

# **gPROMS ModelBuilder Guide**

**Release v3.5**

**June 2012**



---

# **gPROMS ModelBuilder Guide**

Release v3.5

June 2012

Copyright © 1997-2012 Process Systems Enterprise Limited

Process Systems Enterprise Limited

6th Floor East

26-28 Hammersmith Grove

London W6 7HA

United Kingdom

Tel: +44 20 85630888

Fax: +44 20 85630999

WWW: <http://www.psenterprise.com>

## **Trademarks**

gPROMS is a registered trademark of Process Systems Enterprise Limited ("PSE"). All other registered and pending trademarks mentioned in this material are considered the sole property of their respective owners. All rights reserved.

## **Legal notice**

No part of this material may be copied, distributed, published, retransmitted or modified in any way without the prior written consent of PSE. This document is the property of PSE, and must not be reproduced in any manner without prior written permission.

## **Disclaimer**

gPROMS provides an environment for modelling the behaviour of complex systems. While gPROMS provides valuable insights into the behaviour of the system being modelled, this is not a substitute for understanding the real system and any dangers that it may present. Except as otherwise provided, all warranties, representations, terms and conditions express and implied (including implied warranties of satisfactory quality and fitness for a particular purpose) are expressly excluded to the fullest extent permitted by law. gPROMS provides a framework for applications which may be used for supervising a process control system and initiating operations automatically. gPROMS is not intended for environments which require fail-safe characteristics from the supervisor system. PSE specifically disclaims any express or implied warranty of fitness for environments requiring a fail-safe supervisor. Nothing in this disclaimer shall limit PSE's liability for death or personal injury caused by its negligence.

## **Acknowledgements**

ModelBuilder uses the following third party free-software packages. The distribution and use of these libraries is governed by their respective licenses which can be found in full in the distribution. Where required, the source code will be made available upon request. Please contact [support.gPROMS@psenterprise.com](mailto:support.gPROMS@psenterprise.com) in such a case.

Many thanks to the developers of these great products!

---

**Table 1. Third party free-software packages**

<b>Software/Copyright</b>	<b>Website</b>	<b>License</b>
<b>ANTLR</b>	<a href="http://www.antlr2.org/">http://www.antlr2.org/</a>	Public Domain
<b>Batik</b>	<a href="http://xmlgraphics.apache.org/batik/">http://xmlgraphics.apache.org/batik/</a>	Apache v2.0
Copyright © 1999-2007 The Apache Software Foundation.		
<b>BLAS</b>	<a href="http://www.netlib.org/blas">http://www.netlib.org/blas</a>	BSD Style
Copyright © 1992-2009 The University of Tennessee.		
<b>Boost</b>	<a href="http://www.boost.org/">http://www.boost.org/</a>	Boost
Copyright © 1999-2007 The Apache Software Foundation.		
<b>Castor</b>	<a href="http://www.castor.org/">http://www.castor.org/</a>	Apache v2.0
Copyright © 2004-2005 Werner Guttman		
<b>Commons CLI</b>	<a href="http://commons.apache.org/cli/">http://commons.apache.org/cli/</a>	Apache v2.0
Copyright © 2002-2004 The Apache Software Foundation.		
<b>Commons Collections</b>	<a href="http://commons.apache.org/collections/">http://commons.apache.org/collections/</a>	Apache v2.0
Copyright © 2002-2004 The Apache Software Foundation.		
<b>Commons Lang</b>	<a href="http://commons.apache.org/lang/">http://commons.apache.org/lang/</a>	Apache v2.0
Copyright © 1999-2008 The Apache Software Foundation.		
<b>Commons Logging</b>	<a href="http://commons.apache.org/logging/">http://commons.apache.org/logging/</a>	Apache v1.1
Copyright © 1999-2001 The Apache Software Foundation.		
<b>Crypto++ (AES/Rijndael and SHA-256)</b>	<a href="http://www.cryptopp.com/">http://www.cryptopp.com/</a>	Public Domain
Copyright © 1995-2009 Wei Dai and contributors.		
<b>Fast MD5</b>	<a href="http://www.twmacinta.com/myjava/fast_md5.php">http://www.twmacinta.com/myjava/fast_md5.php</a>	LGPL v2.1
Copyright © 2002-2005 Timothy W Macinta.		
<b>HQP</b>	<a href="http://hqp.sourceforge.net/">http://hqp.sourceforge.net/</a>	LGPL v2
Copyright © 1994-2002 Ruediger Franke.		
<b>Jakarta Regexp</b>	<a href="http://jakarta.apache.org/regexp/">http://jakarta.apache.org/regexp/</a>	Apache v1.1
Copyright © 1999-2002 The Apache Software Foundation.		
<b>JavaHelp</b>	<a href="http://javahelp.java.net/">http://javahelp.java.net/</a>	GPL v2 with classpath exception
Copyright © 2011, Oracle and/or its affiliates.		
<b>JXButtonPanel</b>	<a href="http://swinghelper.dev.java.net/">http://swinghelper.dev.java.net/</a>	LGPL v2.1 (or later)
Copyright © 2011, Oracle and/or its affiliates.		
<b>LAPACK</b>	<a href="http://www.netlib.org/lapack/">http://www.netlib.org/lapack/</a>	BSD Style
<b>libodbc++</b>	<a href="http://libodbcxx.sourceforge.net/">http://libodbcxx.sourceforge.net/</a>	LGPL v2

