphics become larger and more complicated, the graphical working environment can become quite unruly - especially when many conditional statements are used.

Fortunately, there is a way to avoid the potential clutter by utilizing the graphical Layers available in the Graphic section. Layers are based solely on conditional statements, in that whenever a conditional statement is entered into an object or field, a new layer is created. Any other Graphic Object, which utilizes an identical conditional statement, will also be included in that particular Layer.

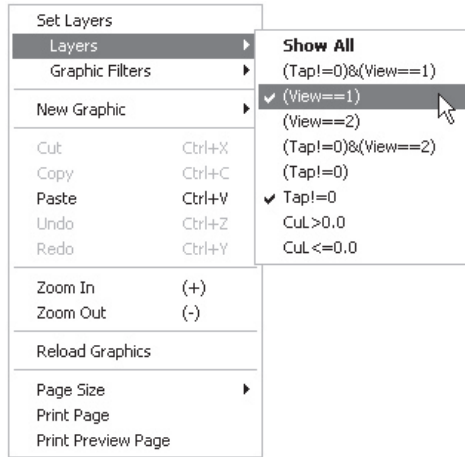
NOTE: Be sure when adding conditional statements that the statements themselves are identical in format, as well as logical truth. If the statements vary slightly in format, a separate layer will be created, even though the statements may indicate the same thing.

Viewing Layers

By default, only the layers visible on the graphic of a particular component instance will be initially visible in the Graphic window when the Design Editor is invoked (i.e. the component definition is edited). Although a component definition may have several instances, only the layers present on the instance used to access the definition will be visible by default.

PSCAD

To adjust the layer visibility, move the mouse pointer over a blank area of the Graphic window. Right-click and select **Layers**. The resulting sub-menu will list all of the available layers (or conditional statements) that currently exist in the component graphics.



Simply select or de-select each layer to make visible or invisible respectively. You can also select **Show All** to make all layers visible. There is also a button on the Graphic Palette, which will allow you to toggle between showing and hiding all Layers called **Show/Hide All Layers**.



Setting Layers

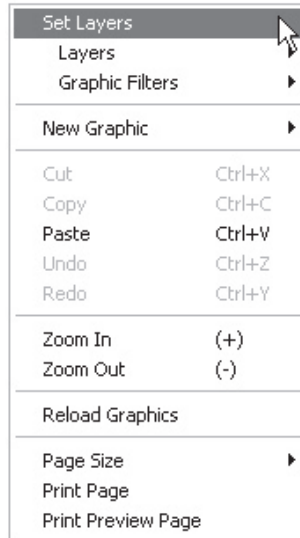
Instead of turning layers on and off through the right-click menu (which can become cumbersome as the number of layers increases), you can set and test the visible layers by using the **Set Layers** feature. This feature provides a fully functional preview of the parameter dialogs as they would be seen by a user. Simply set all the Choice Boxes to the desired settings and click the **OK** button - the graphical layers pertaining to that setting will become visible in the Graphic window.

To set the layers graphically, the most straightforward method is to use the Graphic Palette.

Chapter 9: Designing Components



Left click the **Set Layers** button. Another method is to use the right-click menu: Move the mouse pointer over a blank area (white part) of the Graphic window. Right-click and select **Set Layers**.



NOTE: The setting of layers is also available in the Parameters section to test enable/disable conditions of parameter fields.

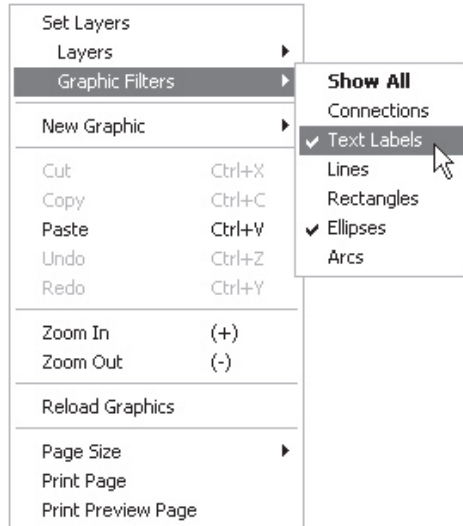
Graphic Filters

In addition to the Layers, you can also use the Graphic Filter function to help alleviate graphical clutter. Graphic Filters allow the user to view objects based on the Graphic Object type (i.e. Connections, Text Labels, etc.).

To adjust the Graphic Filters, the most straightforward method is to use the Graphic Filter Bar.



Another method is to use the right-click menu: Move the mouse pointer over a blank area (white part) of the Graphic window. Right-click and select **Graphic Filters**.



THE SCRIPT SECTION

Contained within the Script section is the heart of the Component Definition, where the fundamental purpose of the component is defined.

This section is used primarily for lower level code entry and consists of a variety of Segments, each performing a specific function. Not all Segments need to be utilized in every component, but the most commonly used Segments are Fortran, Computations and Branch, and these are included by default in every new Component Definition.

The Segments that your component definition will require depend on what function your component is to perform. Are pre-calculations required? Is it an electrical component? Is there Fortran code involved? The following sections describe each available Segment in detail.

Segments

Within the Script section, Segments are ordered similar to pages in a book. Segments are essentially simple text editors, where data,

Chapter 9: Designing Components

script and other code is added to perform a specific task. Segments may be added or deleted, but must follow a strict naming convention.

Managing Segments

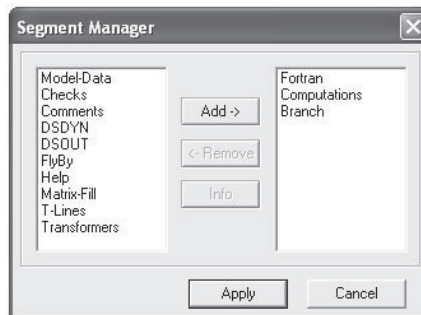
The Segment Manager allows users to add or remove Segments within the same dialog. To invoke the Segment Manager, the most straightforward method is to use the Script Bar:



Simply left-click the **Segment Manager** button on the toolbar. Another method is to use the right-click menu: Move the mouse pointer over a blank area of the Segment page. Right-click and select **Segment Manager**.

Refresh	F5
Find	Ctrl+F
Search/Replace	Ctrl+H
Segment Manager	
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Undo	Ctrl+Z
Redo	Ctrl+Y
Print	
Help	F1

The **Segment Manager** dialog will appear as shown below:



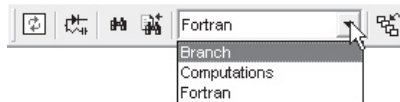
PSCAD

The left-side lists the available Segments, while the right-side lists the Segments already in the component definition. To add or remove Segments to or from the definition, simply select the **Add->** or **<-Remove** buttons. Once satisfied with the selection, click the **Apply** button. If you chose to remove any existing Segments, a dialog will appear asking you to confirm the removal. Press the **Cancel** button to cancel the operation.

NOTE: Deleted Segments will be lost if removed, as removal cannot be undone!

Viewing Segments

As mentioned above, categories are organized similar to pages in a book. Once you have added more than one Segment, there may be a need to navigate through them. The most straightforward method is to use the Script Bar.



Left click the down arrow on the **View Segment** drop list.

Segment Types

There are several Segment types available, each existing for a specific purpose, but only required if that specific function is needed as part of the component design. Most of the time, a component will only need to make use of two or three of these Segments.

Although this section does provide an assortment of simple examples, there is a huge wealth of examples in the PSCAD Master Library itself. Simply edit any Master Library definition (right-click on the component and select **Edit Definition...**) to study its contents.

Computations

The Computations Segment is an environment provided for the pre-processing of component input data. Although component parameter input may be in the most convenient form for the user, it may not be so convenient for the component code itself. For example, the component routine may require the system frequency in radians per second, but the input data field may ask the user for the same quantity in Hertz.

Chapter 9: Designing Components

The Computations Segment is the first Segment considered by the compiler when the Script section is compiled. Due to this fact, any quantities defined here may be substituted elsewhere in any of the other Segments. This Segment allows both mathematical and logical expression evaluation.

NOTE: All script processed within the Computations Segment is done so before any Fortran code is constructed for EMTDC, and only considered once. Therefore, quantities defined here are static for the remainder of the simulation.

A typical Computations Segment entry must have the following standard format:

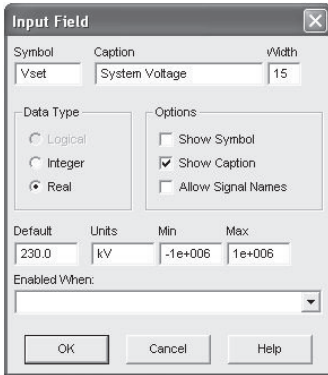
<DataType> <Name> = <Expression>

Where,

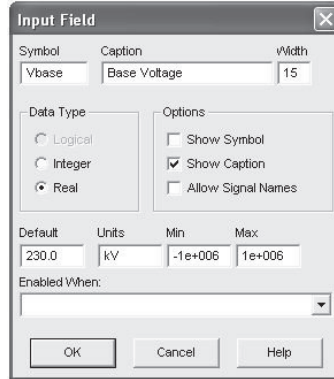
- <DataType> may be either a REAL or INTEGER type value. If omitted, it is assumed to be REAL.
- <Name> is the name of the constant.
- <Expression> is a mathematical or logical expression containing only constant parameters.

EXAMPLE 9-4:

A designer wants to convert an input parameter with Symbol Name 'Vset' in kV, to a per-unit value called 'SetPU,' before it is used in component Fortran code. An additional input parameter with Symbol Name 'Vbase' in kV has also been defined for the system voltage base.



Input Field Property Settings
(‘Vset’)



Input Field Property Settings
(‘Vbase’)

The Computations Segment could then contain something similar to the following:

```
REAL SetPU = Vset / Vbase
```

NOTE: The use of the Substitution Prefix Operator (\$) in this Segment is illegal.

In the above example, the designer has defined a new REAL constant entitled ‘SetPU.’ This constant may now be used in any other Segment in the Script section, both in equations and/or in logical expressions.

Branch

The Branch Segment is an environment provided for the input of electric branch information (if it exists), for insertion into the EMTDC Electric Network Conductance Matrix: It is essentially an input portal for control of the EMTDC Equivalent Conductance (GEQ) Interface. Branch design is accomplished by specifying the type and size of passive elements (i.e. lumped R, L and/or C), and to which electrical Connections they fall between.

Chapter 9: Designing Components

A single Branch statement represents a single electrical branch, to which is associated an impedance $Z = R + jX$, where Z is derived from a combination of series-connected, lumped R, L and/or C values. The Branch Segment may also be used to define switching branches and those that contain an ideal voltage source (i.e. infinite bus).

A typical Branch Segment entry must have the following standard format:

```
[<Branchname> = ] $<TO> $<FROM> [<Keyword>] [$]<R> [$]<L> [$]<C>]
```

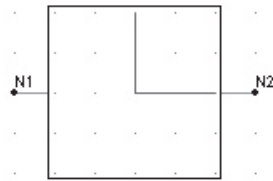
Where,

- <TO> and <FROM> are the Symbol names of the electrical Connections (nodes) defined in the Graphic section.
- <R> is the resistance value of the branch [Ω].
- <L> is the inductance of the branch (optional) [H].
- <C> is the capacitance of the branch (optional) [μF].
- <Branchname> = may be used to give the branch a name, which can be used to refer to the branch (instead of its nodes) in other sections. Note that many EMTDC subroutines require the branch name as an argument.
- <Keyword> can be either 'SOURCE' or 'BREAKER' and is explained in the following sections.
- \$ is the Substitution Prefix Operator (\$).

NOTE: If a branch does not contain all three types of elements, simply substitute 0.0 for the value of the element that is not present.

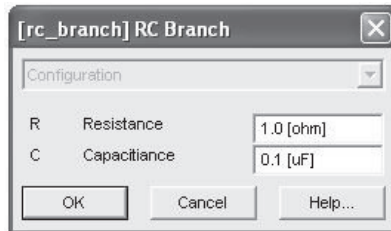
EXAMPLE 9-5:

A designer wants to define a component, which represents an RC series branch, connected in parallel with a purely inductive branch, between electrical Connections (nodes) 'N1' and 'N2.' These two Connections have already been defined in the Graphic section of the Component Definition.



Component Graphic with Electrical Connections 'N1' and 'N2'

The actual values of the 'R' and 'C' elements are to be entered through Input Fields (with Symbol names 'R' and 'C' respectively) in the Parameters section of the component. The inductance value is not accessible to the user and is given a static value of **0.001 H** directly.



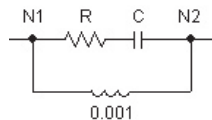
Component Parameters Dialog

The Branch Segment could then contain the following two statements:

```

$N1 $N2 $R 0.0 $C
$N1 $N2 0.0 0.001 0.0
    
```

These statements are equivalent to the following schematic:



The designer then decides to refer to the branches by Branch Name in other sections of the Component Definition. To achieve this, the names 'BRN1' and 'BRN2' are defined. The designer modifies the above Branch statements as follows:

Chapter 9: Designing Components

BRN1 = \$N1 \$N2 \$R 0.0 \$C
BRN2 = \$N1 \$N2 0.0 0.001 0.0

NOTE: If 'BRN1' and 'BRN2' are used in other Segments, they must be preceded by the Substitution Prefix Operator (\$).

The branch may also contain an inherent ideal source in series with the passive elements. The Branch Segment syntax allows you to indicate to PSCAD that there is a voltage source in a particular branch as follows:

```
[<Branchname> = ] $<TO> $<FROM> SOURCE [$<R> [$][<L>] [$][<C>]
```

Now that PSCAD knows that there is a source in this branch, control of the source (i.e. magnitude, etc.) is then up to the user. One way this can be accomplished is by defining the EMTDC Internal Variable EBR every time step, directly in the model code itself.

NOTE: For more examples on this, edit the definition of the Voltage Source Model 2 in the PSCAD Master Library.

EXAMPLE 9-6:

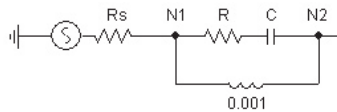
A designer adds a resistive branch containing an ideal voltage source to the component in EXAMPLE 9-5, which is to be connected between node 'N1' and a ground electrical Connection 'GND.' The source possesses a source resistance 'RS,' which can be entered as an input parameter.

The Branch Segment could then contain something similar to the following:

BRNS = \$GND \$N1 SOURCE \$RS 0.0 0.0
BRN1 = \$N1 \$N2 \$R 0.0 \$C
BRN2 = \$N1 \$N2 0.0 0.001 0.0

PSCAD

These statements are a schematic equivalent to:



The Branch Segment syntax also allows you to indicate to PSCAD that a particular branch is a switching branch:

```
[<Branchname> = ] <TO> <FROM> BREAKER <Initial_Value>
```

Where,

- <Initial_Value> is the initial resistance of the switch [Ω]. This value is used for initialization purposes and does not affect the simulation. A good default value to use is 1.0.

Fortran

The Fortran Segment is where any Fortran code, defining what the component is to model, is placed. Code entered in this Segment should either be in the form of standard formatted Fortran 77, or PSCAD Script (or a combination of these two). Fortran 90 format is not recommended here, as some incompatibility problems can occur during compilation (line breaks for example). It is also possible to define a function, or call an external subroutine from this Segment.

Although Fortran Segment code should be in Fortran 77 format, PSCAD will convert it to a form that is compatible with both Fortran 77 and 90 formats when the project is compiled. Code existing in external source files (i.e. subroutines or functions) should be compatible with both Fortran 77 and Fortran 90 however. This will avoid compatibility problems between the various Fortran compilers available for use with PSCAD. When using formatted Fortran in the Fortran Segment, it is important to note that all lines of code should be preceded by six spaces.

Chapter 9: Designing Components

EXAMPLE 9-7:

The following code is a simple example of how standard Fortran would appear in the Fortran Segment. This code defines an instantaneous voltage source magnitude **VT**, according to the value of an input parameter **Type**. If **Type** = 0, then the magnitude is calculated using the standard equation for a sinusoidal source with phase shift:

$$v(t) = V_{\text{peak}} \cdot \sin(2\pi f \cdot t + \phi)$$

If **Type** = 1, then the source magnitude is a constant given by **VDC**. **FREQ** and **PHI** are also input parameters defined in the Parameters section of the Component Definition. **TWO_PI** is an EMTDC Internal Constant.

```
! Calculation of an AC or DC voltage source magnitude
!  
!23456 Remember to precede code by at least six spaces!  
!  
      IF ( $TYPE .EQ. 0 ) THEN  
!  
! AC source  
!  
      $VT = $VPEAK*SIN( TWO_PI*$FREQ*TIME + $PHI )  
      ELSE  
!  
! DC source  
!  
      $VT = $VDC  
      ENDIF
```

This same code could also be written as a combination of standard Fortran and PSCAD Script as follows:

```
! Calculation of an AC or DC voltage source magnitude
!  
#IF $Type == 0  
!  
! AC source  
!  
      $VT = $VPEAK*SIN( TWO_PI*$FREQ*TIME + $PHI )  
#ELSE  
!  
! DC source  
!  
      $VT = $VDC  
#ENDIF
```

Note that the above two examples have been simplified slightly. Generally speaking, the value for $2\pi f \cdot t$ function should never be allowed to simply grow larger and larger as the simulation progresses. It should be reset to zero whenever $2\pi f \cdot t = 2\pi$ is reached.

There are pros and cons to adding standard Fortran code directly into the Fortran Segment, as opposed to calling a subroutine or function. Here are some key points to consider:

- All EMTDC Internal Variables may be utilized, without the need to declare them or add files, when coding in the Fortran Segment.
- PSCAD Script may only be utilized when coding in the Fortran Segment.
- All code placed in the Fortran Segment will be added directly to the module Fortran file (<module>.f), when the project is compiled. If there is a large amount of code and/or there are many instances of this component in the project, the Fortran file can become huge and difficult to debug. In such cases, it would be wise to create a function or subroutine, which could then be called from the Fortran Segment. Also, if standard Fortran line numbers are used in your code, these same numbers may appear multiple times in the Fortran file (<module>.f), causing compiler errors.

Chapter 9: Designing Components

The Fortran Segment is a 'smart' Segment, in that code placed here will be intelligently placed in the EMTDC System Dynamics, in order to avoid problems with time step delays. In other words, PSCAD will choose to place the code in either the DSDYN or DSOUT subroutines, according to what variables are being defined in the code, and what this component is connected to in the project.

DSDYN

This Segment is identical to the Fortran Segment, except that all code will be forced into the DSDYN subroutine in the EMTDC System Dynamics.

DSOUT

This Segment is identical to the Fortran section, except that all code will be forced into the DSOUT subroutine in the EMTDC System Dynamics.

Checks

The Checks Segment is used to ensure that data entered into the component input parameters by the user is reasonable. If a specific condition is true, then a given warning or error message can be made to appear in the Output Window when the case is compiled.

A typical Checks Segment entry would have the following standard format:

<MESSAGE TYPE> <Message> : <Expression>

- <MESSAGE TYPE> is the first word in the message line and must be chosen as either WARNING or ERROR depending on how severe the problem is. If chosen as a warning, the message will appear as a warning in the Output Window. If specified as an error, then the simulation will be stopped until the condition is rectified.
- <Message> is the diagnostic message that is printed if an error or warning is produced. This message should be descriptive enough for another user to determine what the problem is, and where it is coming from.
- <Expression> is the test used to determine if an error or warning message should be generated. This **expression is based on negative logic**, which means the message is generated only if the expression evaluates to False.

PSCAD

EXAMPLE 9-8:

A designer's component contains an input for frequency with Symbol Name **F**. The designer wants to ensure that only a value greater than zero is entered.

The Checks Segment should then contain the following:

```
ERROR System frequency must be greater than zero : F > 0.0
```

Both Logical and Arithmetic Operators may be used in the expression. For example:

```
WARNING R1 / R2 must be greater than 100 : R1 > 100*R2
```

Remember that negative logic is used in the above examples!

Help

The Help Segment is useful when the user wants to provide an external HTML based help file for a particular component. For example if the user has created an HTML document entitled 'help.html' for a particular component, then all that is required is that the name and extension be added to the Help Segment as shown below:

```
help.html
```

Then, when a user right-clicks on the component in the Circuit window and selects **Help** from the pop-up menu, the external document indicated will open.

In order to reduce confusion when linking components to help documents, the path to the Library Project (*.psl) file in which the Component Definition is located, will be appended to the filename. Therefore, **the external help file must always be located in the same directory as the parent Library Project.**

Comments

Add your comments / reminders / notes about the design of this component here. Do not use this Segment as a means of providing help

Chapter 9: Designing Components

to users. This Segment is ignored by PSCAD and is accessible only when you are editing the definition.

FlyBy

The FlyBy Segment provides an avenue to allow the designer to incorporate the PSCAD FlyBy help feature to a Component Definition. FlyBy windows can help to provide users with a quick description of the model itself, or even individual Connections on the component.

The FlyBy section syntax is straightforward:

```
<Descriptive text for the component>  
: <Connection_Symbol_Name>  
<Descriptive text for the connection>
```

EXAMPLE 9-9:

A designer wants to add FlyBy help to the component below:

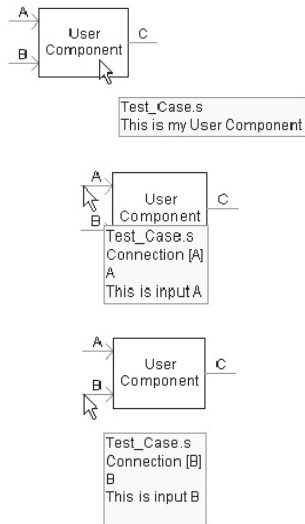


In addition to a general description of the component, the designer wants to provide a FlyBy for each of the input Connections **A** and **B**.

The FlyBy section should then contain the following:

```
This is my User Component  
:A  
This is input A  
:B  
This is input B
```

If the mouse pointer is now moved over the component in normal Circuit view, the following FlyBy windows will appear:



Transformers

The Transformers Segment is used both to define data for any existing mutual impedance matrices, as well as to provide dimensioning information to EMTDC regarding transformers and windings. A single component may contain multiple mutual impedance matrices.

The user must include transformer information in three main parts:

- **Number of transformers (matrices):** This is accomplished through the use of a `#TRANSFORMERS` directive. See Chapter 10 for more details on this directive.
- **Number of Windings:** This number will define the dimension of each matrix. If this number is given as a positive value, then PSCAD will assume that the matrix is in non-inverted format, and therefore the matrix data must be entered directly. If the number is negative, it signifies that this transformer is 'ideal' (i.e. contains only inductance) and PSCAD will assume that the matrix is already inverted. The matrix data must therefore be entered accordingly.
- **Matrix Data:** The matrix data is entered here, depending on whether the transformer is 'ideal' or not.