Software/Copyright	Website	License
Copyright © 1999-2000 Manush Dodunekov <manush@stendahls.net>		
Copyright © 1994-2008 Free Software Foundation, Inc.		
<b>lp_solve</b>	<a href="http://lpsolve.sourceforge.net/">http://lpsolve.sourceforge.net/</a>	LGPL v2.1
Copyright © 1998-2001 by the University of Florida.		
Copyright © 1991, 2009 Free Software Foundation, Inc.		
<b>MiGLayout</b>	<a href="http://www.miglayout.com/">http://www.miglayout.com/</a>	BSD
Copyright © 2007 MiG InfoCom AB.		
<b>Netbeans</b>	<a href="http://www.netbeans.org/">http://www.netbeans.org/</a>	SPL
Copyright © 1997-2007 Sun Microsystems, Inc.		
<b>omniORB</b>	<a href="http://omniorb.sourceforge.net/">http://omniorb.sourceforge.net/</a>	LGPL v2
Copyright © 1996-2001 AT&T Laboratories Cambridge.		
Copyright © 1997-2006 Free Software Foundation, Inc.		
<b>TimingFramework</b>	<a href="http://timingframework.dev.java.net/">http://timingframework.dev.java.net/</a>	BSD
Copyright © 1997-2008 Sun Microsystems, Inc.		
<b>VecMath</b>	<a href="http://vecmath.dev.java.net/">http://vecmath.dev.java.net/</a>	GPL v2 with classpath exception
Copyright © 1997-2008 Sun Microsystems, Inc.		
<b>Wizard Framework</b>	<a href="http://wizard-framework.dev.java.net/">http://wizard-framework.dev.java.net/</a>	LGPL
Copyright © 2004-2005 Andrew Pietsch.		
<b>Xalan</b>	<a href="http://xml.apache.org/xalan-j/">http://xml.apache.org/xalan-j/</a>	Apache v2.0
Copyright © 1999-2006 The Apache Software Foundation.		
<b>Xerces-C</b>	<a href="http://xerces.apache.org/xerces-c/">http://xerces.apache.org/xerces-c/</a>	Apache v2.0
Copyright © 1994-2008 The Apache Software Foundation.		
<b>Xerces-J</b>	<a href="http://xerces.apache.org/xerces2-j/">http://xerces.apache.org/xerces2-j/</a>	Apache v2.0
Copyright © 1999-2005 The Apache Software Foundation.		

This product includes software developed by the Apache Software Foundation, <http://www.apache.org/>.

gPROMS also uses the following third party commercial packages:

- **FLEXnet Publisher** software licensing management from Acresto Software Inc., <http://www.acresso.com/>.
- **JClass DesktopViews** by Quest Software, Inc., <http://www.quest.com/jclass-desktopviews/>.
- **JGraph** by JGraph Ltd., <http://www.jgraph.com/>.