Chapter 9: Designing Components

According to the above description, a basic Transformers Segment would appear as follows (for a classical type transformer) in general format:

```
#TRANSFORMERS <Number_of_Transformers>  
<Prefix><Number_of_Windings> /  
<Node_1> <Node_2> <R_11> <L_11> /  
<Node_2> <Node_3> <R_12> <L_12> <R_22> <L_22> /  
...
```

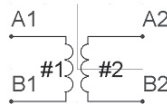
NOTE: The Transformers Segment format is different when defining a UMEC type transformer. For more information on the UMEC format, see the UMEC transformer component definitions in the Master Library, or contact PSCAD Support Services (support@pscad.com).

<Prefix> is not mandatory, but is required to indicate an 'ideal' transformer matrix (i.e. already inverted). Also, resistance values (i.e. <R_xx>) are not required if the transformer is 'ideal.'

Note that only the diagonal and the data below the diagonal need to be entered. The matrix is assumed to be symmetrical.

EXAMPLE 9-10:

A user wants to create a non-ideal single-phase, two-winding transformer. The component Graphics, Parameters and other Script Segments, has already been included. The component electrical Connections have been set-up as shown below in the Graphic section.



The winding self and mutual resistance and inductance parameters have been defined and given **Symbol Names** 'R11,' 'L11,' 'R12,' 'L12,' 'R22' and 'L22.'

PSCAD

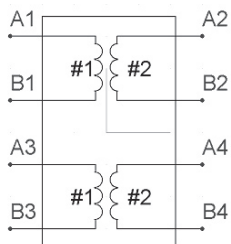
The Transformers Segment should then appear similar to the following:

```
#TRANSFORMERS 1
!  
2 /  
$A1 $B1      $R11 $L11 /  
$A2 $B2      $R12 $L12  $R22 $L22 /  
!
```

When designing a component with more than one transformer, you can use a special syntax (called simply 888 syntax) so that, if all the matrix elements are identical, you can avoid entering the repetitive data. This is especially useful if the mutual impedance matrices are very large and numerous.

EXAMPLE 9-11:

In the example above, the user wants to modify the component, so that it contains two identical single-phase transformers within it. The component electrical Connections have been set up, as shown below in the Graphic section.



The Transformers Segment should then appear similar to the following:

Chapter 9: Designing Components

```
#TRANSFORMERS 2
!  
2 /  
$A1 $B1      $R11 $L11 /  
$A2 $B2      $R12 $L12  $R22 $L22 /  
!  
888 /  
!  
$A3 $B3 /  
$A4 $B4 /
```

Model-Data

The Model-Data Segment is used as a way to bring input data into a user-defined subroutine, without having to declare an argument for each bit of data. There is no specific format for text in this Segment, as this format depends on READ statement within the users code.

When the project is compiled, PSCAD will include all text within this Segment in the corresponding Data File, under the headings DATADSD or DATADSO. If the subroutine is called from DSDYN, the Model-Data contents will appear in the DATADSD section, and DATADSO otherwise.

In order to read this information, a standard Fortran READ statement must be added to the user-defined subroutine as follows:

```
!  
! Add include file to define IUNIT  
!  
!      INCLUDE 'fnames.h'  
!  
! ...  
!  
!      READ(IUNIT,*) ...  
!
```

Matrix-Fill

When a Project is initially compiled, PSCAD will construct a temporary logical matrix for the purpose of indicating how the electrical system is connected (i.e. how nodes and branches are put together). The Optimize Node Ordering algorithm in PSCAD then

PSCAD

uses this information to optimize node placement in the actual system conductance matrix. However, only electrical nodes and branches, defined within the Branch Segments of each component present in the system, are considered.

Any internal node connections defined in Segments other than Branch will not be included in this matrix. Therefore, the logical matrix may have missing information, thereby reducing the effectiveness of the Optimize Node Ordering algorithm. The Matrix-Fill Segment can be used to 'help' this algorithm by providing the omitted connection information for any internal component nodes.

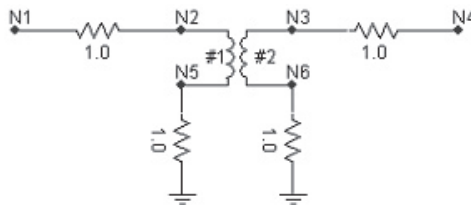
The general format for a Matrix-Fill Segment statement would appear as follows:

```
<Node_1> <Node_2> <Node_3> <Node_4> ...
```

Where <Node_#> is the **Symbol Name** of the Connection involved with Matrix-Fill. All Connections in a single statement are assumed as connected to each other.

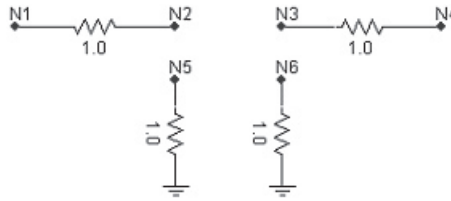
EXAMPLE 9-12:

A user creates a single-phase electrical circuit in PSCAD, which includes a classical transformer component connected as follows:



As can be seen above, there are six electrical nodes in this system, and therefore the system matrix will have dimension 6 x 6. Internal connections internal to the transformer component cannot be determined initially by PSCAD, and so when a logical matrix for the Optimize Node Ordering algorithm is created, the transformer appears as a 'black box' as shown below:

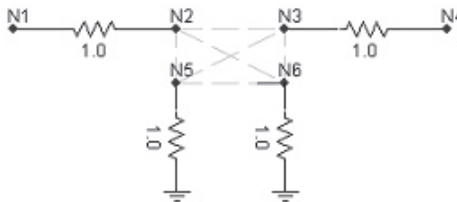
Chapter 9: Designing Components



The logical matrix for the Optimize Node Ordering algorithm would appear as follows, where **X** indicates a known connection:

$$\begin{bmatrix} X & X & 0 & 0 & 0 & 0 \\ X & X & 0 & 0 & 0 & 0 \\ 0 & 0 & X & X & 0 & 0 \\ 0 & 0 & X & X & 0 & 0 \\ 0 & 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & 0 & X \end{bmatrix}$$

The matrix above does not contain all of the connection information required, however. We know through the Transformers Segment in the transformer component, that nodes N2, N3, N5 and N6 are all part of a mutual impedance matrix and are hence all connected together as shown below:

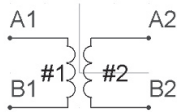


By adding in Matrix-Fill information, we can tell PSCAD to add connection information to the matrix so it will appear as follows, where '+' indicates a connection added by Matrix-Fill:

$$\begin{bmatrix} X & X & 0 & 0 & 0 & 0 \\ X & X & + & 0 & + & + \\ 0 & + & X & X & + & + \\ 0 & 0 & X & X & 0 & 0 \\ 0 & + & + & 0 & X & + \\ 0 & + & + & 0 & + & X \end{bmatrix}$$

PSCAD

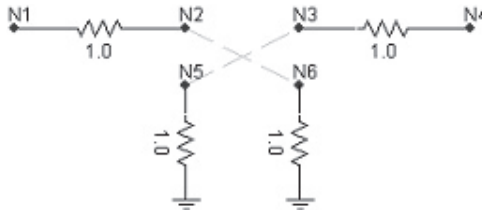
If the four electrical Connections on the transformer were given as follows in the Graphic section of the Component Definition,



then the Matrix-Fill Segment should appear as:

```
$A1 $B1 $A2 $B2
```

Note that in the above example, it assumed that all connections between the four nodes involved are possible. If, however, only connections as shown below are needed,



then the Matrix-Fill Segment should appear as:

```
$A1 $B2  
$A2 $B1
```

T-Lines

The T-Lines Segment is used to specify any electrical Connections, defined in the component Graphic section, to be used as sending or receiving end electrical Connections for a Transmission Line or Cable. The T-Lines Segment is rarely needed in component design. The general format of a T-Lines Segment statement is given below:

```
<Subsystem_#> <Cond_1> <Cond_2> <Cond_3> ...
```

<Cond_#> signifies that any Connection Symbol Name that is entered in this position will be the sending or receiving end Connection

Chapter 9: Designing Components

for that particular conductor number in the T-Line or Cable properties.

This Segment is used only within the Transmission Line and Cable Interface components in the Master Library.

Undo and Redo

To undo or redo any changes in the Script section, select the **Undo** or **Redo** buttons in the Main Toolbar, press **Ctrl + Z** or **Ctrl + Y**, or right-click over a blank part of the Parameters window and select **Undo** or **Redo**.

Refresh	F5
Find	Ctrl+F
Search/Replace	Ctrl+H
Segment Manager	
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Undo	Ctrl+Z
Redo	Ctrl+Y
Print	
Help	F1

Syntax Colouring

In order to make Script section code easier to understand, syntax colouring is used similar to code viewed in popular developer software programs. Most PSCAD Script commands will appear in blue, while all commented code will appear in green, for example:


```
#LOCAL INTEGER M_ACTV
#IF IcTyp==1
  #LOCAL REAL RVD1_1
  #LOCAL REAL RVD1_2
      CALL E_CPFSYNC($P0Z,$Q0Z,$VT,$Pheta,$Vbase,RVD1_1,RVD1_2)
#ENDIF
! Synchronous machine model
  STORI(NSTORI) = NINT(STOR(NEXC+294))
```

Syntax colouring can be disabled in the Workspace Settings dialog under the **Views** tab.

INTERNAL OUTPUT VARIABLES

An Internal Output Variable is internal component data, which has been made available as a signal to be plotted or monitored in PSCAD. Before any internal quantities are made available however, they must be defined within the component Fortran Segment using an #OUTPUT directive.

Internal data used as output can either come from an internal component variable, from an EMTDC Storage Array, or as a measured branch current or voltage from EMTDC.

Outputting EMTDC Measured Voltages and Currents

There are a couple of substitution operators that can be used with #OUTPUT directives to extract measured quantities directly from electrical system nodes and branches. These are described below.

CBR

CBR is a special substitution operator that will output the current measured in a specified branch. A special syntax is used for this - see the example below:

EXAMPLE 9-13:

A user wants to measure the current in a branch called 'BRN.' A Text Field has been added to the component Parameters section with Symbol Name 'Ia.'

The following should then appear in the Fortran segment:

Chapter 9: Designing Components

```
#OUTPUT REAL Ia {$CBR:BRN}
```

Note that 'BRN' is not preceded by a \$ Prefix Operator.

VDC

VDC is a special substitution operator that will take the voltage difference between two different electrical nodes. A special syntax is used for this - see the example below:

EXAMPLE 9-14:

A user wants to measure the voltage difference between two Connections defined within a user component, called 'N1' and 'N2.' A Text Field has been added to the component Parameters section with **Symbol Name** 'Vdiff.'

The following should then appear in the Fortran Segment:

```
#OUTPUT REAL Vdiff {$VDC:N1:N2}
```

Note that 'N1' and 'N2' are not preceded by a \$ Prefix Operator.

ADDING A REFERENCE TO A SOURCE FILE

In PSCAD, you may call a subroutine or function defined in an external source file (i.e. *.f, *.f90, *.for or *.c file) from within the Fortran, DSDYN or DSOUT sections of your component definition. In order to have this code included when the project is compiled and built, you must ensure a reference to this file is provided. This can be done in one of two ways:

1. Add a reference from the Project Settings dialog: Right-click over a blank area of your project page in Circuit view and select **Project Settings....** In the Fortran section, add a reference to the file in the **Additional Source Files** input field. See 'Additional Source (.f) Files' in this chapter for more details.

2. Add a File Reference component: Right-click over a blank area of your project page in Circuit view and select **Add Component | File Reference**. Open the File Reference and select the source file.

INTERFACING TO C CODE

In PSCAD, you may interface to C language source code in the form of a subroutine or function. As C code cannot be used directly within component definition Fortran, DSDYN or DSOUT Segments, it must be provided in the form of an external source (*.c) file or object file.

There are subtle differences between how C source code is handled between the EGCS/GNU Fortran 77 compiler and the Fortran 90 compilers compatible with PSCAD. The method for each is described in the following sections.

NOTE: Before running the case, the user must ensure that the *.c (and *.f) file(s) be referenced in the Additional Source Files input field. Alternatively, object files (*.obj) or a library file (*.lib) (made from the source files) should be referenced in the Additional Library (*.lib) and Object (*.obj) Files input field.

For more details on interfacing to C source code, see the example projects in the ...\\examples\C_interface directory within your installation directory.

Using the GNU Compiler

The most straightforward method to utilizing source code written in C is to use the EGCS/GNU Fortran 77 compiler. This is simply due to the fact that this program converts Fortran code to C anyway before creating the executable. Providing C source code then makes relatively little difference from the perspective of the GNU compiler.

C subroutines and functions can be called directly from within the Fortran, DSDYN, or DSOUT segments of a component definition, just as you would do it for Fortran. The following examples illustrate this for calling both a C subroutine and a C function.

Chapter 9: Designing Components

EXAMPLE 9-15:

A user has written the following simple subroutine in C, and wants to call this source code from a user-defined component:

```
/* User C Source Code */
void test_csub__(int* arg, int* res)
{
    *res = 6*(*arg);
}
```

A standard call statement can be made from within the Fortran, DSDYN or DSOUT segments of the component definition.

```
CALL TEST_CSUB($IN,$OUT)
```

Where \$IN and \$OUT are input and output Connections on the component respectively.

EXAMPLE 9-16:

A user has written the following simple function in C, and wants to use this source code in a user-defined component:

```
/* User C Source Code */
int test_cfun__(int *arg)
{
    return 2*(*arg);
}
```

The function must be declared before it is used within the component. Then, the function is used as it normally would in Fortran in the Fortran, DSDYN or DSOUT segments of the component definition.

PSCAD

```
#FUNCTION INTEGER TEST_CFUN
    $OUT = TEST_CFUN($IN)
```

Where \$IN and \$OUT are input and output Connections on the component respectively.

Using a Fortran 90 Compiler

When using either the Digital Fortran or Compaq Fortran 90 compilers, some extra steps are involved in interfacing with C source code.

An auxiliary Fortran subroutine must be provided for the purpose of constructing an interface (or 'wrapper') to the C code. It is this Fortran subroutine that is called from the Fortran, DSDYN or DSOUT segments of your user-defined component.

The following examples illustrate how to call a C subroutine when using the Fortran 90 compilers.

EXAMPLE 9-17:

A user has written the following simple subroutine in C, placed it into a C file (*.c).

```
/* User C Source Code */
void test_csub(int* arg, int* res)
{
    *res = 6*(*arg);
}
```

The user then defines an interface subroutine in a Fortran (*.f) source file as follows:

Chapter 9: Designing Components

```
SUBROUTINE AUX_CSUB(in, out)
  INTEGER in, out
!
! Fortran 90 interface to a C procedure
!
  INTERFACE
    SUBROUTINE TEST_CSUB (in, out)
      !DEC$ ATTRIBUTES C :: TEST_CSUB
      !DEC$ ATTRIBUTES REFERENCE :: in
      !DEC$ ATTRIBUTES REFERENCE :: out
    !
    ! in, out are passed by REFERENCE
    !
      INTEGER in, out
    END SUBROUTINE
  END INTERFACE
!
! Call the C procedure
!
  CALL TEST_CSUB(in, out)
END
```

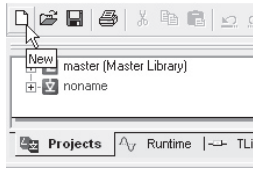
A standard call statement is then added to the Fortran segment of the user component definition:

```
CALL AUX_CSUB($IN,$OUT)
```

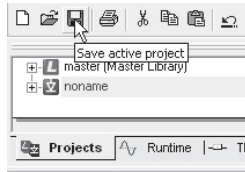
Where \$IN and \$OUT are input and output Connections on the component respectively.

TUTORIAL: CREATING A NEW COMPONENT

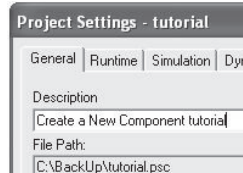
Open PSCAD, create a new Case Project, save the Project under a unique name, edit the Project Settings and add a Description.



Create a New Case Project ('noname')

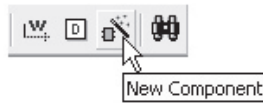


Save the Project Under a Unique Name (ex. 'tutorial.psc')



Edit the Project Description in the Project Settings Dialog

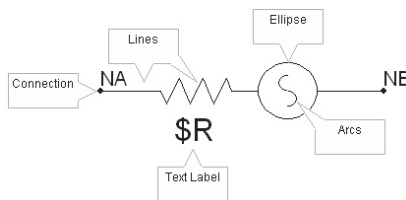
Create a new component by using the Component Wizard.



Invoke the Component Wizard

Adding Graphics

Edit the Component Definition and in the Graphic section, draw the following diagram (minus the sticky notes); where **\$R** is a Text Label, and **NA** and **NB** are Connections.

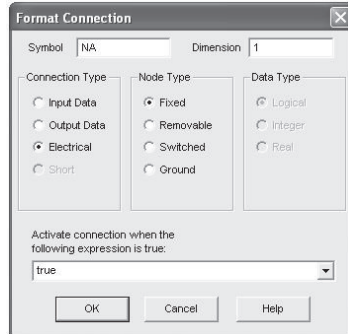


See 'The Graphic Section' in this chapter for more details on drawing objects available.

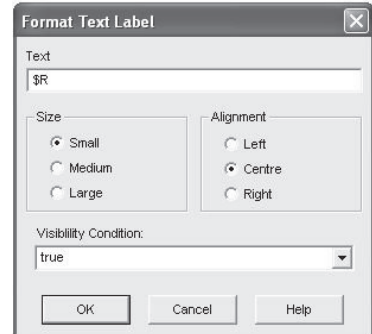
NOTE: Make sure the graphics are of a reasonable size in relation to other components in the Master Library.

Configure the properties of the Connections so that they are fixed electrical nodes. Also, configure the Text Label to display '**\$R**':

Chapter 9: Designing Components



Connection Property Settings ('NA')



Text Label Property Settings ('\$R')

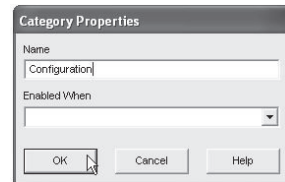
Save your component changes by Returning to Circuit View. Save the Project.

Adding a User Interface

Edit the Component Definition and in the Parameters section, add a new Category page. Give the new Category a name (say 'Configuration').



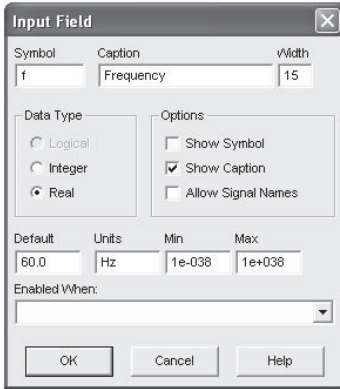
Add New Category in Parameters Section



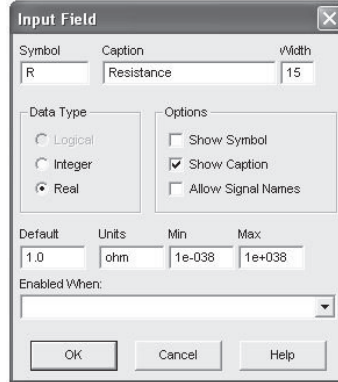
Give Category a Name

Add Input Fields to the Category with the following

Captions: 'Voltage Magnitude (RMS),' 'Frequency,' 'Ramp Up Time' and 'Resistance.' These are to have **Units** set to kV, Hz, sec and ohm respectively. Give each field a short, but descriptive **Symbol Name** (say 'Vrms,' 'f,' 'tr' and 'R' respectively) and provide a reasonable **Default** value (see the image below), as well as **Min** and **Max** settings.

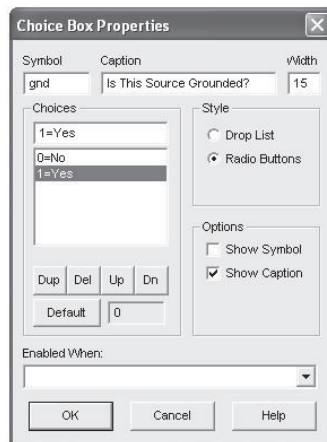


Input Field Properties
(‘Frequency’)



Input Field Properties
(‘Resistance’)

Add a Choice Box to the Category with the **Caption** ‘Is This Source Grounded?.’ Give it an appropriate **Symbol Name** (say ‘gnd’) and then add two Choices: ‘1=Yes’ and ‘0=No.’ Decide on whether to use Radio Buttons or a Drop List format.



Choice Box Properties (‘Is This Source Grounded?’)

Arrange the fields in a neat manner and once completed, your new Category page should look something like the following (when viewed in The Parameters Section):

Chapter 9: Designing Components

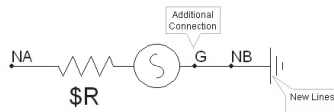
Voltage Magnitude (RMS)	1.0
Frequency	60.0
Ramp Up Time	0.02
Resistance	1.0
Is This Source Grounded?	
<input checked="" type="radio"/> No	
<input type="radio"/> Yes	

Save your component changes by Returning to Circuit View. Save the Project.

Adding Conditional Statements to the Graphic Objects

In order to make our component more convenient to use, we should give the user the option to ground the source at one end. This will be the primary purpose of the 'Is This Source Grounded?' Choice Box. Using the condition of this field (i.e. Yes or No), we can alter both the appearance and the electrical properties of the component.

Edit the Component Definition and in the Graphic section, modify the component graphics as shown in the diagram below, where **G** is a Connection configured as an electrical ground node.



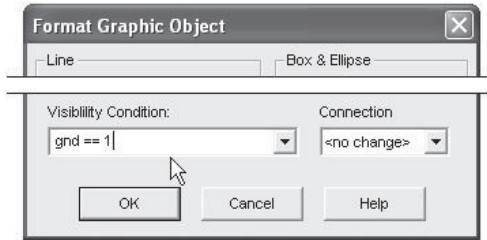
Modified Component Graphics

Format Connection					
Symbol	G	Dimension	1		
- Connection Type		- Node Type		- Data Type	
<input type="radio"/> Input Data	<input type="radio"/> Fixed	<input checked="" type="radio"/> Logical			
<input type="radio"/> Output Data	<input type="radio"/> Removable	<input type="radio"/> Integer			
<input checked="" type="radio"/> Electrical	<input type="radio"/> Switched	<input type="radio"/> Real			
<input type="radio"/> Short	<input checked="" type="radio"/> Ground				
Activate connection when the following expression is true:					
true					
OK		Cancel		Help	

Connection Property Settings ('G')

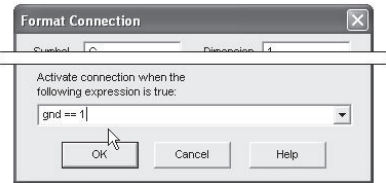
In the **Visible When** field of all the newly added Lines, add Conditional Statements so that they will become Visible/Invisible according to the setting specified in the 'Is This Source Grounded?'

Choice Box. That is, when 'Is This Source Grounded?' is set to Yes, the ground symbol graphics above should be visible.

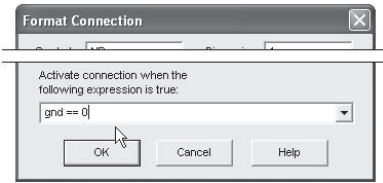


Editing Conditional Statement in Line Objects

In the **Activate Connection when the Following Expression is True** field of Connections **G** and **NB**, add conditional statements so that they become Enabled or Disabled according to the setting specified in the 'Is This Source Grounded?' Choice Box. That is, when 'Is This Source Grounded?' is set to Yes, Connection **G** should be enabled and Connection **NB** should be disabled (and vice-versa):



Editing Conditional Statement in Connection 'G'



Editing Conditional Statement in Connection 'NB'

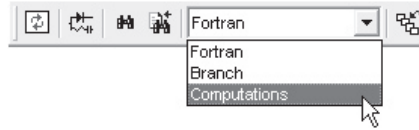
You may be wondering at this point about the placement of the Connections **G** and **NB** in the Graphic section diagram above. When placing your Connections within your component graphic, it is important to consider where the most intuitive position for the Connection is. That is, it would be best to have the Connection **NB** at the end of the Line Object, as this would be an obvious connection point.

Save your component changes by Returning to Circuit View. Save the Project.

Pre-Processing User Input

Edit the Component Definition and in the Script section, navigate to the Computations Segment.

Chapter 9: Designing Components



In the Computations Segment text window, add lines of Script to convert your 'Voltage Magnitude (RMS)' (**V_{rms}**) input from RMS to peak kV and to convert your 'Frequency' (**f**) input from Hz to peak radians per second. Your Computations Segment should appear similar to the following:

```
1 REAL Vp = SQRT(2.0)*Vrms
2 REAL f_rad = 2.0*pi*f
```

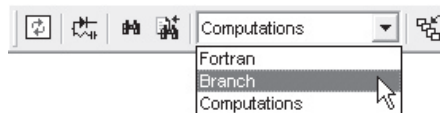
This Computations Script declares two new REAL variables called **V_p** and **f_{rad}**. As the Computations Segment is considered by the compiler before any other code Segments (i.e. Fortran), these new variables can be used anywhere in subsequent code procedures.

The EMTDC Subroutine that we will be using to define the voltage source requires these values in the format defined above. You could, of course, simply let the user enter these values in kV peak and rad/s directly. Then, the need for the Computations Segment will be removed, but this format may not be the most convenient to those using the component. As the designer, this is up to you.

Save your component changes by Returning to Circuit View. Save the Project.

Defining the Electrical Branch

Edit the Component Definition and in the Script section, navigate to the Branch Segment.



In the Branch Segment text Window, add statements that define the electrical properties of the component for use by EMTDC. This branch is to be purely resistive with resistance **R**, as defined by the Input Field 'Resistance.' Also, Conditional Statements are added so that the branch will be defined between **NA** and **NB** or **NA** and **G**,

depending on the condition of the 'Is this source grounded?' Choice Box. Your Branch Segment should then look similar to the following:

```
1 #IF gnd == 1
2   $NA $G $R 0.0 0.0
3 #ELSE
4   $NA $NB $R 0.0 0.0
5 #ENDIF
```

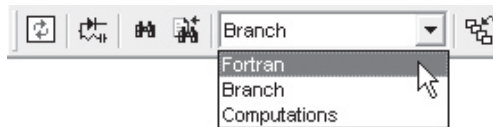
The above defines a source resistance value to the branch where we will place the source. Of course, the branch itself will be defined according to whether the user has grounded the source directly or not.

Save your component changes by Returning to Circuit View. Save the Project.

Adding Code to Define the Source

Instead of writing our own code to define the voltage source, we can use an existing EMTDC Subroutine to do it for us. Then we simply need to ensure that all of the subroutine arguments are defined in the component, and then call the subroutine. The subroutine we will use is called EMTDC_1PVSRC, which represents an ideal voltage source and is the same subroutine used in the Single-Phase Voltage Source Model 2 in the PSCAD Master Library.

Edit the Component Definition and in the Script section, navigate to the Fortran Segment.



Add a subroutine declaration statement at the top of the Fortran Segment text window, using the #SUBROUTINE Directive. Make sure to include the exact name of the subroutine along with a brief description. Add a Fortran CALL statement to call the EMTDC subroutine EMTDC_1PVSRC (be sure to include all of the required arguments in the call statement). Finally, add a #STORAGE Directive to ensure the proper storage allocations for the routine. This information is detailed in Ideal Voltage Source (1-Phase).

When complete, your Fortran Segment should look similar to the following:

Chapter 9: Designing Components

```
1 #SUBROUTINE EMTDC_1PVSRC Single Phase Source
2 #STORAGE LOGICAL:1 REAL:8 INTEGER:6
3 !
4     CALL EMTDC_1PVSRC($SS,$BRN,$tr,1,$Vp,$f_rad,0.0)
```

Note that the storage allocation is necessary only when using a Fortran 90 compiler, and provides memory allocation information for Dynamic Dimensioning in EMTDC. It is a good idea to always include `#STORAGE` statements however, to maintain component compatibility with all compilers.

Save your component changes by Returning to Circuit View. Save the Project.

Setting the Branch Source

According to the given subsystem number (SS) and branch number (BRN), PSCAD can determine where in the electric network to place the source. We have not yet defined this in our Branch Segment however, so we will go back and do that now by first applying a name (i.e. 'BRN') to our defined branch. Also, so that PSCAD knows to enable the internal source in this branch, we must add the 'SOURCE' syntax as follows:

```
1 #IF gnd == 1
2     BRN = $NA $G SOURCE $R 0.0 0.0
3 #ELSE
4     BRN = $NA $NB SOURCE $R 0.0 0.0
5 #ENDIF
```

For more information on branch naming and the SOURCE syntax, see Branch Segment.

Save your component changes by Returning to Circuit View. Save the Project.

Sanity Checks

It is always a good idea to apply sanity checks to the users input, so as to avoid erroneous results. This is accomplished by using the Checks Segment.

Edit the Component Definition and in the Script section, add a new Checks Segment by using the Segment Manager. Add a statement to the Checks Segment text window, which will give an error when the entered source resistance (i.e. R) is less than or equal to 1. Note

that you must use negative logic in this section, and so this statement should appear as follows:

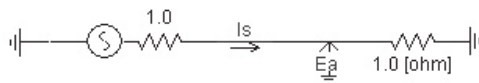
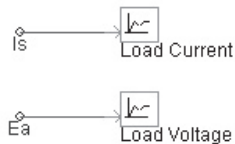
```
1 ERROR Source Resistance must be greater than zero : R > 0.0
```

Save your component changes by Returning to Circuit View. Save the Project.

Testing Your New Component

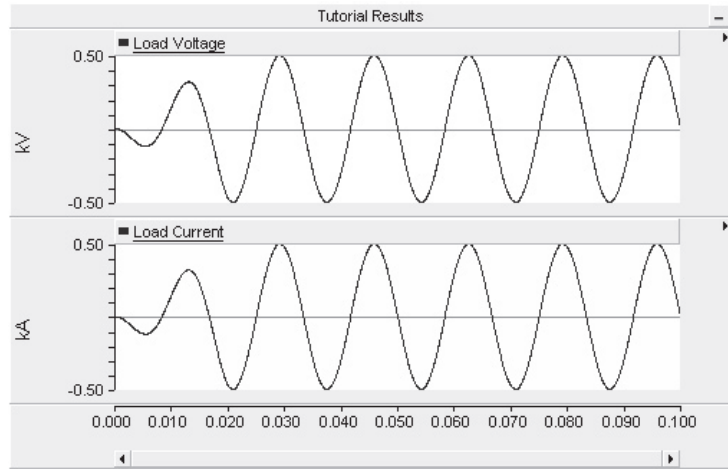
At this point it would be a good idea to check if your component will actually run, and if it does, whether the results are correct. Below are what the results should be if the source model was connected to the indicated circuit, and if the source is set to:

- Voltage Magnitude (RMS) = 0.7071 kV
- Frequency = 60 Hz
- Ramp-Up Time = 0.02 s
- Source Resistance = 1.0 Ω



The results are as shown below:

Chapter 9: Designing Components



Save the Project.

Adding Internal Output Variables

It is possible to monitor quantities internal to the component, for display on plots and meters. One way to accomplish this is by setting up an Internal Output Variable. In this tutorial, we will set up internal variables to monitor source voltage and branch current.

Edit the Component Definition and in the Parameters section, add two Text Fields: One with **Symbol Name** 'Isource' and **Caption** 'Source Current.' The other with **Symbol Name** 'Vsource' and **Caption** 'Source Terminal Voltage.'

Symbol	Caption	Width
Isource	Source Current	15

Text Field Properties
(‘Isource’)

Symbol	Caption	Width
Vsource	Source Terminal Voltage	15

Text Field Properties
(‘Vsource’)

Arrange the fields in a neat manner and once completed, your Category page should look something like the following:

Voltage Magnitude (RMS)	<input type="text" value="1.0"/>
Frequency	<input type="text" value="60.0"/>
Ramp Up Time	<input type="text" value="0.02"/>
Resistance	<input type="text" value="1.0"/>
Is This Source Grounded?	
<input checked="" type="radio"/> No <input type="radio"/> Yes	
Source Current	<input type="text"/>
Source Terminal Voltage	<input type="text"/>

Save your component changes by Returning to Circuit View. Save the Project.

Edit the Component Definition and navigate to the Fortran Segment. With an #OUTPUT Directive statement, extract the current in this source branch (i.e. from branch 'BRN'), using the CBR Internal Variable. Your Fortran Segment should look similar to the following:

```

1 #SUBROUTINE EMTDC_1PVSRC Single Phase Source
2 #STORAGE LOGICAL:1 REAL:8 INTEGER:6
3 !
4     CALL EMTDC_1PVSRC($SS,$BRN,$tr,1,$Vp,$f_rad,0.0)
5 !
6 #OUTPUT REAL Isource {$CBR:BRN}

```

The #OUTPUT statement above will define a variable 'Isource' and output to it the current flowing in branch 'BRN.'

Extract the node voltage from Connection **NA** to Connection **G** or **NB** with another #OUTPUT Directive statement and using the VDC Internal Variable. Remember that we have added Conditional Statements that allow users to directly connect the source branch to ground. So, the branch voltage measurement will depend on which Connections are enabled. We can then choose a branch statement according to the 'Is this source grounded?' Choice Box.

Your Fortran Segment should then look similar to the following:

Chapter 9: Designing Components

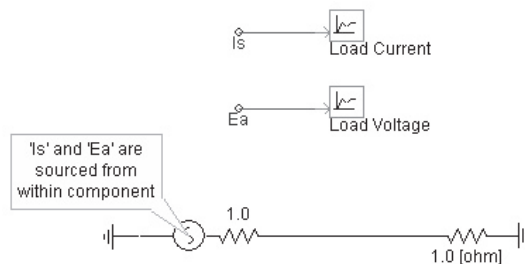
```
1 #SUBROUTINE EMTDC_1PVSRC Single Phase Source
2 #STORAGE LOGICAL:1 REAL:8 INTEGER:6
3 !
4     CALL EMTDC_1PVSRC($SS,$BRN,$tr,1,$Vp,$f_rad,0.0)
5 !
6 #OUTPUT REAL Isource {$CBR:BRN}
7 #IF gnd == 1
8 # OUTPUT REAL Vsource {$VDC:NA:G}
9 #ELSE
10 # OUTPUT REAL Vsource {$VDC:NA:NB}
11 ENDDIF
```

Any Text Field with a Symbol Name identical to those defined above (i.e. 'Isource' and 'Vsource'), will act as Internal Output Variable ports for the component. Any variable name entered into either the 'Source Current' or 'Source Terminal Voltage' Text Fields in the Parameters section will create signals which, when connected to an Output Channel, can be plotted.

Save your component changes by Returning to Circuit View. Save the Project.

Running Your Component

Keeping the test circuit used in the Testing Your New Component section, remove the ammeter and voltmeter from the circuit. Replace these two signals by entering 'Vs' and 'Is' into the Internal Output Variable Text Fields that were created in the component Category page.



Run the case again. You should see identical results as before in Testing Your New Component.

PSCAD Script

PSCAD Script was originally introduced back in PSCAD V2, primarily to provide a more flexible, compiler independent work environment for the programming of custom components. It is recommended that Script be used whenever possible to ensure that component definitions are not effected by the compiler.

Script is given the highest priority by the PSCAD pre-compiler. That is, Script is considered first before all other code, and is hence used by the pre-compiler for expression evaluation, substitutions and to generate formatted Fortran code for the Fortran File or Electric Network Data for the Data File. By using the available Script Directives, Script Operators, and other features available, the appearance, structure and functionality of a component can be manipulated before the greater project compilation is performed.

This chapter is closely linked with Chapter 9 - Designing Components. It is recommended that you read both topics before attempting to design your own component. The following sections describe the available PSCAD Script functions, and provide examples on how each can be used.

NOTE: If the need arises, subroutines or functions (written in Fortran or C) can be called directly, using standard Fortran call statements. Any standard Fortran code entered directly must be preceded by six empty spaces, according to formatted Fortran standards. See the section entitled 'Fortran Guidelines for EMTDC' in the EMTDC Manual for more details on coding rules.

SUBSTITUTIONS

Substitutions are a fundamental element of component design that can appear in all parts of a Component Definition. Substitutions provide a means to communicate information between Definition sections, bring a dynamic aspect to component graphics, and help in formatting code and Comments. The substitution operators available in PSCAD are as listed below:

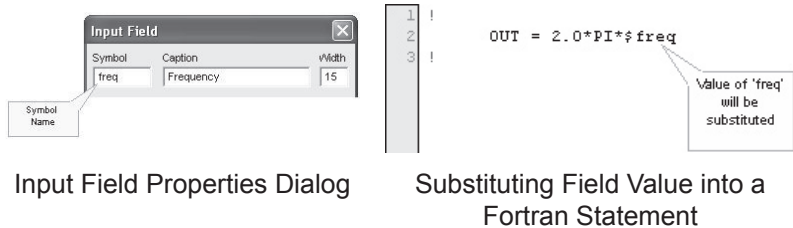
Chapter 10: PSCAD Script

- Value Substitution Prefix: **\$**
- Data Substitution Prefix: **%**
- Braces: **{ }**
- Comment Operator: **!**

\$ Value Substitution Prefix Operator

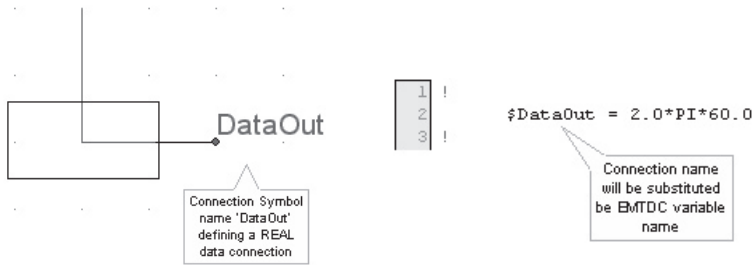
The \$ Value Substitution Prefix Operator (or simply '\$') is the most important of the substitution operators, and is used quite heavily in component design. The \$ operator is normally appended to a character string, where the string may represent a numerical value, or even other text, that has been defined at some previous point in the evaluation process. Regardless of what the character string represents, its value will be substituted in place of the string, whenever a \$ symbol prefixed to it.

There are numerous ways in which the \$ operator may be utilized. For example, the actual data value entered by a user into an Input Field may be substituted into a Fortran statement by prefixing \$ to the Symbol of the field.



NOTE: The value substituted is the value in terms of the default unit specified in the Input Field properties dialog. If the actual entered value is entered in another unit, then this value will be processed first, before substitution takes place. See the Unit System for more details on units.

Another example is the substitution of a one variable name by another. In this case, a Connection defined in the component Graphic is used in a Fortran statement preceded by a \$ prefix. This ensures that the Connection Symbol name will be substituted with a pre-defined EMTDC variable name, compatible with the rest of the Fortran file for the Module where the Component Instance resides.



Component Graphic Containing
a Connection

Substituting Connection
Name into a Fortran
Statement

In an effort to standardize the substitution syntax, the parser has been extended (beginning in PSCAD V4.2) to include a more formal way of handling substitutions from other contexts. This comes from the need to pull information from outside the Component Instance framework. By inserting delimiting symbols directly following the \$ operator, text handling can be easily controlled. The standard syntax is as follows:

$\$(\langle\text{Context}\rangle:\langle\text{Key}\rangle)$

The substitution contains both a $\langle\text{Context}\rangle$ and item $\langle\text{Key}\rangle$ separated by a Scope Operator and delimited by round brackets, where:

- **$\langle\text{Context}\rangle$** : The context that is to be used to scope the item Key. See below for a list of valid Context names.
- **$\langle\text{Key}\rangle$** : Item key (actual substitution) that is to be placed at substitution point. See below for a list of valid Key names.

The \$ Value Substitution Prefix Operator may be used in Text Labels within the Graphic Section, as well as the Fortran, DSDYN, DSOUT, Branch, Model-Data, Matrix-Fill, Transformers, or T-Lines segments of the Script Section. In addition to this, the \$ operator may also be used to substitute Global Constants and various Keys under their respective Contexts.

'\$' may **only precede** variable names defined as one of the following:

- A Connection Symbol name
- An Input Field, Text Field, or Choice Box Symbol name

Chapter 10: PSCAD Script

- Any variable defined in the Computations Segment
- A Global Constant (within input parameters only)
- A Key name

NOTE: If a new variable needs to be defined directly within your Fortran code, you must define it as a local variable. See the '#LOCAL Directive' section in this chapter for more details.

Valid Context and Key Names

The following table lists all valid context names that may be used in place of **<Context>** and **<Key>** as described above:

Context Name (case sensitive)	Description	Available Keys
<none>	An empty context key is automatically context of the Component Instance.	
Defn	Context is that of the Definition of the Component Instance.	Name, Desc, Path
Project	Context is that of the Project that contains the Component Instance.	Name, Desc, Version, Author, Path
Session	Context is that of the current running session of the application (PSCAD).	Name, Version
System	Context is that of the operating system.	Name, Version, User

Local Context Short Form

When using substitutions within the Context of the Component Instance, there is no need to include the <Context> identifier and it hence may be excluded along with the Scope Operator. In such cases, the syntax is:

$\$(\text{<Key>})$ or simply $\$\text{<Key>}$

PSCAD

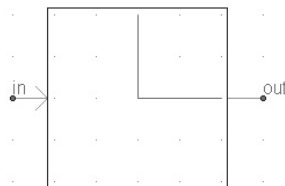
This syntax is backwardly compatible with PSCAD version previous to V4.2 and the simple un-delimited form is still the one that is used in most cases (although both forms are now legal). The delimited version has special properties in that it can be used to format identifiers or other fields where there is an alphanumeric post-fix. This is a new capability and by design, if the character immediately following the \$ operator is the same, then that character is copied directly to the stream without processing. This allows for '\$' insertions, for example, to insert a dollar-sign character, use \$\$.

Substituting Component Constants into Script

One form of possible substitution is that of pre-processed constants into component Fortran code. Here, the actual numerical value of the constant (i.e. Connection, Input Parameter or any variable defined in the Computations Segment) will be directly substituted upon compilation by PSCAD. Substitutions may also be performed within Commented Script, in case the user would like to display certain pre-defined values in the compiled Fortran files for the Project.

EXAMPLE 10-1:

A user component utilizes an external Fortran function, which requires an angle input argument in degrees to convert to radians. The data input in this case is a Connection, defined in the Graphic Section of the component definition, and has a Symbol Name 'in.' The function output 'out' is defined as an output Connection in the Graphic Section.



Component Graphic Section View

By using the \$ Variable Substitution Prefix Operator, the Fortran variables used by EMTDC to represent the Connections 'in' and 'out' can be inserted directly into Fortran code. The function call would appear

Chapter 10: PSCAD Script

similar to the following in either the Fortran, DSDYN or DSOUT segments:

```
1 #FUNCTION REAL PH_CON Phase Angle Converter
2 !
3     $out = PH_CON($in)
4 !
5 ! When this Script is compiled by PSCAD, $in will be substituted
6 ! by whatever EMTDC decides to name the data node to which 'in' is
7 ! connected (ex. RT_1). $out will substituted by whatever EMTDC
8 ! decides to name the data node to which 'out' is connected
9 ! (ex. RT_2).
10 !
```

Fortran Segment Code

NOTE: See the #FUNCTION Directive for more details on declaring Fortran functions.

EXAMPLE 10-2:

A user-defined component definition called 'my_new_comp' has been programmed to display the contents of certain key Input Fields, along with the definition name, within Comments provided in the component Fortran segment, namely: Frequency with Symbol 'freq,' value 60.0 and units in Hz, and Voltage with Symbol 'volts,' value 120.0 and units in kV. Note that both 'freq' and 'volts' are local variables and so the Local Context short form is used.

The screenshot shows the 'Input Field' dialog box for the variable 'freq'. The Symbol is 'freq' and the Caption is 'Frequency'. The Data Type is 'Real'. The Options are 'Show Caption' (checked) and 'Allow Signal Names' (unchecked). The Default value is 60.0, Units are Hz, Min is 0.01, and Max is 1e+038.

Input Field Property Settings
(‘freq’)

The screenshot shows the 'Input Field' dialog box for the variable 'volts'. The Symbol is 'volts' and the Caption is 'Voltage'. The Data Type is 'Real'. The Options are 'Show Caption' (checked) and 'Allow Signal Names' (unchecked). The Default value is 120.0, Units are kV, Min is -1e+038, and Max is 1e+038.