---

# Table of Contents

1. Overview .....	1
2. Projects and the project tree .....	2
Projects .....	2
Project properties .....	3
Cross-referencing and hierarchical libraries .....	4
Specifying a Project's list of cross-referenced Projects .....	4
Search rules for cross-referenced Projects .....	4
Temporary suspension of Project cross-referencing .....	5
Creating a self-contained Project .....	5
Automatic loading of Projects .....	5
Read-only Projects .....	6
Display of empty groups in Project tree .....	6
Library projects .....	6
Model palette .....	7
Displaying the Model palette .....	7
Workspaces .....	8
3. gPROMS Entities .....	9
Entity creation and deletion .....	9
Allowable entity names .....	10
Opening and closing an entity editor .....	10
Entity editors .....	10
The entity properties tab .....	10
The gPROMS language tab .....	11
4. Constructing flowsheet Models .....	14
Component (library) Models .....	14
Constructing the flowsheet .....	15
Instances of component Models (Units) .....	15
Defining the connectivity of a composite Model .....	17
Hierarchical Model construction .....	21
Making a Unit into an Array .....	21
The Topology editor tool bar .....	27
Groups of Units .....	28
Making Model specifications .....	30
Flowsheet layers .....	31
Adding, renaming or deleting a layer .....	32
Assigning a Model entity to a layer .....	32
Specifying layer attributes .....	33
Adding graphs and other annotations to the flowsheet .....	33
Viewing results on the Flowsheet during and after simulation .....	34
Text annotations .....	35
Image annotations .....	37
Value annotations .....	37
Plot annotations .....	40
5. Executing simulations .....	42
To execute a simulation .....	42
Cross-reference check .....	42
Cases .....	42
The Case configuration and execution control dialog .....	42
Management of Cases .....	45
Creating Projects from Cases .....	46
Interacting with executing simulations .....	46
Execution output .....	47
Diagnostics console .....	48
Specifying Solution Parameters .....	49
Examples of Solution Parameter Specifications .....	51

---

Global Specification and Inheritance of Solution Parameters .....	55
Filtering the Display of Solution Parameters and Resetting Default Values .....	58
6. Viewing results .....	60
Inspecting results for an individual variable .....	60
Exporting the results .....	62
Printing the results .....	62
Viewing stream tables .....	63
Exporting stream tables .....	65
Viewing Model reports .....	65
7. Modelling Support Tools .....	66
MBG_Search_and_Replace. Global Search-and-replace .....	66
Project and entity Compare .....	67
Entity comparison .....	67
Project comparison .....	67
Entity group comparison .....	68
Import files .....	68
Create links to external files .....	69
Export .....	69
Export Entity with dependencies .....	69
Encryption .....	70
Hide output diagnostics .....	70
Export to ModelBuilder v2.3 Project .....	70
Export to Simulink .....	70
Export to CAPE-OPEN .....	72
Basic properties .....	72
Port mappings .....	73
Parameter mappings .....	74
Additional files .....	75
Advanced options .....	75
Entity generation options .....	76
Export options .....	78
The Simple Process Modelling Environment .....	79
8. Miscellaneous Utilities .....	88
ModelBuilder Preferences .....	88
Number formats .....	89
Text editor short-cut keys .....	90
Navigation shortcuts .....	90
Navigation shortcuts - Location shortcuts .....	91
Navigation shortcuts - Jump list shortcuts .....	92
Navigation shortcuts - Miscellaneous .....	92
Navigation shortcuts - Find shortcuts .....	92
Edit shortcuts .....	93
Edit shortcuts - Indentation shortcuts .....	93
Edit shortcuts - Capitalization shortcuts .....	94
Editing using external editor software .....	94
Printing .....	94
The Page Setup Dialog .....	95
Print Preview .....	98
The Print Dialog .....	100
Initial Print Selection .....	101
Exporting Data to CSV Files .....	101
Multiple selection .....	105
Desktop view .....	105
Collapse project tree action .....	106
9. gRMS Output Channel .....	107
gRMS processes .....	107
Plotting 2D graphs .....	108
Adding lines to a plot .....	109

---

Formatting lines .....	110
Formatting 2D plots .....	111
Plotting 3D graphs .....	114
Adding a surface to a plot .....	114
Formatting surfaces .....	114
Formatting 3D plots .....	115
Printing gRMS plots .....	116
Viewing and exporting data .....	117
2D plots .....	117
3D plots .....	117
Exporting images .....	117
Templates .....	118
Line templates .....	118
Plot templates .....	118
Advanced use of gRMS .....	122
Preventing gRMS from starting automatically with gPROMS .....	122
Starting gRMS independently from gPROMS .....	122
Running gPROMS and gRMS on different machines .....	123
Multiple gPROMS runs communicating with a single gRMS .....	123
gRMS resources under UNIX .....	123
10. Microsoft Excel Output Channel .....	126
Enabling the Microsoft Excel Output Channel .....	126
Format of the Microsoft Excel output .....	126
Additional options .....	126
Using the graph generation macro .....	128
11. gPLOT Output Channel .....	129

---

## List of Figures

2.1. gPROMS ModelBuilder project tree .....	2
2.2. New project tree entries .....	3
2.3. Project properties dialog .....	3
2.4. Specifying cross-referenced Projects .....	4
2.5. Creating a self-contained Project .....	5
2.6. Read-only Projects .....	6
2.7. Display of empty groups in Project tree .....	6
2.8. Creating Libraries .....	7
2.9. Model palette .....	7
3.1. Entities in project tree .....	9
3.2. Entity creation .....	9
3.3. Entity properties .....	11
3.4. The gPROMS language tab in an Entity Editor .....	11
3.5. Syntax error reporting .....	12
3.6. Automatic pathname completion .....	12
3.7. Locating declarations .....	13
4.1. The Topology tab .....	14
4.2. The Model Interface tab .....	14
4.3. Model topology .....	15
4.4. Unit shortcut menu .....	16
4.5. An invalid connection .....	17
4.6. Array of Ports .....	18
4.7. Additional information display .....	18
4.8. Connection shortcut menu .....	19
4.9. Hierarchical Model construction .....	21
4.10. Creating an Array of Units .....	22
4.11. Flowsheet with unconnected Arrays of Reactor_stirred_tank_kinetic and Pipe .....	22
4.12. Connection details dialog for the outlet of the CSTR Array to the inlet of the Pipe Array .....	23
4.13. Flowsheet with outlet of CSTR Array connected to inlet of Pipe Array .....	23
4.14. Connection details dialog for the outlet of the Pipe Array to the inlet of the CSTR Array .....	24
4.15. Connection details dialog for the outlet of the inlet Pipe to the inlet of the CSTR Array .....	25
4.16. Flowsheet with Source connected to first CSTR of the Array .....	25
4.17. Connection details dialog for the outlet of the CSTR Array to the inlet of the outlet Pipe .....	26
4.18. Final Flowsheet .....	26
4.19. The Topology editor toolbar .....	27
4.20. A selected group .....	29
4.21. A selected Unit within a nested group structure .....	30
4.22. A dialog box of a PID controller .....	30
4.23. Specification bounds violation .....	31
4.24. The Flowsheet Layers Dialog .....	32
4.25. Assigning a group of entites to a layer .....	32
4.26. The Annotations Palette .....	33
4.27. The Playback Toolbar .....	34
4.28. Flowsheet results with annotations and playback toolbar .....	35
4.29. The Editor tab .....	36
4.30. The Properties tab .....	37
4.31. The Value Annotation Dialog .....	38
4.32. Path Completion in Value Annotation Dialog .....	38
4.33. Style Options in Value Annotation Dialog .....	39
4.34. The Plot Annotation Dialog .....	41
5.1. Simulating: executing a process .....	42
5.2. Automatic cross-reference checking using gPROMS .....	42
5.3. The execution control dialog .....	43
5.4. Read-only Cases .....	45
5.5. Cases and activity execution .....	45