Input Field Property Settings
(‘volts’)

PSCAD

When the component is evaluated, the Symbol names ('freq' and 'volts') are declared as variables internal to the component, and are given the value entered in their respective fields. In this case, *freq* = 60.0 and *volts* = 120.0. The user is now free to insert these variables elsewhere within the component code by using the \$ Variable Substitution Prefix Operator.

Something similar to the following comments could appear in either the Fortran, DSDYN, or DSOUT segments:

```
1 ! User-Defined Component: $(Defn:Name)
2 !
3 ! Frequency: $freq Hz, Voltage: $volts kV
4 !
```

Fortran Segment Code

```
1 ! User-Defined Component: my_new_comp
2 !
3 ! Frequency: 60.0 Hz, Voltage: 120.0 kV
4 !
```

PSCAD Fortran File Code
following Compilation

This feature is excellent for ensuring the Project Fortran File code is always properly commented.

NOTE: The value substituted is the value in terms of the default unit specified in the Input Field properties dialog. If the actual entered value is entered in another unit, then this value will be processed first, before substitution takes place. See 'Unit System' in Chapter 5 for more details on units.

Substituting Component Constants into Text Labels

Pre-processed constants can be substituted directly into Text Labels in the Graphic Section using the \$ Variable Substitution Prefix Operator. This is a convenient feature, as it allows users to change text appearance on the component graphic according to input parameters.

EXAMPLE 10-3:

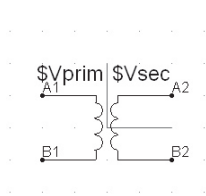
A user has created a single-phase, two-winding transformer component and wishes to display both the primary and secondary winding voltage ratings directly on the component graphic. The primary

Chapter 10: PSCAD Script

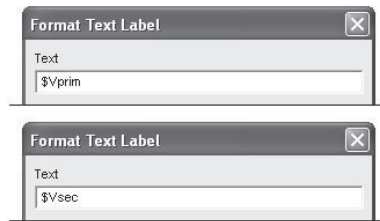
voltage rating input parameter 'Vprim' is entered as 25.0 [kV] and the secondary input 'Vsec' is set at 4.0 [kV] as shown below:

vWinding #1 voltage (RMS)	25.0 [kV]
vWinding #2 voltage (RMS)	4.0 [kV]

In the component Graphic Section, the user adds two new Text Labels (positioned accordingly) and enters '\$Vprim' and '\$Vsec' in the **Text** field of the two Text Label Properties dialogs as shown below:

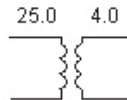


Component Graphic



Text Label Property Settings

When the component is now viewed in Circuit view, values of the input parameters 'Vprim' and 'Vsec' should appear directly on the component graphic. For this example,



This feature is an excellent method of displaying key component parameters directly in the circuit. See 'Text Labels' in Chapter 9 for more details on adding and editing Text Labels.

% Data Substitution Prefix Operator

The % Data Substitution Prefix Operator (or simply '%') can be used to make a simple text substitution into Fortran Comments, as well as into Text Label objects. Data substitution is very similar to the \$ Value Substitution Prefix Operator, except that it substitutes the exact contents of an input parameter as text, before it is processed.

PSCAD

The '%' operator may be used in Text Labels within the Graphic Section, as well as within Commented code in the Fortran, DSDYN, DSOUT, Branch, Model-Data, Matrix-Fill, Transformers, or T-Lines segments of the Script Section.

'%' should **only precede** variable names defined as Input Parameters. Bracket delimiters (i.e. %(...)) may be used when post-fix formatting is required.

Substituting Text into Commented Script

One form of substitution is that of text into component Fortran Comments. Unlike the \$ Value Substitution Prefix Operator, the entire, unprocessed contents of an input parameter (including the units) will be directly substituted upon component evaluation.

EXAMPLE 10-4:

Consider Example 10-2 above, but replace the \$ Value Substitution Prefix Operator with the % Data Substitution Prefix Operator. The following comments could then appear in either the Fortran, DSDYN or DSOUT segments:

```
1 | User-Defined Component: $(Defn:Name)
2 |
3 | Frequency: $freq Hz, Voltage: $volts kW
4 |
```

Fortran Segment Code

```
1 | User-Defined Component: my_new_comp
2 |
3 | Frequency: 60.0 Hz, Voltage: 120.0 kW
4 |
```

PSCAD Fortran File Code
following Compilation

Substituting Text into Text Labels

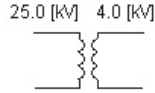
Input parameter text can be substituted directly into Text Labels in the Graphic Section using the % Data Substitution Prefix Operator. This is a convenient feature, as it allows users to change text appearing on the component graphic.

EXAMPLE 10-5:

Consider Example 10-4 above, but replace the '\$' operator with the % Data Substitution Prefix Operator in the new Text Labels. When

Chapter 10: PSCAD Script

the component is now viewed in Circuit view, values of the input parameters 'Vprim' and 'Vsec' should appear on the component graphic as follows:



{ } Braces

Braces can be used to perform mathematical, logical and formatting operations before other Script is considered by the compiler. Braces can appear all over the component Definition. There are two main forms in which Braces may be used: Anonymous Expression Substitution and Block Processing. The Brace syntax is as follows:

[\${<Expression>}

Where:

- **\$**: Optional. The \$ Value Substitution Prefix Operator is not required in some instances, such as in most Block Processing operations. The important thing to remember is that the \$ Operator should be included if some sort of operation is to be performed within the Braces before the contents are substituted.
- **<Expression>**: Can be a mathematical expression, or simply a block of text.

It is important to note that mathematical and logical operations within Braces are performed prior to the insertion of code into the EMTDC System Dynamics (or the electrical system Data File). Subsequently, these operations cannot involve variables – only constants (such as constant input parameters, local constants, or constants defined in the Computations segment).

Although simple examples are provided here, you can find more examples by looking at the Definitions of Master Library components.

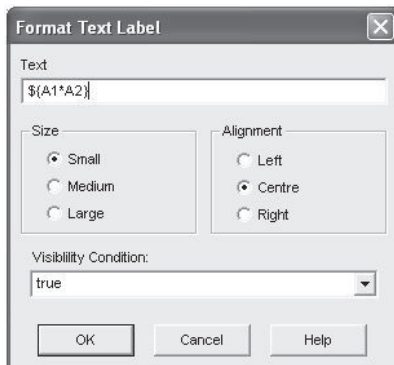
Anonymous Expression Substitution

EXAMPLE 10-6:

The following example shows how expression braces can be used within Text Labels to perform a mathematical operation on substituted values from the \$ Variable Substitution Prefix Operator. Say that two input variables are entered as 'A1 = 2.0' and 'A2 = 3.0.' The user wants to display a value based on these values directly on the component graphic as follows:

$$\text{Text Label} = A1 * A2 = 6.0$$

The following could then be entered into the Text Label Properties dialog:

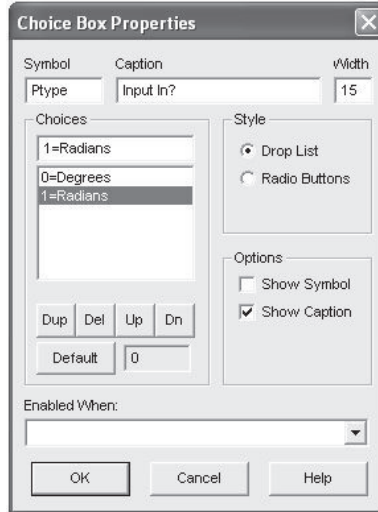


Block Processing

EXAMPLE 10-7:

A user has modified the Fortran function PH_CON in Example 10-1 to include phase angle conversion in both directions (i.e. degrees to radians and radians to degrees). In the component Parameters section, a Choice Box with Symbol 'Ptype' is designed with two options as shown below:

Chapter 10: PSCAD Script



Choice Box Property Settings ('Ptype')

The function PH_CON is modified to include a second argument to indicate the type of conversion required, based on the numerical value of 'Ptype.' The additional argument is a character string, where 'D2R' indicates degrees to radians conversion and 'R2D' radians to degrees. Braces are used in formatting the Fortran code, along with a #CASE Conditional Directive and ~ Line Continuation Operator:

```
1 #FUNCTION REAL PH_CON Frequency Converter
2 !
3     $out = PH_CON($in,-
4 !
5 #CASE FType {'-D2R'}) {'-R2D'})
6 !
```

Fortran Segment Code

```
1 !
2     RT_2 = F_CON(RT_1,'D2R')
3 !
```

PSCAD Fortran File Code
following Compilation
(if 'Ptype' = 0)

'RT_1' and 'RT_2' are variable substitutions given by EMTDC. Their names may vary.

! Comment Indicator

The exclamation point '!' is used throughout the component definition segments to indicate that a particular line of code is a comment. All

PSCAD

lines of script, which is preceded by '!', will be written as standard Fortran comments in the project Fortran or Data Files.

NOTE: Commented code that is written to the data file, requires that the Comment Indicator be in the first column from the left.

EXAMPLE 10-8:

Comments should be used to describe the user's code, as well as to provide spacers and section breaks in order to enhance the readability of the user's code. The '!' comment indicator may be used everywhere except the Branch segment.

The following example illustrates the use of comments in the Fortran or DSDYN segments in a user component definition:

```
! MY FIRST PSCAD SCRIPT
! -----
!
! Storage:
!
#STORAGE REAL:1
!
! Main body of script:
!
```

NOTE: See the #STORAGE Directive for more details.

SCRIPT DIRECTIVES

Script directives are used to facilitate the creation of Fortran code. Each directive type has a specific purpose, as explained in the following sections. Script directives always begin with a pound '#' symbol prefix. Spaces before and after the '#' symbol are allowed for flexibility in code structure.

Chapter 10: PSCAD Script

#STORAGE

This directive was first introduced with PSCAD V3 in order to provide custom component compatibility with the Fortran 90 dynamic dimensioning feature, also introduced in V3. Its purpose is to specify what type and how many storage elements should be dynamically allocated for a specific component.

The #STORAGE directive is closely related to the EMTDC STORx internal variables. That is, if EMTDC STORx variables are used in component code, they must be specified here to ensure that the component storage is properly dimensioned by PSCAD.

The #STORAGE directive is used only in the Fortran, DSDYN or DSOUT segments, and should appear as follows:

```
#STORAGE <TYPE>:<Number>
```

<TYPE> can be one of four types of STORx variable. These are REAL, INTEGER, LOGICAL and COMPLEX. <Number> is the amount of elements to be stored.

NOTE: If using a Fortran 90 compiler, the #STORAGE directive must be included for dynamic memory allocation.

EXAMPLE 10-9:

A user's custom component code specifies that 10 REAL, 4 INTEGER and 1 LOGICAL variable are to be stored each time step. The user wants to ensure that storage is dynamically allocated by PSCAD.

The storage statement should appear as follows, in either the Fortran, DSDYN or DSOUT segments:

```
#STORAGE REAL:10 INTEGER:4 LOGICAL:1
```

The actual #STORAGE statement can appear anywhere in script, but should be placed near the top.

PSCAD

All three variable types above do not need to appear in the #STORAGE statement if they are not needed. For example, if only REAL variables need to be stored, only the REAL type needs to be specified.

#LOCAL

This directive is used to define any local variables that are not classified as pre-defined. That is, any variables introduced within the code that do not require the pre-defined variable prefix '\$.'

The #LOCAL directive is used only in the Fortran, DSDYN or DSOUT segments, and should appear as follows:

```
#LOCAL <TYPE> <Name> <Array_Size>
```

<TYPE> can be either REAL, INTEGER, or LOGICAL. <Name> is the given name for the local variable. <Array_Size> is an optional integer that defines the size of the array. If the variable has only a single dimension, then leave <Array_Size> blank.

<p>NOTE: Care should be exercised when using #LOCAL. Make sure to give your variable a unique name, so as to avoid the same name being used in different variable types.</p>

EXAMPLE 10-10:

A user's custom component requires two local variables for use as subroutine arguments. According to a conditional statement based on a pre-defined variable 'A,' a local INTEGER variable 'MY_X' and a local REAL array 'Error' are defined before the subroutine is called.

The local variable definitions should then appear as follows in the Fortran, DSDYN or DSOUT segments:

Chapter 10: PSCAD Script

```
#LOCAL INTEGER MY_X
#LOCAL REAL Error 2
!
#IF A > 1
    MY_X = 1
    Error(1) = 0.2
#ELSE
    MY_X = 0
    Error(2) = 0.8
#ENDIF
!
    CALL SUB1(MY_X, Error)
!
```

NOTE: A > 1 is a logical expression. See Expression Evaluation for more details.

#FUNCTION

This directive is used to declare the existence of a function, and the argument type it returns. #FUNCTION is mandatory if a function is called within the component definition.

The #FUNCTION directive is used only in the Fortran, DSDYN or DSOUT segments, and should appear as follows:

#FUNCTION <TYPE> <Name> <Description>

<TYPE> can be either REAL, INTEGER, or LOGICAL. <Name> is the given name of the function. <Description> will be included as a comment line near the beginning of the EMTDC Fortran File.

EXAMPLE 10-11:

The Hard Limiter component in the PSCAD Master Library utilizes a REAL function called 'LIMIT' to determine the output signal to be sent to an external connection 'O,' according to pre-defined inputs 'LL,' 'UL,' and 'I.'

PSCAD

The following code appears in the Fortran segment of the Hard Limiter component definition:

```
#FUNCTION REAL LIMIT Hard Limiter
!  
    $O = LIMIT($LL, $UL, $I)  
!
```

#SUBROUTINE

This directive is used to provide a description for a subroutine that is being called from the component. #SUBROUTINE is used simply for cosmetic purposes, and is not mandatory (although its use is recommended anyway).

The #SUBROUTINE directive is used only in the Fortran, DSDYN or DSOUT segments, and should appear as follows:

```
#SUBROUTINE <Name> <Description>
```

<Name> is the given name for the subroutine. <Description> will be included as a comment line near the beginning of the EMTDC Fortran File.

EXAMPLE 10-12:

A user includes a call to a subroutine, named 'SUB1,' within the component code. It is suggested that a description be added for clarity.

The following code should appear in the Fortran, DSDYN or DSOUT segment of the component:

```
#SUBROUTINE SUB1 User Subroutine
!  
    CALL SUB1($X, $Y, $Z)  
!
```

Where 'X', 'Y' and 'Z' are pre-defined variables.

Chapter 10: PSCAD Script

#OUTPUT

This directive was first introduced in PSCAD V3 and was added to provide a method of extracting certain quantities for the purpose of monitoring, plotting, or for use externally by other components.

#OUTPUT performs two tasks: It first defines a variable according to a given name, and then gives it a value according to an expression.

The #OUTPUT directive is used only in the Fortran, DSDYN or DSOUT segments, and should appear as follows:

```
#OUTPUT <TYPE> <Name> <Array_Size> {<Expression>}
```

<TYPE> can be either REAL, INTEGER, or LOGICAL. <Name> is the given name for the variable. <Array_Size> is an optional integer, which defines the size of the array. If the variable has only a single dimension, then leave <Array_Size> blank. <Expression> can be a mathematical expression, a storage location, or simply a defined variable.

EXAMPLE 10-13:

The manner in which the value of the named output variable may be defined, can appear in many different ways. The following list shows some of the possibilities:

```
! Defines a REAL variable 'freq' and substitutes
! the value of a pre-defined variable 'Fout.'
!
#OUTPUT REAL freq {$Fout}
!
! Defines an INTEGER variable 'Xon' and assigns
! it the value of a storage location.
!
#OUTPUT INTEGER Xon {STORI(NSTORI+1)}
!
! Defines a REAL variable 'POut' and assigns
! it the value of a given mathematical
! expression.
!
#OUTPUT REAL POut {$V*$I}
!
```

#TRANSFORMERS

This directive should be included whenever mutually coupled coils are to be represented in a model. Examples of Master Library models that use this directive are (of course) the transformers and the PI Section components.

#TRANSFORMERS has two primary purposes:

1. Provides a method to sequentially number all models that contain mutually coupled coils for the purpose of dynamic dimensioning (when using a Fortran 90 compiler).
2. The same number provides addressing information for the monitoring of mutually coupled winding currents. Transformer winding currents are measured using the EMTDC CDCTR(M,N) matrix. CDCTR(M,N) is the current through the Mth winding of the Nth transformer.

The #TRANSFORMERS directive is used only in the Transformers segment, and should appear as follows:

#TRANSFORMERS <Number>

<Number> indicates the total number of transformers within the component.

NOTE: Assigning #TRANSFORMERS correctly is important even if you are not using the dynamic version of EMTDC, as the information for monitoring winding currents is still needed.

EXAMPLE 10-14:

The 3-Phase, 2-Winding classical transformer component, located in the Transformers section of the Master Library, consists of three, 1-Phase transformers.

The directive would appear as follows in the Transformers segment:

```
#TRANSFORMERS 3
```

#WINDINGS

This directive was first introduced in PSCAD V4.0.2, in order to remove a fixed limitation on the number of coupled coils in a transformer, which was previously set to 10.

#WINDINGS can be used to assign the number of coupled windings in a transformer. PSCAD will look for the highest number associated with all existing #WINDINGS directives and then assigns that number (as the maximum number of windings) to the Map (.map) file.

The #WINDINGS directive is used only in the Transformers segment, and should appear as follows:

```
#WINDINGS <Number>
```

<Number> indicates the maximum number of coupled windings within that specific component.

EXAMPLE 10-15:

The 3-Phase, 2-Winding classical transformer component, located in the Transformers section of the Master Library, consists of three, 1-Phase transformers.

The directive would appear as follows in the Transformers segment:

```
#WINDINGS 2
```

#IF, #ELSEIF, #ELSE, #ENDIF DIRECTIVES

#IF, #ELSEIF, #ELSE, #ENDIF directives are a special type of script directive that provide one way for the user to alter the component definition, according to component input parameters, before compilation.

PSCAD

These directives can be used to change the electrical configuration, or act as standard Fortran IF-THEN-ELSE keywords in the EMTDC System Dynamics (i.e. DSDYN and DSOUT). Similar to normal script directives, #IF, #ELSEIF, #ELSE, #ENDIF directives always begin with a '#' prefix.

#IF, #ELSEIF, #ELSE, #ENDIF directives can be used in all segments and should appear as follows:

```
#IF <Logic>
  ...Application_Code...
#ELSEIF <Logic>
  #IF <Logic>
    ...Application_Code...
  #ELSE
    ...Application_Code...
  #ENDIF
#ELSE <Logic>
  ...Application_Code...
#ENDIF
```

If just a simple IF-THEN condition is required, the following shortcut using expression braces may be used:

```
#IF <Logic> {<Expression>}
```

<Logic> is a logical expression using logical operators. <Expression> can be a variable definition.

EXAMPLE 10-16:

A user needs to change the output of a signal generator model, according to whether a sine or cosine output is required. The component definition provides an input parameter choice box in the Parameters section, called 'Type.' If this parameter is 1, then the output is sinusoidal (if 0 then co-sinusoidal).

The following code should appear in the Fortran, DSDYN or DSOUT segments of the component:

Chapter 10: PSCAD Script

```
!  
! Signal Generator  
!  
#IF Type == 1  
    $OUT = SIN(TWO_PI*$F)  
#ELSE  
    $OUT = COS(TWO_PI*$F)  
#ENDIF  
!
```

where 'F' is a pre-defined variable and 'OUT' is an output Connection in the Graphic section.

NOTE: 'Type == 1' is a logical expression. See the section entitled 'Expression Evaluation' in this chapter for more details.

Using Expression Braces, the above example could also be written as:

```
!  
! Signal Generator  
!  
#IF Type == 1 {          $OUT = SIN(TWO_PI*$F) }  
#IF Type != 1 {         $OUT = COS(TWO_PI*$F) }  
!
```

EXAMPLE 10-17:

A user has created an electrical component that can either be a simple resistor, inductor or capacitor. The component definition provides an input parameter choice box in the Parameters section, called 'Type.' This parameter can either be 1, 2 or 3 to represent a resistor, inductor or capacitor respectively. Three other input text boxes also exist (called 'R', 'L' and 'C') in order to specify the respective values of these elements.

The following code should appear in the Branch segment of the component:

PSCAD

```
#IF Type == 1
    $N1 $N2 $R 0.0 0.0
#ELSEIF Type==2
    $N1 $N2 0.0 $L 0.0
#ELSE
    $N1 $N2 0.0 0.0 $C
#ENDIF
```

where 'N1' an 'N2' are electrical Connections in the Graphic section.

Using Expression Braces, the above example could also be written as:

```
!
#IF Type == 1 {$N1 $N2 $R 0.0 0.0}
#IF Type == 2 {$N1 $N2 0.0 $L 0.0}
#IF Type == 3 {$N1 $N2 0.0 0.0 $C}
!
```

~ LINE CONTINUATION OPERATOR

Occasionally, it may be desirable to split a segment line over multiple lines. This is particularly useful when a line is very long, or if a variable in a central portion of the line may vary according to conditional statements.

Line continuations are used in PSCAD Script mostly for clarity. That is, PSCAD will automatically break segment lines that exceed the formatted Fortran standard of 72 columns anyway, and so the user does not normally need to worry about this. However, to ensure that code is easily readable to everyone, line continuations should be used.

The line continuation operator is used as follows: A line ending in the '~' symbol will be joined with the next line if it starts with a '~' symbol, as well. This processing is performed after conditional statements have been interpreted and is used in formatting the text for output into the Fortran or data files.

The line continuation operator can appear anywhere on the segment line, and it can be used with the #CASE Conditional Directive.

EXAMPLE 10-18:

In Example 10-16, conditional statement directives were used in order to determine the output of a signal generator model. As a simple illustration of how line continuations can be used, the code in Example 10-16 has been re-written below:

```
!  
! Signal Generator  
!  
          $OUT = ~  
#IF Type==1  
~SIN~  
#ELSE  
~COS~  
#ENDIF  
~ (TWO_PI*$F)  
!
```

If Type is “2,” after conditional statement processing and line continuation, the PSCAD Script would appear as follows, before being transformed into formatted Fortran:

```
!  
! Signal Generator  
!  
          $OUT = COS (TWO_PI*$F)  
!
```

#CASE CONDITIONAL DIRECTIVE

The #CASE conditional directive is essentially a short-form notation method to be used instead of #IF, #ELSEIF, #ELSE, #ENDIF directives. It is used in conjunction with the Line Continuation Operator method to provide a space saving alternative, especially when a great number of #IF, #ELSEIF, #ELSE, #ENDIF Directives are required.