5.6. ModelBuilder preference dialogs relating to Case management and activity execution .....	46
5.7. Cases and activity execution .....	47
5.8. Cases and activity execution .....	48
5.9. Diagnostics console Toolbar button .....	48
5.10. Diagnostics console .....	49
5.11. Solution Parameters tab .....	50
5.12. Solution Parameters tab .....	50
5.13. Opening the Output generation section .....	51
5.14. Selected Solution Parameter .....	52
5.15. Entering a Value .....	52
5.16. Solution Parameter changed .....	53
5.17. Invalid entry of Solution Parameter values .....	53
5.18. Multiple choice value specification .....	54
5.19. Entering a value instead of the two provided .....	54
5.20. Solution Parameters for Dynamic Optimisation .....	55
5.21. Changing top-level DASolver OutputLevel .....	56
5.22. Inheritance of OutputLevel in DASolver parameter .....	56
5.23. Inheritance of OutputLevel in DASolver parameter .....	57
5.24. Inheritance of OutputLevel in DASolver parameter .....	58
5.25. Inheritance of OutputLevel in DASolver parameter .....	58
6.1. Finding a single variable in the project tree .....	60
6.2. Simulation results: table .....	61
6.3. Simulation results: graph .....	62
6.4. Stream tables tab .....	63
6.5. Docked stream tables .....	64
6.6. Detached stream table .....	64
6.7. A model report .....	65
7.1. The search and replace tool .....	66
7.2. Comparison of two Entities .....	67
7.3. Comparison of two Projects .....	68
7.4. Binary files imported in Project .....	68
7.5. Link to external files .....	69
7.6. The export tool .....	69
7.7. Open encrypted file dialog .....	70
7.8. gPROMS Model with Ports .....	71
7.9. Export to Simulink dialog .....	71
7.10. Export CAPE-OPEN Unit Operation Wizard: Basic properties page .....	73
7.11. Export CAPE-OPEN Unit Operation Wizard: Ports page .....	73
7.12. Export CAPE-OPEN Unit Operation Wizard: Parameters page .....	74
7.13. Export CAPE-OPEN Unit Operation Wizard: Additional files page .....	75
7.14. Export CAPE-OPEN Unit Operation Wizard: Advanced page .....	76
7.15. Export CAPE-OPEN Unit Operation Wizard: Entity generation page .....	78
7.16. Export CAPE-OPEN Unit Operation Wizard: Export page .....	78
7.17. Example 'unit operation list' output from SimplePME .....	79
7.18. Example 'initialization' output from SimplePME .....	80
7.19. Example 'property package list' output from SimplePME .....	82
8.1. ModelBuilder Preference dialog .....	88
8.2. Page Setup Dialog .....	96
8.3. The Print Preview Dialog .....	98
8.4. Print Preview Controls .....	99
8.5. The Tree View (with default selection for Projects) .....	100
8.6. Print Dialog .....	101
8.7. Export to CSV Dialog .....	102
8.8. Example of Custom Range specification .....	104
8.9. Multiple editor mode .....	105
8.10. Single editor mode .....	106
9.1. Temperature profile in tubular reactor .....	107
9.2. 2D graph .....	108

9.3. New 2D Plot Window .....	108
9.4. Add line .....	109
9.5. Line properties .....	110
9.6. Line styles .....	110
9.7. Axis Format Dialog (Windows) .....	111
9.8. Axis Format Dialog (UNIX) .....	112
9.9. Font selection .....	113
9.10. Grid format .....	113
9.11. Legend .....	113
9.12. Title and footer .....	114
9.13. Plot rotation .....	115
9.14. Print dialog .....	116
9.15. Instantiating a plot template .....	119
9.16. Create a plot of P and T from one model. ....	120
9.17. Save the plot as a template. ....	120
9.18. Open the template. ....	121
9.19. Instantiate the template from a different model with P and T. ....	121
9.20. View the resulting plot. ....	122