The #CASE directive can be used in all segments, and should appear as follows:

PSCAD

`#CASE <Expression> {<Clause_0>} {<Clause_1>} ...`

<Expression> must return an integer from 0 to n, and can either be a mathematical expression or simply a defined variable. <Clause_n> represents what is to occur given the value of the expression, and where in the sequence of clauses it resides. For example, the results of <Clause_0> will occur if <Expression> is equal to 0. If <Expression> is equal to 1, <Clause_1> will occur, etc.

EXAMPLE 10-19:

Once again referring to Examples 10-16, a simple illustration of how the `#CASE` directive can be used instead of the `#IF`, `#ELSEIF`, `#ELSE`, `#ENDIF` directives.

```
!  
! Signal Generator  
!  
    $OUT = ~  
#CASE Type {~COS~} {~SIN~}  
~(TWO_PI*$F)  
!
```

Note that 'Type' can be either 0 or 1.

EXAMPLE 10-20:

The following illustrates how the `#CASE` directive is used in the Transformers segment of the 3-Phase, 2-Winding classical transformer component, located in the Master Library.

Here, 'YD1' and 'Lead' are both input parameters. 'YD1' can be either 0 or 1 and 'Lead' is either 1 or 2. Multiplying of these two integers can result in 0, 1 or 2, and this information is used to determine which clause to select:

```
!  
#CASE YD1*Lead {$A1 $G1~} {$A1 $B1~} {$A1 $C1~}  
!
```

EXPRESSION EVALUATION

In addition to various arithmetic and logical operators, PSCAD possesses the ability to perform a limited set of mathematical functions.

Mathematical expression evaluation is used mostly in the Computations segment, while Arithmetic and logical operators are used throughout the entire component definition.

Mathematical Functions

Mathematical computations can be performed on input parameters, input signals, or on results of other computations.

The table below lists all of the available mathematical functions:

Function	Description
SIN(x)	Sine function
COS(x)	Cosine function
TAN(x)	Tangent function
ASIN(x)	Inverse sine function
ACOS(x)	Inverse cosine function
ATAN(x)	Inverse tangent function
SINH(x)	Hyperbolic sine function
COSH(x)	Hyperbolic cosine function
TANH(x)	Hyperbolic tangent function
LOG(x)	Natural logarithm
EXP(x)	Exponential
LOG10(x)	Base 10 logarithm
SQRT(x)	Square root
ABS(x)	Absolute value
REAL(x)	Real part of complex number
IMAG(x)	Imaginary part of complex number
NORM(x)	Norm of complex number ($x^2 + y^2$)
CEIL(x)	Rounds fraction to next upper integer
FLOOR(x)	Rounds fraction to next lower integer

ROUND(x)	Adds 0.5 to a REAL value and then performs an INT function (i.e. ROUND(X) = INT(X+0.5)). Do not use with negative numbers
INT(x)	Removes fractional part of REAL value (right side of decimal)
FRAC(x)	Removes integer part of REAL value (left side of decimal)
RAND(x)	Random value between 0 and x
P2RX(m,θ)	Polar to rectangular conversion (θ in degrees)
P2RY(m,θ)	Polar to rectangular conversion (θ in degrees)
R2PM(x,y)	Rectangular to polar conversion
R2PA(x,y)	Rectangular to polar conversion

Arithmetic Operators

Listed below are the arithmetic operators available in PSCAD. These operators are used to perform mathematical calculations primarily in the Computations, Fortran, DSDYN and DSOUT segments.

Function	Description
+	Add
-	Subtract
*	Multiply
/	Divide
%	Remainder
**	Raise to power
\	Parallel (xy) / (x +y)

Logical Operators

Listed below are the logical operators available in PSCAD. Logical expressions are primarily used in conjunction with #IF, #ELSEIF, #ELSE, #ENDIF Directives and the Ternary Operator. They are also used in the Checks segment.

NOTE: Logical expressions will return a value of 1 if true, and a value of 0 if false.

Chapter 10: PSCAD Script

Function	Description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
	OR
&&	AND

EXAMPLE 10-21:

Logical Operators can appear in many different areas. The following examples illustrate the various ways these operators can be used:

```
!  
! ...with #IF, #ELSEIF, #ELSE, #ENDIF Directives  
!  
#IF F >= 60.0  
    Fout = 60.0  
#ENDIF  
!  
! ...with a Ternary Operator in the Computations  
!     segment  
!  
REAL L = X == 0.0 ? Y*2.0 : Y/3  
!  
! ...in the Checks segment  
!  
ERROR Value too small : R > 0.001  
!
```

See the Ternary Operator below for more details

Ternary Operator

The ternary operator is yet another short form method offered in PSCAD Script for representing an IF-ELSE-ENDIF type

PSCAD

expression. It allows the user to define a variable, according to certain conditions, in a single line.

The ternary operator is used only in the Computations segment, and should appear as follows:

`<Logic> ? <Value_if_True> : <Value_if_False>`

`<Logic>` is a logical expression using logical operators. `<Value_if_True>` and `<Value_if_False>` can either be a single constant, or a mathematical expression.

NOTE: Care must be taken when designing components with ternary operators: Ensure that variables used within the ternary expression will not be disabled under otherwise valid conditions. In other words, unrelated logic may disable one or more of the variables used within the ternary expression, which may render the ternary result invalid. Input variables are enabled/disabled by way of Conditional Statements, Layers and Filters.

EXAMPLE 10-22:

A user wants to define a REAL variable 'X' in the Computations segment of a component definition. The value of 'X' is to be equal to 1.0 if an input parameter 'N' is 2 or 3, or defined by a mathematical expression otherwise.

The following code should appear in the Computations segment of the component definition:

```
REAL X = (N==2 || N==3) ? 1.0 : SQRT(2)*V
```

where 'V' is a pre-defined constant.

Chapter 10: PSCAD Script

EXAMPLE 10-23:

A user wants to define a REAL variable 'Torq' in the Computations segment of a component definition. The value of 'Torq' is defined by a mathematical expression, where one element of this expression varies according to a condition. This can be accomplished by using the ternary operator as follows:

```
REAL Torq = (X > 1 ? 0.0 : Tm) + Te*100
```

where 'X,' 'Tm' and 'Te' are pre-defined constants.

Debugging and Finalizing

Ironing out the bugs in a Project can be the most challenging part of the design process. No matter how carefully a system is constructed, there are almost always a few errors or warnings issued when the Project is compiled and run. This chapter describes the reasons for some of the most common errors and warnings, along with how to deal with them.

Debugging a Project can become even more difficult when user-written code is present. PSCAD does not include a debugging mechanism for user code, and so an integrated debugger must be used. Some methods of accomplishing this are described in this chapter.

In finalizing a Project, the user may want to protect (i.e. pre-compile) source code when supplying projects and components to other users. A section is included in this chapter, which describes how to pre-build any source code into a library (*.lib) file. These files can then be provided along with the Project without running the risk of giving away the secrets of your hard work.

COMPILING PROJECTS

PSCAD does not include a full time error messaging system, and so in order to view any warnings or errors, you must first Compile and Build your Project. As a matter of fact, it is always a good idea to compile your project first, before actually running the simulation. This is especially true for the first run.

See 'Compiling and Building a Project' and/or 'Compiling an Individual Module' in Chapter 5 for more on this.

COMMON OUTPUT WINDOW MESSAGES

There are a large number of possible error and warning messages that can be generated and displayed in the Output Window. Messages can come from PSCAD, EMTDC or individual

Chapter 11: Debugging and Finalizing

components. Most of these messages however, will never occur if a project is judiciously constructed in the first place.

The following descriptions outline the most common occurring messages.

Warning: Suspicious isolated node detected

This warning is issued if PSCAD detects an open electrical circuit connection on a component - usually caused by an electrical node that is not connected to anything.

The suggested method to deal with this warning is to connect a large resistance (approximately 1 M Ω) to ground at the node. This will ensure numerical stability and will not have a great affect on the simulation results.

Parser: Suspicious substitution of empty key '<name>'

This warning is issued if PSCAD detects an empty text input field in a component instance. This warning most commonly occurs when forgetting to name a source or transformer instance, or when a component input parameter field called **Fortran Comment** is left blank. **<name>** is the name of the text input field.

Signal '<name>' type conversion may lose accuracy

This warning is issued if PSCAD detects a REAL signal being sent to an input expecting an INTEGER. PSCAD will automatically convert the REAL signal value to the nearest integer, hence the warning. **<name>** is the name of the signal.

Signal '<name>' source contention -> '<source_1>' and '<source_2>'

This error is issued if PSCAD detects a signal of the same name being generated from more than one source. This error is most commonly caused when component instances with defined internal output variables are copied, hence duplicating the internal output variable. **<name>** is the name of the signal. **<source_1>** and **<source_2>** are the devices generating the signal.

Signal '<name>' dimension mismatch -> <dim_1> != <dim_2>

This error is issued if PSCAD detects that a signal of dimension **<dim_1>** is being sent to an input expecting a signal of dimension **<dim_2>**. This error commonly occurs with the Power Electronic Switch components, which expect a 2-dimensional input gate signal when set for interpolation. **<name>** is the name of the signal.

Signal '<name>' size mismatch at connection '<con>'

This error usually accompanies the previous error message. <name> is the name of the signal, and <con> is the name of the connection to which the signal is being input.

Array '<Name>' cannot be typecasted

This error is issued if PSCAD detects an array signal type mismatch. For example, if a data signal array was defined as type INTEGER and the user attempted to tap off a single element with the Data Signal Array Tap component set to type REAL (or vice-versa). Or, if an array of type REAL is input into a component, where the external input Connection in question is defined as an INTEGER array.

Invalid breakout connection to ground at '<Node>.' Node array elements cannot be individually grounded.

This error is related to the use of the Breakout component: Ground components cannot be directly connected to Breakout terminals. The Breakout is designed specifically for mapping multiple connections on the scalar side to a single array. Since Ground nodes cannot be mapped, the compiler will issue this warning. The suggested work around is to use a Current Meter as a series element between the Breakout terminal and ground.

<Node> is the name of the Breakout reference node connected to ground.

Short in breakout at '<Node>.' Node array elements must be uniquely defined.

This error is related to the use of the Breakout component: The nodes on the 3-phase side of this component are not actual electrical nodes, but references that will assume the node number to which they are connected. This error is posted if these reference nodes are shorted (i.e. electrically connected together). Each node on the 3-phase side of the Breakout component must be unique.

<Node> is the name of the Breakout connection which is shorted.

Branch imbalance between breakouts at '<Node>.' Node array elements cannot be shared between signals.

This error is related to the use of the Breakout component: The nodes on the 3-phase side of this component are not actual electrical nodes, but references that will assume the node number to which they are connected. A special condition that cannot be referenced is

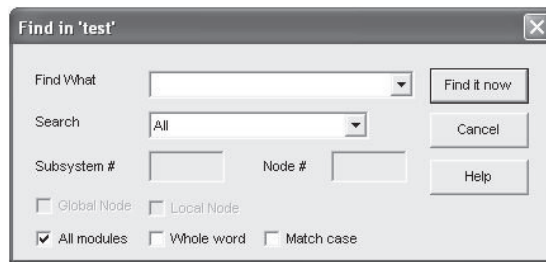
Chapter 11: Debugging and Finalizing

referred to as an 'unbalanced' condition, where the imbalance refers to electrical nodes, not actual impedance. The basic rule to remember here is that all branches on the 3-phase side must include at least one series impedance.

<Node> is the name of the Breakout connection which is shorted.

FIND FEATURE

If the user wishes to search a project for a Data Label, Connection, node number, etc., the Find feature in PSCAD can be used. To bring up the Find dialog window, simply select **Edit | Search...** from the Main Menu bar, press the **Find** button in the Main Toolbar or press **Ctrl + F** on your keyboard, while in Circuit view. Any of the above operations will open the Find dialog.



The various input parameters for this dialog are described below.

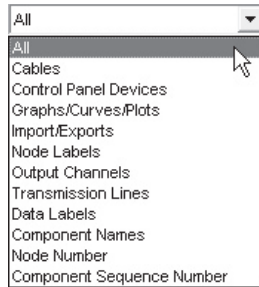
Find What

Enter the character string that you would like to search for. Be sure to check the selection boxes at the bottom of the dialog to modify your search options. If **All Pages** is selected, then Find will search the entire project, as opposed to just the current page. If **Whole Word** is selected, then Find will only look for the Find What string as a whole word, as opposed to part of a word. If **Match Case** is selected, then the search will be case sensitive.

Note that if you are performing a **Node Number** search, this input is disabled.

Search

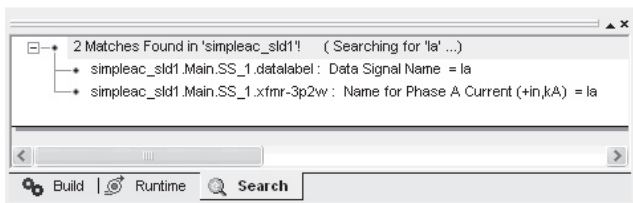
It is possible to narrow a search to just a certain type of object in the project.



Node Numbers is a special type of search, where you can look directly for an electrical node, by node and subsystem number, as opposed to a character string. Some additional selections and fields become enabled if this option is selected: Enter the subsystem number for the node in the **Subsystem #** input field. Enter the node number in the **Node #** input field.

Viewing Search Results

Search results can be viewed in the Search section of the Output window. The Search section will automatically open upon invoking the Find feature. Any instance matching the search criteria will appear in a list form as shown below:



Each entry in the list will have a description including the location of the object plus the object type. Simply left double-click on a particular list entry and PSCAD will automatically point to the source with an arrow icon in Circuit view.

VIEWING BUILD AND DATA FILES

Whenever a project is compiled and built, several files involved in this process are created and stored in the temporary (*.emt) directory associated with the project. Some of these files, such as Fortran, Data and Make files can be useful in debugging projects.

Chapter 11: Debugging and Finalizing

Most of these are simple text files, which can be opened manually by using a simple text editor program. However, some of these files can be viewed directly from within PSCAD.

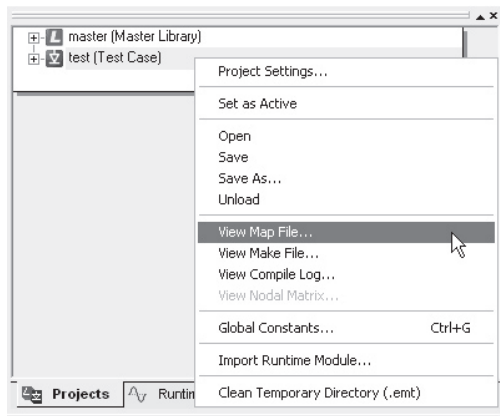
Fortran and Data Files

These files can be viewed by simply clicking the Fortran or Data tabs at the bottom of the Design Editor window (following compilation of course).

These files are module specific, hence their corresponding tabs in the Design Editor. Note that these files are automatically generated by PSCAD. Any editing will be overwritten following the next compilation.

Map and Make Files

Map and Make files involve the entire project, and are therefore not included in the Design Editor tab bar. To view either of these files, right-click on the project title in the Workspace window and select either **View Map File...** or **View Make File....**



This will invoke a separate file viewer window, which will temporarily replace the Design Editor window. This viewer has its own tab bar, which allows you to switch between files quickly.



Note that these files are automatically generated by PSCAD. Any editing will be overwritten following the next compilation. The Log file is a simple file created by the compiler to log the compilation process.

COMPONENT ORDERING

PSCAD V4 now includes a smart algorithm, which will automatically sequence (order) all components in the EMTDC System Dynamics. This is done to ensure that variables are calculated in the proper sequence, and that single time step delays are minimized.

The ordering method in PSCAD V3 was simple: The components were ordered from left to right and from top to bottom of where they appeared in the project page. In PSCAD V4, the algorithm iteratively scans the entire project and then assigns sequence numbers to all existing blocks. Input constants are moved to the top of the sequence, whereas outputs are moved to the bottom.

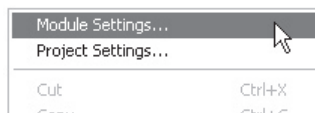
This function should be left on by default at all times, however there may be instances where it is desirable to turn it off and manually order the components during the debugging process. This may be the case if you wish to manually control the feedback points. Alternatively, a feedback can be introduced by inserting a Feedback Loop Selector component in the signal path.

NOTE: Component ordering functions are performed by PSCAD on a per Module basis. By default, the **Assign Execution Sequence Numbers Automatically** option is enabled in all new and existing Modules. This option may be disabled by the user in selected Modules, and yet still maintain automatic ordering in others.

Component Ordering features may be accessed through the Module Settings Dialog. See 'Editing Module Settings' in Chapter 5 for more details.

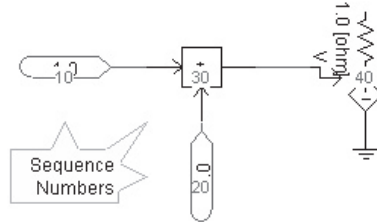
Showing Sequence Numbers

Before manually ordering any components, you must first compile the case and then ensure that the **Show Sequence Numbers on Circuit Schematic** setting is enabled in the Module Settings dialog: To bring-up the Module Settings dialog, right-click on a blank part of the Circuit page and select **Module Settings...**



Chapter 11: Debugging and Finalizing

Once this option is enabled, each component instance in your project should have numbers overlaid on top of their graphic, similar to that shown below:



There are two possible locations within the EMTDC System Dynamics (i.e. DSDYN or DSOUT) where the code for a particular component can reside. As a result, the sequence numbers are colour coded so that the user can graphically determine where code is. These colours are listed below:

Colour Legend:

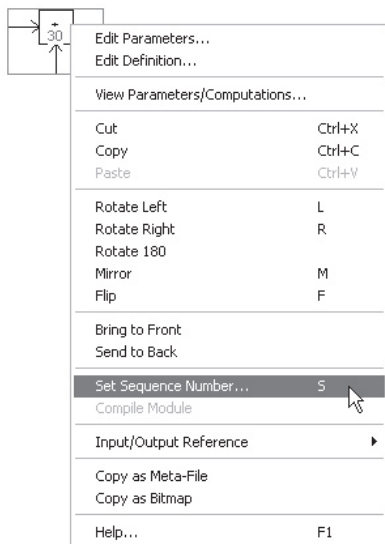
- **Aqua:** Indicates that the component code appears in sequence within DSDYN for that Module.
- **Olive:** Indicates that the component code appears in sequence within DSOUT for that Module.

Manually Setting Sequence Numbers

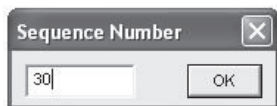
To manually set sequence numbers, first ensure that the **Show Sequence Numbers on Circuit Schematic** option is enabled (as described above), and that the **Assign Execution Sequence Numbers Automatically** option is disabled in the Module Settings dialog.

NOTE: You should first compile the case with **Assign Execution Sequence Numbers Automatically** enabled so as to setup a preliminary code structure before manually adjusting.

Right-click on the component and select **Set Sequence** from the pop-up menu.



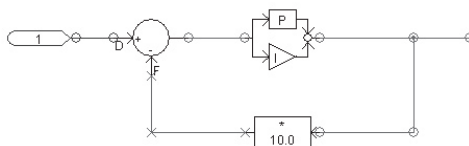
This will bring up the Sequence Number dialog.



Enter the desired sequence number and click the **OK** button. Repeat this process for the remaining components and modules.

SHOW SIGNAL LOCATIONS

Another feature that is helpful when ordering components is the **Show Signal Locations** option in the Module Settings dialog. When this option is enabled, PSCAD will use icons placed on connections and wire termination points to allow for easy, graphical determination of where time step delays are present. In addition, icon colour is used to represent the signal type.



Chapter 11: Debugging and Finalizing

The icons used are listed below with explanations:

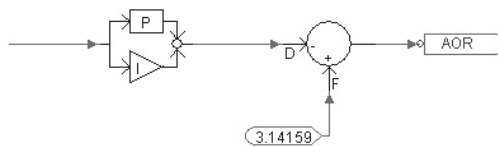
- **○ Feed-Forward Connection:** This symbol indicates that the signal passing through the connection point is classified as a feed-forward signal. This means that the value of the signal is always defined within the present time step.
- **× Feedback Signal:** This symbol indicates that the signal passing through the connection point is classified as a feedback signal. This means that the value of the signal was defined in the previous time step $t - \Delta t$. Due to this fact, feedback signals must be written to, and then extracted from storage each time step.
- **□ Feed Fixed:** This symbol indicates that the signal passing through the connection point is classified as a feed fixed signal. Feed fixed signals are similar to feedback signals, in that their value is always extracted from storage. The difference is that their values are usually defined by an online control, such as a slider or switch.

Colour Legend:

- Green: Indicates a signal of type REAL
- Blue: Indicates a signal of type INTEGER
- Magenta: Indicates a signal of type LOGICAL
- Grey: Indicates an electrical signal

CONTROL SIGNAL FLOW INDICATORS

Control signal flow indicators can be used to help visualize the flow of control signals (i.e. from source to sink) following compilation of the Project. The indicators will appear on Wire components that are part of a control signal path. The signal flow indicators appear as arrowheads directly on the Wire.



To enable this feature, go to **Edit | Workspace Settings...** in the PSCAD Main Toolbar to bring up the Workspace Settings

dialog. Click the **Views** tab and select **Show control signal flow on wires**.

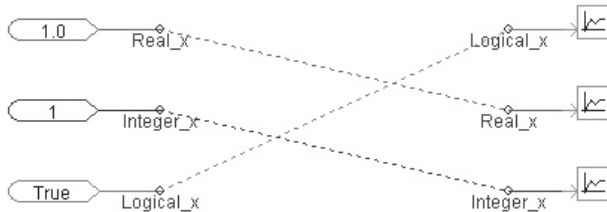
NOTE: The indicators are orientated by default according to the Wire component direction (i.e. towards the End Point of the Wire), not the actual control signal flow. If a flow indicator appears reversed, simply Reverse the Vertices of the Wire. You must compile the case first before you can view the flow indicators.

The indicator colour will appear according to the legend below:

- Green: Indicates a signal of type REAL
- Blue: Indicates a signal of type INTEGER
- Magenta: Indicates a signal of type LOGICAL
- Grey: Indicates an electrical signal

VIRTUAL CONTROL WIRES

Virtual control wires can be used to provide a visual 'virtual connection' between two or more Data Labels of the same name within a specific Module. Virtual control wires appear as a dashed line, which runs directly between corresponding Data Labels as shown below:



To enable this feature, right-click on a blank part of the Module canvas and select **Module Settings...** to bring up the Module Settings dialog. Select the Show Virtual Control Wires option. See 'Editing Module Settings' in Chapter 5 for more details on the other options in this dialog.

NOTE: The dashed connections are purely visual, that is you cannot use them as physical connections for data sources or sinks. You must compile the case first before you can view the control wires.

Chapter 11: Debugging and Finalizing

The colour of the virtual control wire indicates indicates the data type of the Data Labels it is connecting. The colour legend is provided below:

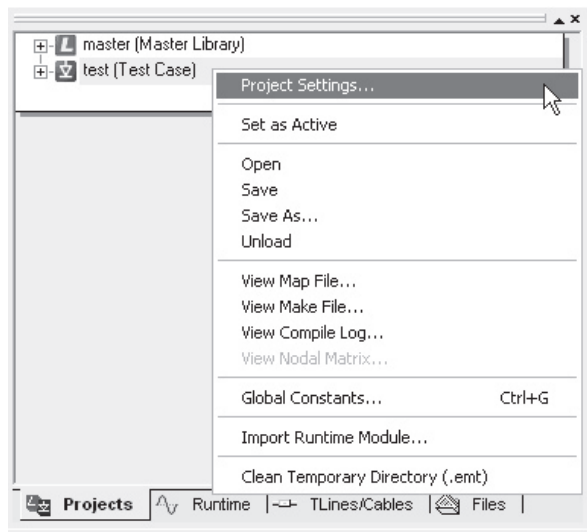
- Green: Indicates a signal of type REAL
- Blue: Indicates a signal of type INTEGER
- Magenta: Indicates a signal of type LOGICAL

USING AN INTEGRATED DEBUGGER

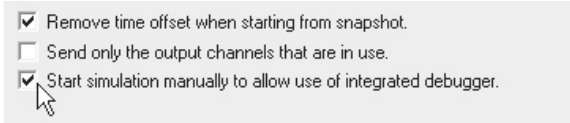
If you are using Digital Fortran 5 or the Compaq Visual Fortran 6 compiler, then it is possible to use either of these programs as an integrated debugger. The following sections describe how to preset relevant Project Settings, as well as step-by-step instructions on how to link EMTDC with your chosen debugging program.

Project Options to Preset

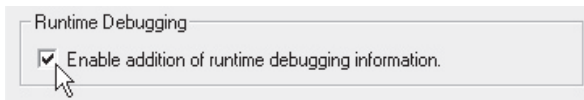
There are a few project settings that must be preset before attempting to use an integrated debugger - all of which are set in the Project Settings dialog. This dialog can be accessed through a right-click on the project title itself (either a case or a library) in the Workspace window, and selecting **Project Settings...** from the pop-up menu.



In the Runtime section of the dialog, enable the **Start simulation manually to allow use of integrated debugger** option. Enabling this option allows EMTDC to be started manually.



Next, navigate to the Fortran section of the Project Settings dialog and enable the **Enable addition of runtime debugging information** option in the Runtime Debugging area.

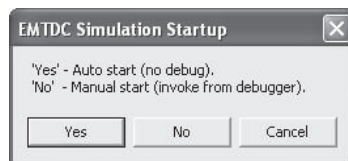


Ensure that any user source files are referenced in the Additional Source (.f) files field, as well. If you fail to do this, your source files will not be included when the project is compiled, and you will not be able to debug your code.

Linking to the Debugger

The following procedure outlines how to set-up an integrated debugging session for your case project. Note that the following steps are described using the Compaq Visual Fortran 6 compiler. If you are using Digital Fortran 5, the steps are very similar.

1. Ensure that the project settings described above in Project Options to Preset are enabled. Run the simulation (press the **Run** button in the Main Toolbar). A pop-up message should appear as follows:



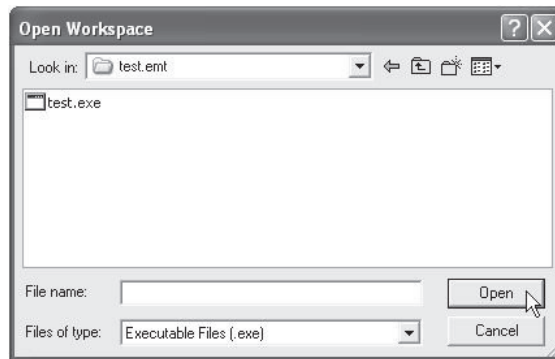
2. Select the **No** button to continue with debugging.
3. On the Status Bar (at the bottom of the PSCAD Work Environment) you should see a command line indicated. If you cannot see the Status Bar, select **View | Status Bar** from the Main Menu. The general format of this command is as follows:

Command: <project_name> -v4 localhost #####

Chapter 11: Debugging and Finalizing

Jot down the four-digit number at the end of the command line so as to not forget it (you will need it soon).

4. Open Compaq Visual Fortran and select **File | Open** from the Main Menu. Change the **Files of Type** drop list to 'Executable Files.' Navigate to the temporary directory (*.emt) associated with your project file (located in the same directory), select the executable file (*.exe) for your project and then click the **Open** button. For example, if your project is entitled 'test.psc,' then you would select 'test.exe' from the 'test.emt' directory as shown below:



5. While still in Compaq Visual Fortran, select **Project | Settings...** to bring up the Project Settings dialog. Click the **Debug** tab and in the **Program Arguments** field, enter the following:

-v4 localhost #####

Note that the four-digit number (####) must be the same as that in Step 3. Click the **OK** button.

6. While still in Compaq Visual Fortran, open the appropriate Fortran source file (*.f) by selecting **File | Open** from the Main Menu. Change the **Files of Type** drop list to 'Fortran Files' and then navigate to the required Fortran file.
7. Insert a breakpoint at the appropriate place in the source code and press the **Go** button.

Please note that it is assumed that the user is familiar with the debugging software and can continue from here. Also, Steps 1 through 7 must be followed for subsequent runs.

Once debugging is complete and the code is clean, make sure that all debugging options outlined in the 'Project Options to Preset' section are disabled. Failure to do so may affect simulation speed.

CREATING LIBRARY (*.LIB) AND OBJECT (*.OBJ) FILES

There are occasions where user-written source code may contain trade secrets, or simply represent a large corporate investment in development time. In instances such as this, it may be in the user's best interest to secure this code before it is distributed, especially when models are to be sold or used by clients or partners in joint ventures. Source code can be easily protected by providing it to clients in a pre-compiled (i.e. binary) format. Any source files linked to a PSCAD project are by default compiled into separate object files, by whichever Fortran compiler is being used. PSCAD also provides a utility to efficiently incorporate multiple object files into a single, compiled library file.

Source files are linked to a Project by one of two methods:

1. Using the File Reference component.
2. The Additional Source (.f) Files field in the Project Setting dialog.

Object (*.obj) Files

Normally, when a source file is linked to a PSCAD project, the Fortran compiler used to compile the project will automatically create a compiled object (*.obj or *.o) file for each linked source file. This file is placed in the project Temporary Directory. The user may choose to supply clients with these compiled object files, rather than the source code. This is fine if only one or two source files are involved. Large projects however, can contain many source file links, and supplying an object file for each source file can quickly become cumbersome. One way to circumvent this problem is to merge all source routines into a single file. This process can also be tedious, and may create problems in the on-going development of the source code.

Library (*.lib) Files

A more efficient means for merging many source files together is to combine all the individual object files into a single compiled library (*.lib) file. PSCAD provides an easy avenue to create a compiled

Chapter 11: Debugging and Finalizing

library file for any Library Project (*.psl) file, provided that links to the source files are set within the Library Project. In Library Projects, this may be accomplished through the use of File Reference components.

Creating a Library (*.lib) File

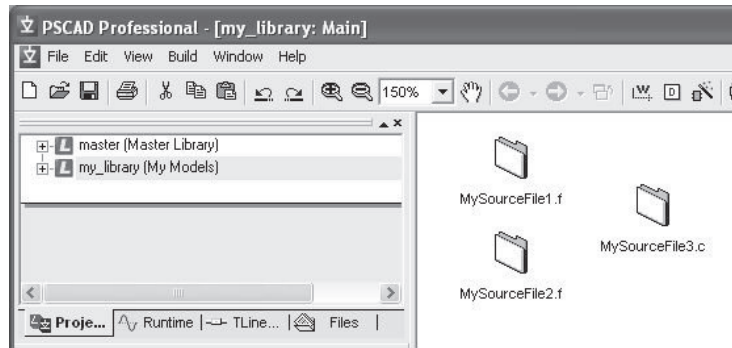
The first thing to consider before creating a library file is the Fortran compiler. Each Fortran compiler will create compiled files, which may or may not be compatible for use with other compilers. In other words, it is important to know what type of Fortran compiler the client is using, so that the files provided are compatible. Most PSCAD users will create an equivalent file for each supported Fortran compiler. These files can then be placed in the appropriate directories as described in Additional Library (*.lib) and Object (*.obj) Files.

Step 1:

Create a new Library Project as described in the section 'Creating a New Project' in Chapter 5, and then link each source file to be included in the compiled library using File Reference components.

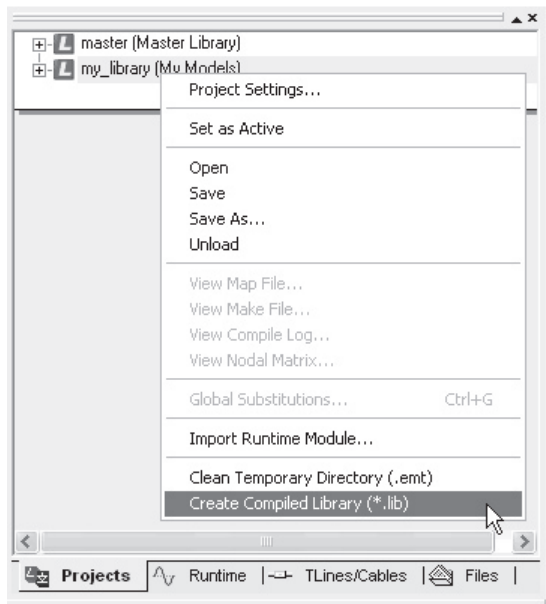
OR

Edit an existing Library Project by ensuring that all source files to be included in the compiled library are linked using File Reference components.



Step 2:

Right-click on the Library name in the Workspace and select **Create Compiled Library (*.lib)**:



When the **Create Compiled Library (*.lib)** function is invoked, PSCAD will create a Temporary Directory for the Library Project located in the same directory as the Library Project (*.psl) file called **<Library Project Name>.emt**. In it will be placed the compiled library file (*.lib), plus an individual object file (*.obj) for each linked source file.

MATLAB®/Simulink®

PSCAD provides users with the ability to interface and utilize the functionality of MATLAB commands and toolboxes (including all graphical commands) through a special interface. This is achieved by calling a special subroutine from within a standard component in PSCAD.

Components that interface to MATLAB/Simulink are not offered as part of the Master Library, and must be specially developed for this purpose. In other words, if a specific MATLAB/Simulink component is required, the user must design his or her own to do the job. Once designed however, this component will be treated as a normal component in PSCAD, and may be used interactively with other components in a particular project.

There are two important things to remember before attempting to interface with MATLAB from PSCAD:

1. The MATLAB Interface is only compatible with the dynamic version of EMTDC. That is, it will not function when using the free EGCS/GNU Fortran 77 Compiler.
2. MATLAB must be installed on your computer in order to use the MATLAB interface.
3. PSCAD can be interfaced with both MATLAB versions 5 and 6 library files.

MATLAB INTERFACE SUBROUTINE

PSCAD interfaces to MATLAB through a single Fortran subroutine called MLAB_INT. This routine is included in the main EMTDC library and may therefore be called from any user-defined component. This routine performs the following functions:

- Launches MATLAB engine using MATLAB Fortran API 'engOpen' commands.
- Changes the working directory to where MATLAB "*.m" files are located.

Chapter 12: MATLAB/Simulink Interface

- Accesses EMTDC variables from the PSCAD STORF and STORI arrays.
- Converts Fortran variables to C-style pointers and allocates/de-allocates memory locations.
- Uses the MATLAB Fortran API to pass the variables/pointers to the MATLAB engine so they can subsequently be accessed from '*.m' files.
- Gets MATLAB output variables using MATLAB Fortran API and places them into the STORF and STORI arrays.

Arguments

```
SUBROUTINE MLAB_INT(MPATH, MFILE, INPUTS, OUTPUTS)
```

Inputs

Argument	Type	Description
MPATH	CHARACTER	Character string of MATLAB '*.m' file path
MFILE	CHARACTER	Name of module within '*.m' file (the .m extension should not be added)
INPUTS	CHARACTER	Format string for all input variables.

Outputs

Argument	Type	Description
OUTPUTS	CHARACTER	Format string for all output variables

The formatting for the INPUTS and OUTPUTS variables should be as follows:

- R for REAL type
- I for INTEGER type
- R(dimension) or I(dimension) for array variables
- Ensure that a space is placed between variables

In PSCAD V4, INPUTS and OUTPUTS variables can be empty strings, in which case the '*.m' file will run without arguments. This

PSCAD

is helpful in initializing the MATLAB environment and designing a component that runs MATLAB *.m files and Simulink *.mdl files simultaneously.

EXAMPLE 12-1:

A MATLAB module is called by [D] = TEST(A,B,C), where TEST is a module in a MATLAB file 'TEST.m,' that is located in C:\TEMP\MLAB_FILES. The input 'A' is a REAL variable, 'B' is a REAL array of dimension 31 and 'C' is an INTEGER. The output 'D' is a REAL array of dimension 10.

The MATLAB interface subroutine call would then appear as follows:

```
CALL MLAB_INT("C:\TEMP\MLAB_FILES", "TEST", "R R(31)
I", "R(10)")
```

EXAMPLE 12-2:

A MATLAB file entitled 'TEST.m' is located in C:\TEMP\MLAB_FILES. It consists of MATLAB commands that may take a snapshot of MATLAB results, or initialize the environment (such as setting global variables or changing directory, etc.).

The MATLAB interface subroutine call would then appear as follows:

```
CALL MLAB_INT("C:\TEMP\MLAB_FILES", "TEST", "", "")
```

SIMULINK INTERFACE SUBROUTINE

PSCAD interfaces to Simulink through a single Fortran subroutine called 'SIMULINK_INT.' This routine is included in the main EMTDC library and may therefore be called from any user-defined component. This routine performs the following functions:

Chapter 12: MATLAB/Simulink Interface

- Launches MATLAB through Fortran API functions the same as the MATLAB Interface Subroutine.
- Changes the working directory to where the Simulink '*.mdl' files are located.
- Accesses EMTDC variables from the PSCAD STORF and STORI arrays.
- Uses the MATLAB Fortran API to pass the variables/pointers to the MATLAB engine so they can subsequently be accessed from '*.mdl' files.
- Sets the simulation data, specified by the Workspace I/O pane of the Simulation Parameters dialog box, and runs the Simulink module using the MATLAB command 'set_param.'
- Synchronizes Simulink to PSCAD. That is, PSCAD proceeds only after the Simulink module simulation is completed each time step, which ensures that the correct results from Simulink are passed to PSCAD.
- Gets Simulink output variables and places them into the EMTDC STORF arrays. This is done in two steps, first the MATLAB command 'get_param' is employed to get the variable name of Simulink outputs from the Workspace I/O pane in the Simulation Parameters dialog box, then MATLAB Fortran API is utilized to exact and place them into EMTDC STORF arrays.

Arguments

```
SUBROUTINE SIMULINK_INT(MPATH, MFILE INPUTS)
```

Inputs

Argument	Type	Description
MPATH	CHARACTER	Character string of MATLAB '*.mdl' file path
MFILE	CHARACTER	Name of module within '*.mdl' file (the *.mdl extension should not be added)
INPUTS	CHARACTER	Format string for all input variables.

The formatting for the INPUTS variable is the same as that in the MATLAB interface subroutine. Note that the OUTPUT of the

PSCAD

SIMULINK interface is automatically handled inside the subroutine and is always put into the corresponding EMTDC STORF array.

EXAMPLE 12-3:

A Simulink module called 'TEST.mdl,' has external inputs A, B and C and is located in C:\TEMP\SIMULINK_FILES. The input 'A' is a REAL variable, 'B' is a REAL array of dimension 31 and 'C' is an INTEGER.

The Simulink interface subroutine call would then appear as follows:

```
CALL SIMULINK_INT("C:\TEMP\SIMULINK_FILES", "TEST", "R  
R(31) I")
```

DESIGNING A MATLAB COMPONENT

Designing a MATLAB component involves two simple steps:

1. Create a new component
2. Write a MATLAB file (.m) to perform the required modeling.

Component Design

Any number of signals or parameters can be passed to or from a MATLAB component. The Fortran code inserted into the Fortran segment of the component definition should perform four tasks:

1. Input variables for the MATLAB function should be transferred to STORF and/or STORI arrays.
2. The MLAB_INT subroutine must be called with arguments for the MATLAB module and path name, input format string and output format string (more information to follow).
3. Output variables should be transferred from STORF and/or STORI arrays into the PSCAD component output connection nodes.
4. Increment the NSTORF and/or NSTORI index pointers by the total number of variables used.

Chapter 12: MATLAB/Simulink Interface

EXAMPLE 12-4:

Consider a simple example for a PSCAD component, which has 2 REAL input connection nodes (A and B), and a single REAL output connection node (C).



The following code should then appear in the Fortran segment of the component definition:

```
#STORAGE REAL:3
    STORF(NSTORF) = $A
    STORF(NSTORF+1) = $B
!
    CALL MLAB _INT("$Path", "$Name", "R R", "R")
!
    $C = STORF(NSTORF+2)
    NSTORF = NSTORF + 3
!
```

The component definition will need to define at least two input fields in a Parameters section category page. In this case for example, **\$Path** is a text field symbol name expecting the pathname to where the *.m files are located. **\$Name** is also a text field symbol name expecting the name of the MATLAB module. For example, if the MATLAB function is called 'TEST1,' contained within a file called 'TEST1.m,' then the **\$Name** parameter should then be entered as 'TEST1.'

Matlab Module Name	TEST1
Path of .m Files	C:\matlab\mfiles

More complex input and output arguments can also be used. If an array signal is used, the input or output format string arguments should contain the type and dimension. For example, a REAL array

PSCAD

of dimension 31 would appear as R(31), or an INTEGER array of dimension 10 as I(10). Each variable should be separated by one or more spaces, and the order of variables should be identical to the order expected in the MATLAB function.

A good mechanism in Fortran to transfer array variables into or out of the EMTDC STORF or STORI arrays is the DO/ENDDO loop.

EXAMPLE 12-5:

Here is an example illustrating a MATLAB component, which has a single REAL input connection array of dimension 31, and a single output connection of dimension 2.

The following code should then appear in the Fortran segment of the component definition:

```
#STORAGE REAL:33
#LOCAL INTEGER I _CNT
!
! First Input Array (REAL(31))
!
      DO I _CNT = 1,31,1
          STORF(NSTORF+I _CNT-1) = $INPUT(I _CNT)
      ENDDO
!
      CALL MLAB _INT("$Path","$Name","R(31)","R(2)")
!
! First Output Array (REAL(2))
!
      DO I _CNT=1,2,1
          $OUTPUT(I _CNT) = STORF(NSTORF+31+I _CNT-1)
      ENDDO
!
! Increment STORF pointer
!
      NSTORF = NSTORF + 33
!
```

Note that when the output variables are extracted from the STORF array, the offset 31 must be added, as 31 input variables were al-

Chapter 12: MATLAB/Simulink Interface

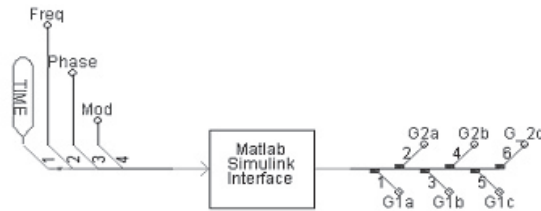
ready used to put the INPUT variables into the STORF array. In the entire routine, 33 STORF locations were used (31 inputs, 2 outputs).

DESIGNING A MATLAB/SIMULINK COMPONENT

The design steps and principles used in creating a MATLAB/Simulink component, are similar to that explained in the preceding section. Here, an example is provided to illustrate how to design a component that utilizes both the MATLAB and the Simulink interface.

EXAMPLE 12-6:

The figure below illustrates a MATLAB/Simulink component in PSCAD. The input to the component is an array of four variables, named 'TIME', 'Freq', 'Phase', and 'Mag'. The output from the component is an array of six variables as labelled.



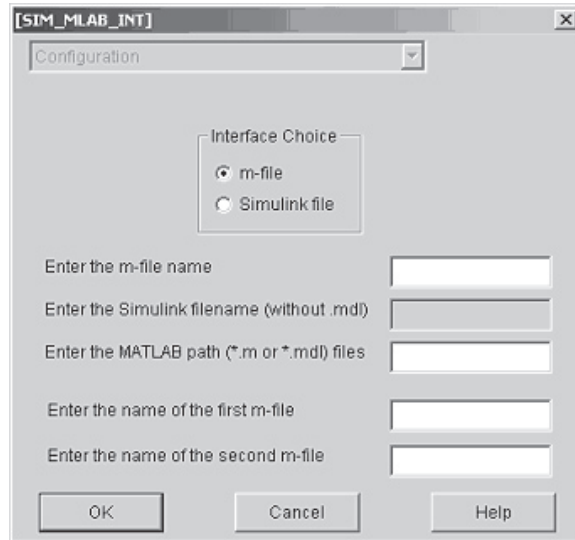
The following code appears in the Fortran segment of the component definition:

```
#STORAGE REAL:12
#LOCAL INTEGER I_CNT, M_CNT
!
! PSCAD MATLAB INTERFACE
! MODULE: Matlab Interface with Simulink
!
      I_CNT = 1
      M_CNT = 0
!
! Call *.m files in order to initialize the
! environment...
!
      IF (TIMEZERO) THEN
#IF "$mName1" != ""
      CALL MLAB_INT("$Path", "$mName1", "", "")
#ENDIF

```

```
#IF "$mName2" != ""
    CALL MLAB_INT("$Path","$mName2",""," , ")
#ENDIF
    ENDIF
!
!   TIME info to run either *.m or *.mdl module
!
#IF ($OPTSEC == 0 )
    STORF(NSTORF) = TIME
    STORF(NSTORF + 1) = DELT
    M_CNT = 2
#ELSE
    STORF(NSTORF) = TIME
    M_CNT = 1
#ENDIF
#ENDIF
!
!   Transfer inputs to EMTDC STORF array
!
    DO I_CNT = 1, $#DIM(sig_in)
        STORF(NSTORF + M_CNT + I_CNT-1) = $sig_in(I_CNT)
    END DO
    M_CNT = M_CNT + I_CNT - 1
!
!   CALL PSCAD MATLAB INTERFACE
!
#IF $OPTSEC == 0
    CALL MLAB_INT("$Path","$mfile","R(6)","R($#DIM(sig_
out))")
#ELSE
    CALL SIMULINK_INT("$Path","$simfile","R(5)")
#ENDIF
!
!   Transfer MATLAB output variables
!
    I_CNT = 1
    DO WHILE (I_CNT .LE. $#DIM(sig_out))
        $sig_out(I_CNT) = STORF(NSTORF + M_CNT + I_CNT
- 1)
        I_CNT = I_CNT + 1
    END DO
!
!   Update storage array
!
    NSTORF = NSTORF + M_CNT + I_CNT - 1
```

Various Parameters category pages within the component definition define the variables used in the above code. This category page would appear similar to that shown below:



Each of the input fields above defines a variable (preceded by a \$ symbol) in the component code.