---

## List of Tables

1. Third party free-software packages .....	3
5.1. Effects of Output level on execution diagnostics .....	45
7.1. Fields in Export to Simulink Dialog .....	71
8.1. ModelBuilder preferences .....	88
8.2. Shortcuts with standard navigation keys .....	90
8.3. Insertion point/screen position shortcuts .....	91
8.4. Jump list shortcuts .....	92
8.5. Miscellaneous navigation shortcuts .....	92
8.6. Find shortcuts .....	92
8.7. Shortcuts with standard edit keys .....	93
8.8. Indentation shortcuts .....	93
8.9. Capitalization shortcuts .....	94
9.1. Process menu items .....	108
9.2. Selection field items .....	109
9.3. Line menu items .....	110
9.4. Axis Format Dialog (Windows) Entries .....	111
9.5. Default Line Styles .....	112
9.6. Transformation desired .....	114
9.7. Surface menu items .....	114
9.8. Command line switches .....	122
9.9. Resources mimicking the command line switches. ....	124
9.10. Resources controlling individual windows and dialogs. ....	124
9.11. Names of individual windows and dialogs. ....	124
9.12. Resources controlling general appearance. ....	125
11.1. Format of gPLOT files .....	129

---

## List of Examples

7.1. Example 'calculation' input file for SimplePME .....	83
7.2. Example 'calculation' output from SimplePME .....	86

---

# Chapter 1. Overview

The gPROMS ModelBuilder is at the centre of:

- all model development and maintenance activities;
- the archiving of models and results;
- the execution of all model-based activities.

This guide gives a description of the features and functionality of gPROMS ModelBuilder - including some of the advanced tools that are available:

- An introduction to the ModelBuilder environment looks at Projects and the general properties of all Entities.
- Constructing flowsheet Models explains how to graphically develop and configure hierarchical flowsheets from (library) Models.
- Executing simulations explains how to configure solvers and result management, and how to execute simulations.
- In Viewing results it is shown which options exist to view results of Simulations, including plots, Stream Tables and customised Model Reports.
- Building Models is supported by various tools and utilities.

The ModelBuilder Guide is the first of a comprehensive set of manuals that cover all aspects of gPROMS:

- ModelBuilder Guide
- ModelDeveloper Guide
- Model Validation Guide
- Process Model Library Guide
- Optimisation Guide
- Physical Properties Guide
- gO:Simulink Guide
- Foreign Objects and Foreign Processes
- *gO:RUN Guide*
- *gPROMS System Programmer Guide*
- *gPROMS Server Guide*

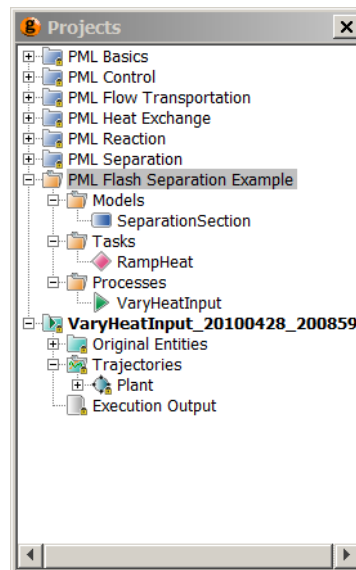
Guides in *italics* are currently available as PDF documents in the gPROMS documentation folder.

---

# Chapter 2. Projects and the project tree

The project tree allows the user to navigate all the (Library) Projects and Cases that have been respectively opened and created during a ModelBuilder session. They are distinguished by their colours: yellow (Projects); green (Library projects); and blue (Cases) and their ordering: Library projects, Projects and Cases. Projects of the same type are then ordered alphabetically.

**Figure 2.1. gPROMS ModelBuilder project tree**



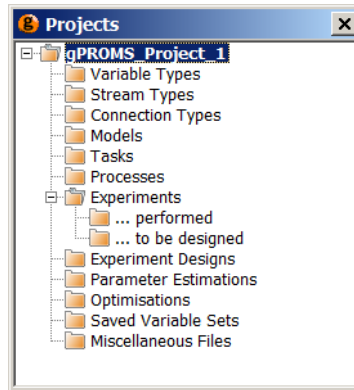
## Projects

To create a new gPROMS project select New from the Project menu. This will bring up a tree in the left-hand pane containing a number of folders:

- Variable Types
- Stream Types
- Connection Types
- Models
- Tasks
- Processes
- Optimisations
- Parameter estimations
- Experiments
  - ... performed
  - ... to be designed

- Saved Variable Sets
- Miscellaneous Files

**Figure 2.2. New project tree entries**



The Project can be renamed from its default of *gPROMS\_Project\_1.gPJ* by:

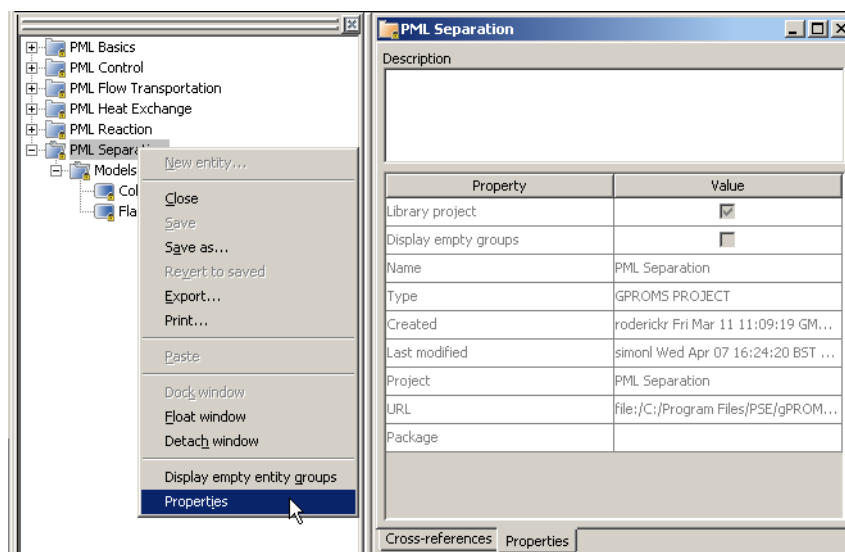
- Selecting Save As from the Project menu
- Entering a new File Name
- Clicking Save

## Project properties

Each Project records information relating to the user(s) who created and last modified it, as well as the dates and times of creation and last modification. This information is read-only and is recorded and maintained automatically by ModelBuilder under the Project's Properties.

Project Properties can be viewed by right-clicking on the Project's name in ModelBuilder's navigation tree and selecting Properties in the context-sensitive menu that appears.

**Figure 2.3. Project properties dialog**



The Properties dialog has a second tab that controls the list of Projects that can be cross-referenced.

## Cross-referencing and hierarchical libraries

In gPROMS ModelBuilder, the Entities in one Project may make references to Entities that belong to another Project. For example,

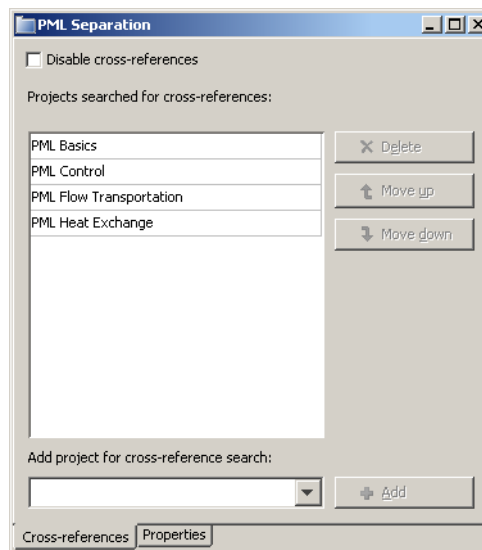
- a Model Entity in Project A may refer to Variable Types in a Project B in declaring its Variables;
- a Model Entity in Project A may refer to Models in a Project B in declaring its Unit sub-models;
- a Task or Process Entity in Project A may refer to Models and/or Tasks in Project B;

and so on. This provides a convenient way of building libraries of Variable and Connection types, Models and Tasks - or indeed any type of Entity - within gPROMS. Please note that only Projects currently loaded within ModelBuilder can be searched for Entities.