INTERFACING NOTES

The MATLAB engine performs operations very slowly, compared with the same equivalent operation hard-coded directly into a PSCAD component. The MATLAB source code is interpreted each time it is called, allowing users to dynamically edit the “*.m” file in the middle of a run and see its effect immediately. This inter-activity is also possible in PSCAD through the use of on-line sliders, switches, dials and buttons. Any combination of the two methods can be used simultaneously.

Alternative Simulink Interface

There is an alternative way to invoke a Simulink module. Instead of calling the ‘SIMULINK_INT’ subroutine, users may call ‘MLAB_INT’ subroutine to invoke an “*.m” file, which uses the MATLAB command ‘sim’ to handle the Simulink module. However, the use of ‘SIMULINK_INT’ is highly recommended due to the synchronization mechanism between MATLAB and PSCAD implemented within this subroutine. This is especially true for Simulink modules that run longer than the time step defined inside EMTDC.

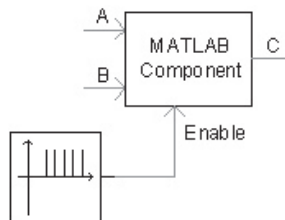
PSCAD

Simulation Speed

To try and speed up the MATLAB solution, it is often a good idea to try and use a larger time step when invoking MATLAB components (wherever possible or practical). An enable/disable switch can also be implemented, so as to allow PSCAD to operate at close to full speed.

EXAMPLE 12-7:

The figure below illustrates one way to speed up your PSCAD/MATLAB simulation. Here, an impulse train is applied to an additional enable/disable input to control how often the MATLAB solution engine is invoked.



The Impulse Train frequency can be varied for optimal speed/accuracy considerations.

Conversion to C

It should also be possible to convert '*.m' source code directly to 'C' code using the MATLAB 'C' compiler, and then directly compile and link the 'C' source code into the EMTDC executable. This has not been tested thoroughly, but no obvious technical reasons exist as to why it should not. With this 'hard-compiled' approach, you lose the ability to edit the MATLAB '*.m' interpreted file during simulations. The 'hard-compiled' approach also may not work with any MATLAB graphical functions.

A good compromise can be reached by first using MATLAB components to develop and test algorithms, but in the end to optimize the speed of the final design by hard-coding the algorithms in Fortran or C (either manually or using the MATLAB 'C' compiler). The final hard-

Chapter 12: MATLAB/Simulink Interface

coded algorithms are linked directly with the EMTDC solution engine and are very fast (as they can be optimized using modern compilers).

Plotting Enhancements

The MATLAB graphics functions are a very powerful addition to the PSCAD plots and graphical interface. Three-dimensional plots, active graphics and rotating images are possible and integrate seamlessly with the PSCAD graphical libraries.

ENABLING AND USING THE INTERFACE

In order to make use of the MATLAB/Simulink interface, you must first enable it in the project you are working with. This is accomplished by selecting the option **Use MATLAB Interface** in the Fortran section of the Project Settings dialog.

Also, you must ensure that your MATLAB/Simulink Workspace options are properly set. These are adjusted in the Fortran section of the Workspace dialog.

Note that you may find discontinuities when starting from a snapshot file if you use complex models in MATLAB/Simulink. If this is the case, it can be prevented by adding extra code in the *.m or *.mdl file and the component definition calling the interface.

Migrating from Older Versions

During the development of PSCAD V4, compatibility with older versions (especially with V3) was a primary concern. As a result, migrating PSCAD V3 projects into V4 is quite simple and straightforward. Although not much has changed with the PSCAD V2 migration process, an important new feature is that V2 users may now load transfer files (*.dfx) directly into PSCAD V4. This eliminates the need for extra steps in migrating user components and libraries - these definitions can simply be included in the V2 transfer file.

This chapter discusses topics in migrating PSCAD related projects from older versions. For more information on migrating user-defined code and other EMTDC related issues, see *Converting V2 Fortran Files in EMTDC Manual* chapter entitled *V2 Conversion Issues*.

CONVERTING PSCAD V2 PROJECTS

PSCAD V2 projects can be directly migrated into PSCAD V4 without too much difficulty. This, of course, mainly depends on the complexity of the V2 project, the components used and the condition of any user-written code involved. Unlike V2, PSCAD V4 stores all information pertaining to a project in a single portable project file (either a case project (*.psc) or a library project (*.psl) file).

Some V2 files cannot be directly imported into PSCAD V4. The following is a list of which file types can and cannot be migrated:

Can be migrated:

- Draft circuit files (*.dft and *.dfx)
- Runtime batch files (*.rtb)
- Draft component definitions (for example in `xdraft_lib`)
- EMTDC Fortran source code
- Line constants solved data (*.tlb or `tlines` files)

NOTE: If you have EMTDC source code written in C, then you must upgrade this code manually.

Cannot be migrated:

- Cable files (*.cbl)
- Multiplot or Uniplot batch files

The following topics describe the procedures involved when importing PSCAD V2 circuits, components and associated files into PSCAD V4.

User-Written EMTDC Source Code

Fortran source code written specifically for use with EMTDC in PSCAD V2 must be filtered before it can be used in PSCAD V4. For more information on importing user-written Fortran source code, see *Converting V2 Fortran Files*.

Conversion Issues

The following sections describe a general overview and background of some important conversion issues. Please review these sections first before attempting to upgrade to PSCAD V4.

V2 Control Type Components

All control type components (i.e. components whose code appears in the EMTDC System Dynamics only) should be upgradable. One important fact to consider, however, is that using the EMTDC internal variable for time step (i.e. DELT) in the Computations portion of a component definition is no longer allowed in PSCAD V4.

V2 Electrical Interface Components

There have been many changes made to EMTDC since V2, mostly associated with electrical signals and their interface to user-written components. It is possible that these types of components may cause some compatibility problems when upgrading to V4. For instance, electrical branches in EMTDC were originally referenced using a branch TO and FROM node convention. This tended to cause problems with parallel branches (as they have identical connection nodes), and resulted in numerous workarounds to avoid the problem. One example was the output of the calculated current in faults, breakers, thyristors, diodes, GTOs, arresters, etc. - all of these had a single time step delay. This was because all parallel switching branches were combined into a single branch for solution in the main program. One side effect was that the current had to be output as an argument of the DSDYN subroutine call, and could not be placed in DSOUT.

PSCAD

In PSCAD V4, the above problem is circumvented, as each branch is given a unique branch number and the current in each branch is referenced directly in DSOUT. This system of referencing electrical branches directly (by a branch number instead of by node numbers) results in some obsolete function calls. These include all switching routines, as well as any electrical interface array names.

V2 Component Libraries

Since V2, PSCAD has allowed users to create their own components to represent custom models. In V2, this was accomplished by editing a text file, where graphical information, parameter form data, and Data/Fortran output code were entered. Each component was contained within its own file, which resided in either the user's library (~/PSCAD/xdraft_lib), or in a Group Library (only two libraries were allowed).

Those who received a project from other users always experienced difficulties in maintaining these components and great confusion (as to what the most recent version of the component was) generally resulted. If two or more components by the same name were kept, then the PSCAD V2 Draft program would search first in the user's xdraft_lib directory, then in the Group directory (if one has been specified), and then finally in the Master library. This allowed users to overwrite the functionality of Master Library components, often with confusing or inconsistent results.

In PSCAD V4, the Design Editor is used to graphically manipulate custom components. Component definitions are stored within a single library project file (*.psl), where any number of library projects can be loaded in the Workspace simultaneously. Users who wish to transfer custom components to other users need only to supply the library project containing the components.

When PSCAD V4 loads a project, it knows not only the name of the component, but that it came from a library project. This allows users to copy a component definition from any other library, customize it, and then place it in their own library. Whenever either of these components is used, PSCAD will keep track of which library each is from. PSCAD V4 also allows components to be kept directly in case projects (*.psc), so that a temporary component can be developed for a specific case, without having to clog up a library with the test code.

Chapter 13: Migrating from Older Versions

Component Definitions

If a required component definition cannot be found when a V2 draft project is loaded into PSCAD V4 (i.e. the definition is not included in any loaded V4 library project or the specific V2 draft project), the draft project will still load successfully. However, any component instances based on the missing definition will be displayed with a 'placeholder' component. To rectify this situation, ensure that a V4 library project already containing the required component definition is loaded before the draft project, or that draft project itself was saved in V2 as a transfer file (*.dfx).

The PSCAD V4 Master library is automatically loaded when PSCAD V4 is started. It contains many new components, in addition to all PSCAD V2 components from the following libraries:

- PSCAD V2 Master library
- PSCAD V2 PEMISC library
- PSCAD V2 Machines library

Note that it is important to always use unique names for user-defined component definitions. This will avoid unwanted interactions between user and Master Library component definitions.

Importing V2 Draft and Runtime Batch Files

Most PSCAD V2 draft projects can be directly imported into PSCAD V4. Projects containing more complex circuits however, may require some manual adjustments. This section describes procedures for importing PSCAD V2 draft files (*.dft or .dfx) and runtime batch files (*.rtb) into PSCAD V4.

Before attempting to import your PSCAD V2 projects, please ensure the following:

- All obsolete PSCAD V2 components are replaced with the latest versions from the most current V2 Master Library.
- All PSCAD V2 files associated with the draft project itself (i.e. *.rtb, *.tlb, and tline_out) must be located in the same directory as the draft file before the draft file is imported into PSCAD V4.

NOTE: PSCAD V4 will initially allow the import of only a single runtime batch (*.rtb) file, associated with a specific V2 draft (*.dft or *.dfx) file. This *.rtb file must have the identical filename as the draft file being imported. For more information on importing any remaining Runtime batch files, see Importing Additional Runtime Batch Files below.

- If the draft project contains custom written components (i.e. from user or group libraries), then ensure that the draft file has been saved as transfer (i.e. *.dfx file) in PSCAD V2. Otherwise, see Importing V2 User Libraries or Importing Individual V2 Components in this chapter for more details.

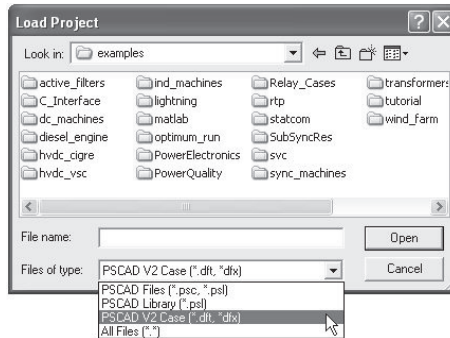
NOTE: If a V2 draft project containing undefined components is loaded into PSCAD V4, all undefined components will be substituted with a temporary 'placeholder' component.

Please follow these steps to avoid problems:

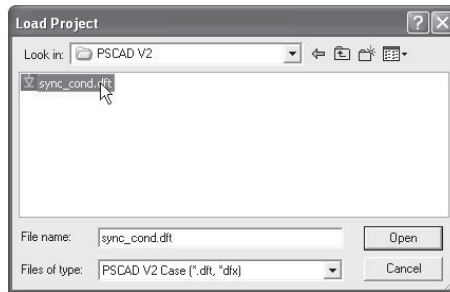
1. To load a V2 draft file, start PSCAD V4 and select **File | Load Project...** from the Main Menu bar. The Load Project dialog window should appear.
2. Near the bottom of the Load Project dialog, change the **Files of Type** drop list to **PSCAD V2 Case (*.dft, *.dfx)**, so as to view PSCAD draft files only. If you do not have a direct network connection from your PC to your UNIX system, then you must first copy (or ftp) the files to your PC from UNIX.

NOTE: Your V2 files may still be in UNIX format. In order to successfully import the file to PSCAD V4, you must ensure that all files are in DOS format. This can be accomplished by using the 'unix2dos' UNIX utility, or FTP in ASCII mode.

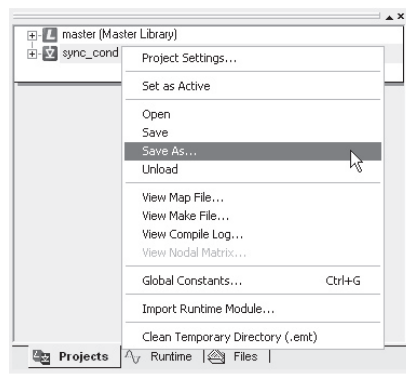
Chapter 13: Migrating from Older Versions



3. Navigate to the directory containing your V2 project files, select the V2 draft file to be loaded, and then press the **Open** button.



4. Check for warning or error messages in the Output window. See Common Warning Messages below for details. Rename and save the new PSCAD V4 case project in V4 format (right-click on the project name in the Workspace window and select Save As...).



PSCAD

Common Warning and Error Messages

There are some common warnings and error messages that may appear in the Output window following the migration of a V2 draft file. Some of these are described below:

Warnings:

- Unresolved keyword 'xxxxx': This warning indicates that data in a PSCAD V2 component was stored, but is not required in the PSCAD V4 component. This often occurs in components, which can be used in both EMTDC and RTDS, where some of the data is not required for EMTDC runs.
- Component aliased from 'xxx' to 'yyy': This warning indicates that a component has been renamed in PSCAD V4, but the data from the old V2 component has been converted.

Errors:

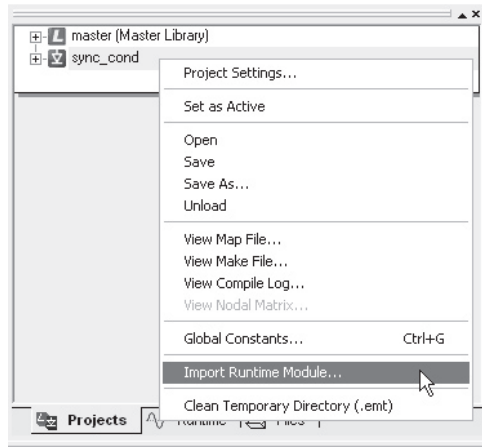
- Component Definition 'xxx' not resolved: This means that a custom component was used in PSCAD V2, but was not available in any of the library projects loaded in the PSCAD V4 Workspace. If you have previously converted your PSCAD V2 libraries into PSCAD V4 library projects, then simply ensure that the libraries are loaded in PSCAD V4 first before converting the PSCAD V2 draft project. If you have not yet converted PSCAD V2 libraries into PSCAD V4 library projects, then follow the procedure outlined in Importing V2 User Libraries. You can also go back to PSCAD V2 and save your draft project as transfer (i.e. *.dfx file), so as to incorporate any custom components into it before importing to V4.

Importing Additional Runtime Batch Files

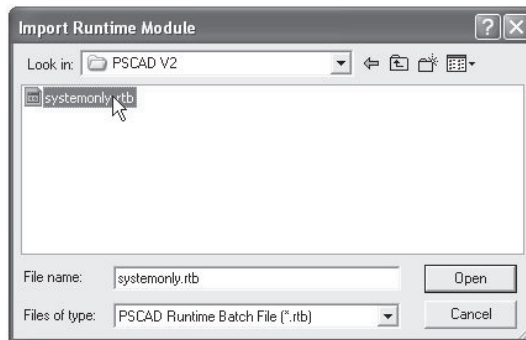
As mentioned above, only a single runtime batch (*.rtb) file will be initially imported along with a PSCAD V2 draft (*.dft or *.dfx) file. If more than one runtime batch file exists in a specific V2 draft project, then the remaining *.rtb files can be imported separately. This is accomplished as follows:

Once steps 1 to 4 above have been completed (i.e. the draft file has been successfully imported into PSCAD V4), right-click on the case filename in the Workspace and select **Import Runtime Module...**

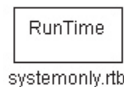
Chapter 13: Migrating from Older Versions



This will bring up the **Import Runtime Module** dialog window. Select the *.rtb file you wish to import and click the **Open** button.



A new Module containing the contents of the Runtime batch file should appear near the top-left corner of your case project main page, similar to that shown below. Simply left double-click the Module to enter.



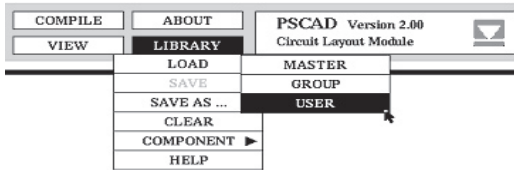
Importing V2 User Libraries

The PSCAD V2 Draft program used a library (*.lib) file as a palette (appears on the right-hand side of the V2 Draft program canvas) for the actual component library.

PSCAD

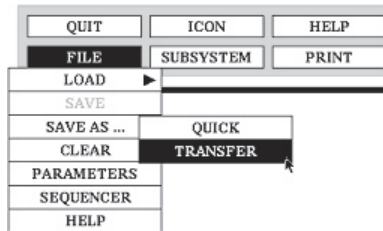
To transfer this palette into PSCAD V4:

1. Open PSCAD V2 on your UNIX terminal and run the Draft program. Press the **LIBRARY** button in the main menu and select **LOAD | USER**.



NOTE: Make sure that any of the components included in your PSCAD User Library file have a unique component definition name. Otherwise these components may be overwritten by other component definitions.

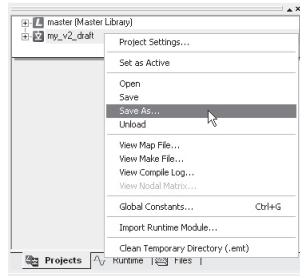
2. Select the desired user library file from the Load User Library dialog and then press the **PROCEED** button.
3. Copy the desired component instances from the user library palette over to the Draft canvas. From the **FILE** menu, select **SAVE AS... | TRANSFER** to save the Draft project as a transfer file (i.e. as a *.dfx file). This will ensure that the component definitions are included within the V2 draft file.



4. Transfer this V2 project file to a location where it can be accessed from your Windows PC. If you do not have a direct network connection from your PC to your UNIX system, then you must copy (or ftp) the files to your PC from UNIX.

Chapter 13: Migrating from Older Versions

5. Follow Steps 1 to 4 in Importing V2 Draft and Runtime Batch Files to load the V2 draft project into PSCAD V4.
6. In the PSCAD V4 Workspace, right-click on the project filename and select **Save As...** from the pop-up menu.



7. Rename the case project to a library project in the **File Name** field of the Save Project As dialog window. Press the **Save** button.

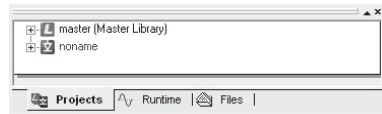
You now have converted your V2 user library to a PSCAD V4 library project!

Importing Individual V2 Components

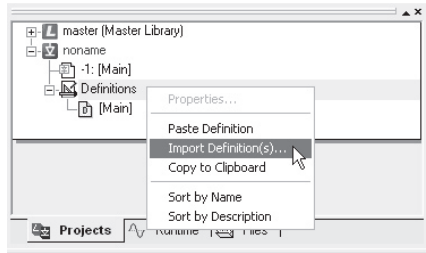
It is possible to import individual PSCAD V2 component definition files, directly into a PSCAD V4 library project. Once this is accomplished, an instance of the definition may be created and displayed within the library.

Please follow these steps to avoid problems:

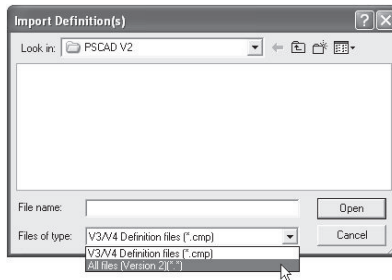
1. Create a new library project in PSCAD V4 (or open an existing library). To create a new library, select **File | New | Library**. A new library called 'noname' should appear in the Workspace window.



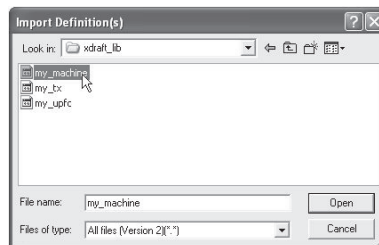
2. Expand the new library project tree by clicking on the '+' symbol beside its name. Right-click on the Definitions branch, and select **Import Definition(s)...**



3. In the Import Definition(s) dialog window, change the **Files of Type** drop list to 'All Files.'



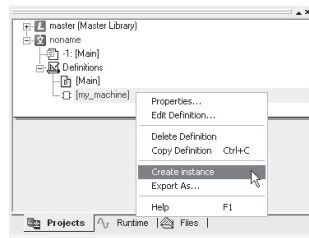
4. Navigate to the directory where you have stored your PSCAD V2 component definition files ('~/PSCAD/xdraft_lib' for example). If you do not have a direct network connection from your PC to your UNIX system, then you must first copy (or ftp) the files to your PC from UNIX.



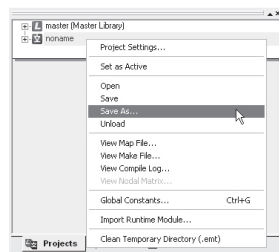
NOTE: Do not select any PSCAD V2 macro files (*.g), library files (*.lib), or any file beginning with a 'period.'

Chapter 13: Migrating from Older Versions

5. Select one or more files, and then click on the **Open** button. The new definition(s) should appear within the Definitions branch in the Workspace window.
6. Check for warning or error messages in the Output window. If you receive an error message similar to “Macro file ‘xxx.g’ not found,” you need to move all associated V2 macro files (*.g) from the ‘~/PSCAD/script’ directory on your UNIX system to the same directory from where you are importing your V2 definition files. Unload the new library and start over from Step #1.
7. Expand the Definitions branch in the Workspace window by clicking on the ‘+’ symbol beside its name. Select a definition, right-click and select **Create Instance** from the pop-up menu.



8. Open the library project main page in Circuit view. Right-click on the page and select **Paste** from the pop-up menu. A new graphical instance of the component should appear. You can also use Drag and Drop to perform steps 7 and 8.
9. Right click on the ‘noname’ library in the Workspace window and select **Save As...** to rename and save the library project.



Manual Revisions to the New PSCAD V4 Project

Open the new PSCAD V4 project in Circuit view (left double-click on the project name in the Workspace window) and you should see a parent (top-level) module that did not exist in your PSCAD V2 project. This module will contain other modules representing each page that existed in your original V2 draft project. A separate module will also be included, which contains all information from the associated runtime batch (*.rtb) file.

Depending on whether or not there were any associated import or export connections in the original V2 draft project, each module may now possess external input and output connections on their respective graphics. Each of these connections corresponds to an import or export connection in PSCAD V2. PSCAD V4 uses this input/output information to order modules in such a way as to minimize any feedback paths in signals.

Too Many External Connections

If the parent module is visually unreadable due to too many external connection graphics on the module, then this is probably because the original V2 Draft page had an excessive number of import or exports (i.e. > 100). The auto-routing of these connections in PSCAD V4 becomes too complex in such situations, but the connections are still functionally correct. To simplify matters, you can go back to the PSCAD V2 Draft circuit, and manually collapse related import and export signals into arrays. Then go through the import procedure again. This will reduce the number of connections in the parent module and help to keep things organized.

Import/Export Tags with Array Variables

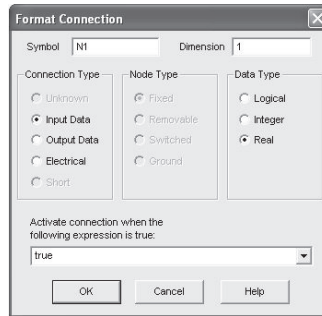
Each module may also require editing if you used any import or export tags with array variables in the original PSCAD V2 draft project. The old import and export components did not contain any information regarding the dimension or type of variable (all import/exports were performed with REAL numbers).

PSCAD V4 now allows import/exports to be performed with the original variable type, thus avoiding needless (and potentially incorrect) data conversions. The user must manually enter the import/export variable type and dimension information into the modules (there will be at least two modules that must be edited - one for the import and one for the export).

Chapter 13: Migrating from Older Versions

If you already know which signals are affected (i.e. imported or exported signals using arrays, or of a type other than REAL), then you must edit the definition of the module in PSCAD V4 to identify the signal types.

1. Open the definition of the module: Right-click over the module and select **Edit Definition....**
2. In Graphic view (appears by default), double-click on the Connection, having the incorrect type or dimension, to bring up the Format Connection dialog.



3. Modify the **Data Type** and **Dimension**. Press the **OK** button to save changes and then go back to Circuit view. Note that the **Connection Type** will be 'Input Data' if the signal in the original PSCAD V2 project was being imported, and will be 'Output Data' if the original signal was being exported.
4. Repeat steps 1- 3 for any other existing connections for this signal.

To find out if there are any more connection errors for a particular module, compile the module manually (i.e. right-click on the module and select **Compile Module**). This will force PSCAD to generate the data and Fortran files for this module alone, and any error or warning messages should appear in the Output window.

Examples of such errors are:

- **Signal 'ABC' dimension mismatch at signal 'XYZ':** This indicates a dimension mismatch between two connected signals.

- **Signal 'DEF' type mismatch at signal 'UVW':** This indicates a data type mismatch between two connected signals.

Runtime Module

The Runtime module represents the contents of the V2 runtime batch file. This module may require some 'house cleaning' and re-organization (i.e. re-size graph frames, etc.). Control panels are used to contain sliders, dials, switches and meters. Grouped components in the PSCAD V2 Runtime program are now added merely as additional modules.

Migrating V2 Cable Systems

PSCAD V2 cable migration into V4 is not supported. Any cable systems in your V2 project (i.e. defined in *.clb files) must be re-constructed from scratch once the project is migrated into PSCAD V4. See the chapter entitled Transmission Lines and Cables in this manual for more details.

Migrating V2 Transmission Line Systems

There are fundamental differences in how transmission line systems are represented between PSCAD Versions 2 and 4. In V4, transmission line systems are interfaced to the rest of the electric circuit through special interface components. The properties of the actual transmission corridor are defined within a special properties component (i.e. tower geometry, conductor properties, etc.).

When a PSCAD V2 draft project containing transmission line systems is migrated into V4, the above described transmission system is not automatically constructed. Instead, the transmission line system is inserted into the V4 project using special V2 alias components called V2 Style T-Line Connection. The transmission system properties remain based on the transmission line batch (*.tlb) file created by V2.

Although this substitution will provide correct results when the simulation is run, it may prove cumbersome if the transmission system properties need to be changed (must edit the *.tlb file), or if the project file is transferred to other users (must move the *.tlb file with it). It is therefore recommended that V2 style transmission systems be eventually converted to V4 format.

Although this process must be performed manually, it is straightforward. Simply replace the V2 Style T-Line Connection

components with an equivalent V4 transmission line interface component. The transmission line properties can then be read directly from the V2 transmission line batch file, and inserted into a properly constructed transmission line corridor. See the chapter entitled Transmission Lines and Cables in this manual for more details.

CONVERTING PSCAD V3 PROJECTS

Loading PSCAD V3 projects into PSCAD V4 is fairly straightforward, and for the most part will be accomplished without incident. Simply load the V3 case or library project exactly as you would a V4 project.

Conversion Issues

As is normal when software is updated, there are changes made to existing features, which may or may not affect your case. There are a few important issues to be aware of before starting to use your V3 projects in PSCAD V4. These are described in the following sections.

System Dynamics Component Ordering

Components used in the EMTDC System Dynamics (i.e. CSMF components, modules, etc.) are automatically ordered in V4 with a sophisticated new sequencing algorithm. In PSCAD V3, these types of components were ordered using a simpler method.

This important feature should be considered when initially verifying your V3 project results in PSCAD V4. The new sequencing algorithm may adjust the order in which components appear in the Fortran code. Depending on where the components within the EMTDC System Dynamics (i.e. DSDYN or DSOUT) are, a single time step delay may be added or removed in comparison with V3 results.

If this does indeed occur in your case, PSCAD V4 allows you to manually adjust the component sequence. Please see the Component Ordering section in the chapter entitled Debugging and Finalizing for more details.

Additional Libraries and Object Files

In PSCAD V3, additional library and object file paths, entered in the **Additional .obj and .lib Files** field in the Project Settings dialog, were relative to the respective temporary (*.emt) directory for the project. In PSCAD V4, these paths are now relative to the **User**

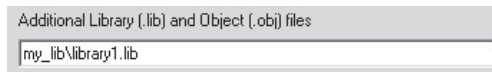
PSCAD

Library Path (set in the Workspace Settings dialog window), or can be entered directly as absolute paths.

These paths must be changed accordingly in your V4 projects. Open the respective Project Settings dialog for the project (right-click on the project filename in the Workspace and select **Project Settings...**) and select the Fortran tab. Modify the paths in the Additional Fortran Library (*.lib) and Object (*.obj) Files input field. For example, a V3 reference is shown below:



would appear as follows in PSCAD V4:



Flyby Windows

A new feature was added to PSCAD V4, which optimizes variable storage between time steps during a simulation. Unfortunately, when this optimization algorithm is enabled, it also disables Flyby window functionality. To enable Flyby windows while debugging your project, the optimize storage feature must be turned off.

Right-click on the project filename in the Workspace window and select **Project Settings...** Click the Dynamics tab and select the option called **Store Feed-Forward Signals for Viewing**.

Node Loop Component Output Format

The output format of the Node Loop component was altered to reflect changes made to the subsystem splitting algorithm in PSCAD V4. This becomes important if your PSCAD V3 user components utilize the Node Loop for input. If this is the case, then you must alter your components before running any V4 projects.

In PSCAD V3, the subsystem global variable (SS) always retained the same value within the same circuit module. In PSCAD V4, the components on a single circuit can reside in different subsystems. As such, a new Loop prefix has been added to tell the compiler to use the subsystem number of the Node Loop component when processing.

Chapter 13: Migrating from Older Versions

V3 format (single subsystem):

```
$SS, $Loop:NA, $Loop:NB, $Loop:NC
```

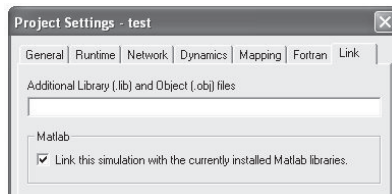
V4 format (multi-subsystem):

```
$Loop:SS, $Loop:NA, $Loop:NB, $Loop:NC
```

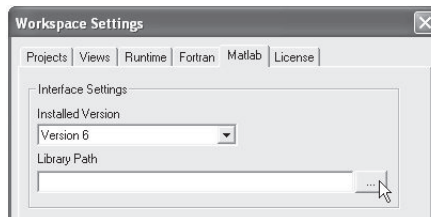
MATLAB Libraries and Interface

Project and Workspace settings should be adjusted slightly when migrating V3 projects that used the MATLAB interface. Please perform the following adjustments:

1. Open the respective Project Settings dialog for the project (right-click on the project filename in the Workspace window and select **Project Settings...**) and select the **Link** tab. Delete all of the MATLAB installation library paths in the **Additional Library (*.lib) and Object (*.obj) Files** input field. Ensure that the **Link this Simulation with the Currently Installed Matlab Libraries** check box is selected.



2. Open the Workspace Settings dialog by selecting **Edit | Workspace Settings...** and click the **Matlab** tab. In the Interface Settings area, select the **Installed Version** and **Library Path** to the MATLAB installation libraries.



References

1. "Hydraulic Turbine and Turbine Control Models for System Dynamic Studies," Working Group on Prime Mover and Energy Supply Models for System Dynamic Performance Studies, Transactions on Power Systems, Vol. 7, No. 1, February 1992.
2. "Dynamic Models for Fossil Fuelled Steam Units on Power System Studies," Working Group on Prime Mover and Energy Supply Models for System Dynamic Performance Studies, Transactions on Power Systems, Vol. 6, No. 2, May 1991.
3. P. Kundur, "Power System Stability and Control," McGraw Hill Inc., 1994.
4. "IEEE Recommended Practice for Excitation System Models for Power System Stability Studies," IEEE Std. 421.5-1992.
5. "Computer Models for Representation of Digital-Based Excitation Systems," IEEE Transactions, 1996.
6. Franz Karlecik-Maier, "A New Closed Loop Control Method for HVDC Transmission," IEEE Transactions on Power Delivery, Vol. 11, No. 4, October 1996, pp 1955-1960.
7. W. Tinney, N. Sato, "Techniques for Exploiting the Sparsity of the Network Admittance Matrix," IEEE PAS Vol. 82, December 1963, pp. 944-950.
8. W. Tinney, J. Walker, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization," Proc. IEEE Vol. 55, pp. 1801-1809, November 1967.
9. P. Kuffel, K. Kent, G. Irwin, "The Implementation and Effectiveness of Linear Interpolation Within Digital Simulation," Proceedings, International Conference on Power Systems Transients (IPST '95), Lisbon, September 3-7, 1995, pp. 499-504.
10. A. M. Gole, I. T. Fernando, G. D. Irwin, O. B. Nayak, "Modeling of Power Electronic Apparatus: Additional Interpolation Issues," Proceedings, International Conference on Power Systems Transients (IPST '97), Seattle, June 22-26, 1997, pp. 23-28.
11. B. A. Adkins, R. G. Harley, "The General Theory of Alternating Current Machines," Chapman and Hall, 1975.

Appendix A: References

12. J. R. Lucas, "*Representation of Magnetization Curves Over a Wide Region Using a Non-Integer Power Series,*" International Journal of Electrical Engineering Education, Vol. 25, pp. 335-340, Manchester U.K.
13. D. C. Jiles and D. L. Atherton, "*Theory of Ferromagnetic Hysteresis,*" Journal of magnetism and magnetic materials, vol. 61, pp. 48, 1986.
14. J. T. Thrope, D. C. Jiles and M. Devine, "*Numerical Determination of Hysteresis Parameters Using the Theory of Ferromagnetic Hysteresis,*" IEEE Transactions on Magnetics, vol. 28, pp. 27-35, 1992.
15. U. D. Annakkage, P. G. McLaren et al, "*A Current Transformer Model Based on the Jiles-Atherton Theory of Ferromagnetic Hysteresis,*" IEEE transactions on power delivery, Jan. 2000.
16. P. G. McLaren and R. P. Jayasinghe, "*Transformer Core Models Based on the Jiles - Atherton Algorithm,*" IEEE WESCANEX Communications, Power and Computing, May 1997.
17. P. G. McLaren, E. Dirks, U. D. Annakkage, R. P. Jayasinghe, A. Parker, "*Real time Simulation of CT's in a Current Differential Protection Scheme,*" Transmission and Distribution Conference, 1999 IEEE, Volume 2, 11-16, pp. 470-475, April 1999.
18. P.M. Anderson, Anjan Bose, "*Stability Simulation Of Wind Turbine Systems,*" Transactions On Power Apparatus And Systems. Vol. PAS 102, No. 12, December 1983, pp. 3791-3795.
19. "*Control Design And Performance Analysis Of A 6 Mw Wind Turbine Generator,*" IEEE Transactions On Power Apparatus and Systems, Vol. PAS 102, No. 5, May 1983, pp. 1340-1347.
20. A. Deri, G. Tevan, A. Semlyen, A. Castanheira, "*The Complex Ground Return Plane: A Simplified Model for Homogeneous and Multi-Layer Earth Return,*" IEEE Transactions on Power Apparatus and Systems, vol PAS-100, pp. 3686-3693, Aug. 1981.
21. (Pending)

Symbols

#CASE directive 470
#FUNCTION directive 462
#IF, #ELSEIF, #ELSE, #ENDIF directives 466
#LOCAL directive 461
#OUTPUT directive 464
#STORAGE directive 460
#SUBROUTINE directive 463
#TRANSFORMERS directive 465
#WINDINGS directive 466
~ 469

A

Absolute value 472
Active project 88,150
Aperture 268
Arc 363
Associations 82

B

Bus 315

C

C 431
Cables 325
 Constructing 330
 Cross-sections 337
 Editing properties 329,333
 Finding using workspace 112
 Migrating V2 521
 Model selection 341
 Output data 347
Case project 87,95
Categories 381,409
CBR 429
Chatter 311
Choice box 393
Comment operator 458
Compaq visual fortran 488

Compilers 19,20,46,53,57
 Limitations 14
Components 85
 Adding to project 162
 Connecting 166
 Creating new 182
 Cut/copy/paste 165
 Drag and drop 202
 Editing a definition 360
 Editing parameters 176
 Moving 165
 Printing 189
 Rotate/mirror/flip 165
 Selecting 162
 Viewing parameters/computations 179
Component definition 86,121,359,360
 Categories 381,409
 Choice box 393
 Conditional statements 401
 Connections 375
 Creating first instance 187
 Cut/copy/paste 98
 Directives 459
 Editing properties 178
 Field alignment 398
 Graphic objects 363
 Import/export 99,188
 Importing 516
 Input field 389
 Layers 404
 Line numbers 77
 Saving 400
 Segments 407,409
 Substitutions 447
 Syntax colouring 77,428
 Text field 387
 Text label 371
Component instance 86
Component ordering 483
Conditional statements 401
Connections 375

- Contact information 16
- Control groups 110,294
- Control panels 284
 - Displaying in reports 298
 - Grouping 294
 - Minimizing 297
 - Preparing data for display 233
 - Settings 213
 - Style 77
- Control settings 213
- Control signal flow indicators 486
- Cosine 472
- Cross-hairs 281
- Curves 245
- Curve fitting 356

D

- Data files 481
- Debugging 309,320,477,488
- Delete 166
- Design editor 121,359,381
- Detailed output files 352
 - Creating 353
 - Viewing 353
- Dfx 507
- Diagnostics 309
- Dimension limits 13
- Dongles 23
- Drag and drop 202
- Drop list 394
- Dynamic aperture adjustment 268

E

- Eigenvalues 352
- Eigenvectors 352
- Ellipse 363
- Emtdc system dynamics 483
- Emt directory 136,160
- Environment variables 57
- Error messages 119,120,152,477
- Exponential 472
- Expression evaluation 472
- External applications 82
- External files 319,322,430

F

- Feedback signals 486

- Fft 253
- Files 83,114,136,481
- File path 302
- File reference 84
- Find symbol 399
- Fixed node 378
- Flip 165
- Flybys 208,313
- Fortran 20,79
- Forum 15

G

- Global substitution 216
- Gnu fortran compiler
 - Installing 42,46
- Graphics 363,371,375,379,380,401
- Graphs 235,239
 - Aperture 268
 - Cross hairs 281
 - Curves 245
 - Displaying 298
 - Drag and drop 202
 - Grouping 296
 - Markers 270
 - Phasometer 255
 - Polymeter 252
 - Pop-up toolbars 282
 - Preferences 277
 - Preparing data for display 233
 - Style 77
 - Tool tips 267
 - Traces 251
 - Xy plot 260
 - Zoom 278
- Graph frames 235,272
 - Minimizing 297
- Grid 174
- Ground node 379
- Group names 111,296

H

- Hardware locks 23
- Hardware requirements 19
- Harmonics 253
- Help browser
 - Setting 75
- Help system 136,207

PSCAD

Hotkeys 132
Hyperbolic functions 472

I

Ideal branch 311
Idle time polling 310
Import
 Runtime batch files 510
 V2 draft files 510
 V2 user libraries 514
Infinite bus 311
Information file 231
Inf file 231
Initialization 308
Input field 389
Installation 20,27,42,53,54,57
Interpolation 311

L

Layers 401
Lib 322
Library files 322
library files 491
Library project 87,96
License key 24
License manager 20,25
 Installing 33,42
 Local 25
 Manually starting/stopping 71,72
 Standalone 25
 Trial 26
Licensing 20,81
 Active license 68,69
 Adding/upgrading 66,67
 Getting info 70
 License host 68
 License key 24
 Multi-user 20
 Single-user 22
 Trial 23,60
Line 363
Line constants program 344,345
 Constants file 347
 Editor 332
 Input file 345
 Log file 347
 Manually solving 344

 Output file 347
 Viewing detailed output 352,353
LiveWire
 Installing 42,46
Logarithm 472
Logical operators 473

M

Make -f message 56
Make file 482
Map file 482
Markers 270
Master library 101
Mathematical functions 472
Matlab interface 80,324,495,504,506
 Designing a component 499,502
 Simulink 495,497,502
Memory settings 78
Menus 122
Messages 35,118,119,120,152,477
Meta-file 189
Meters 283
Minimum requirements 19
Mirror 165
Modules 86
 Compiling 187
 Creating new 182
 Editing a definition 360
 Editing settings 173
 Page setup 175
 Printing 189
 Transferring data 168
 Viewing 103
Module definition 360
Mul 20
Multi-user licensing 20
Multiple run 307

N

Network splitting 317
Nmake -f message 56
Node 378
 Optimal ordering 318
Node loop 523

O

Obj 322

- Object files 322
- object files 491
- Observers 109
- Online plotting 233
- Optimization 317
- Output channel 78,294
- Output files 114,156,229,304
- Output window 118,477

P

- Phasor display 256
- Plotting 233,250,267,268,281,298
- Plot step 154,304
- Point to source 152
- Polymeters 252
- Pop-up toolbars 282
- Printing 93,189
- Projects 74,86,159
 - Active 150
 - Adding 190
 - Adding components 162
 - Automatic reload 74
 - Changing run duration 154
 - Compile/build 150
 - Creating new 143
 - Default directory 75
 - Description 302
 - Loading 144
 - Migrating from V2 507
 - Navigating 145,361
 - Opening 145
 - Output files 156,229
 - Plot step 154
 - Printing 93
 - Recently used 75
 - Running 54,153,303
 - Saving 156
 - Show signal locations 485
 - Unloading 161
 - View component/signal tables 210
- Project settings 301
 - Accessing 301
 - Editing 159
- Pscad community forum 15
- Push button control 289

R

- Radio buttons 394
- Radio links
 - Viewing 103
- Recent files 145
- Rectangle 363
- Redo 76,428
- Refresh 149
- Removable node 377
- Reports 298
- Revision tracking 303
- Rotary switch 285
- Rotate 165
- Round 472
- Router 21
- Rtb 510,519
- Run-ahead 310
- Runtime batch file 510,519
- Runtime messages 120

S

- Saving
 - As older versions 158
 - Automatic 75
 - Projects 157
- Script 77,407,428,447,451,459,469
- #CASE directive 470
- #IF, #ELSEIF, #ELSE, #ENDIF directives 466
- Expression evaluation 472
- Search 209,399,480
- Sequence numbers 174,483
- Serial port 24
- Shortcut keys 132
- Signal flow paths 315
- Signal locations 174
- Simulink 495
- Sine 472
- Single-line diagrams 182
- Single-user licensing 22
- Slider control 290
- Snapshots 154,155,306
- Software requirements 19
- Source code 431,508
- source code 491
- Source files 319,322,430,431
- Spectrum analyzer 252

PSCAD

- Square root 472
- Start simulation 305
- Storage
 - Signals 313
- Storage dimensions 312
- Substitutions 216,447
- Subsystems 317
- Sul 22
- Support 16
- Switched node 378
- Switching
 - Switching node optimization 318
- Switching devices 311
- Switch control 289
- Syntax colouring 77

T

- Tabs 135
- Tangent 472
- Temporary directory 136,160,230
- Ternary operator 474
- Text field 387
- Text labels 371
- Time offset 308
- Time step
 - Changing 154,304
- Tinney algorithm 318
- Toolbars 125
- Tool tips 208,313
- Towers 335
- Traces 251
- Transmission lines 325
 - Constructing 326
 - Editing properties 327,333
 - Finding using workspace 112
 - Migrating V2 521
 - Model selection 341
 - Output data 347
 - Towers 335
- Trial licensing 23,60
- Troubleshooting 40,120
- Tutorial projects 140
- Typical studies 3

U

- Undo 76,428
- Units 218

- Upgrading 42,507,522
 - Issues with pscad V2 508,510,514,516,519
- Usb port 24
- User library path 322

V

- Valid connections 166
- VDC 430
- Viewing Errors 152
- Virtual control wires 174,487
- Visibility condition 368

W

- Warning messages 35,119,120,152,477
- What's new 3
- Wire 486
- Wire mode 173
- Workspace 94,102,114
 - Tlines/cables 112
- Workspace settings 73,77,79,80,82

X

- Xnode 172
- Xy plotting 260

Z

- Zoom 148