## Specifying a Project's list of cross-referenced Projects

A Project may refer to Entities residing in any number of other Projects. As shown in the figure the latter can be specified explicitly via the Cross-references tab of the Project's Properties dialog. Note that only Projects that are currently open within the ModelBuilder session can be selected in this manner.<sup>1</sup>

**Figure 2.4. Specifying cross-referenced Projects**



The ordering of a Project's list of cross-referenced Projects is important in that it determines the order in which the various Projects in the list will be searched. Each new Project added is inserted immediately below the Project in the list that is currently highlighted. However, the position of any Project in the list may be modified at any time via the Raise and Lower buttons provided.

## Search rules for cross-referenced Projects

In technical terms, ModelBuilder supports multiple libraries organised in hierarchies of arbitrary depth which are searched in a depth-first manner. This sophisticated capability is best explained via an example. Suppose a Model Entity M1 in a certain Project A contains a Unit sub-model that is of type Model M2. ModelBuilder starts by looking for M2 within A itself. If this search is not successful, then it starts looking for M2 within the Projects that can be cross-referenced by A in the order in which these have been specified (see previous section). If one of these other Projects, say B, has its own list of cross-referenced Projects, then these will also be searched - and

<sup>1</sup>This is consistent with the fact that only Projects that are currently open are actually searchable.



this will be done before moving on to the next Project in A's list<sup>2</sup>. The search for a particular Entity ends when M2 is either found or there are no more Projects to be searched.

Assuming that Model M2 has been found, it is entirely possible that it, in turn, refers to lower-level Entities such as other Models, or variable or stream types. The search for these Entities will follow exactly the same rules as that for M2 itself. It is worth noting that the search will always commence from the current Project A and not from the Project in which M2 was found. This is designed to allow, for example, a user's Project to make use of a standard library Model (e.g. "Reactor") while overriding the library definition of a variable type (e.g. "Temperature") or indeed of a sub-model (e.g. "KineticMechanism") by providing its own Entity of the same name.

## Temporary suspension of Project cross-referencing

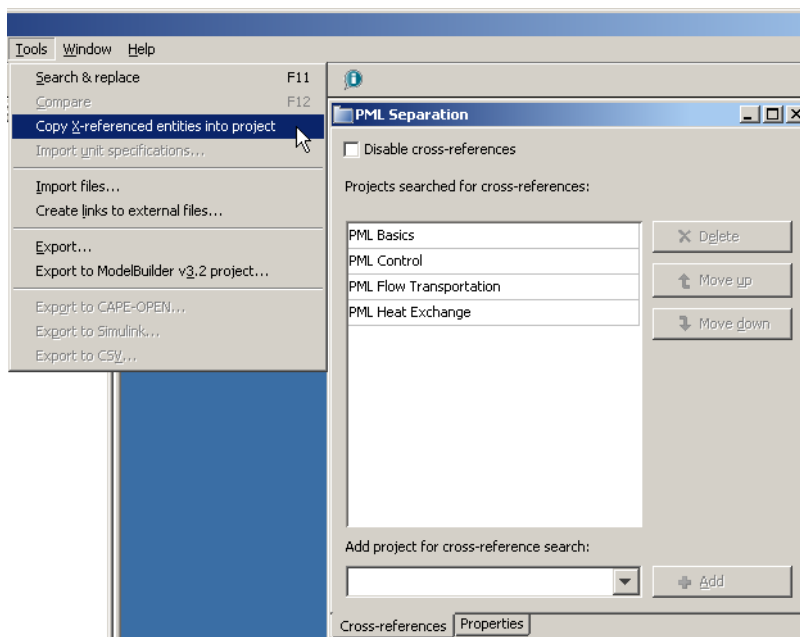
At any point during a ModelBuilder session, the user may instruct ModelBuilder not to search other Projects for any missing Entities until further notice. This can be done by checking the Disable cross-references check-box in the cross-references tab of the Project's Properties dialog.

## Creating a self-contained Project

It is sometimes desirable to create a completely self-contained Project that contains physical copies of all necessary Entities, including any that may reside in cross-referenced Projects.

The creation of a self-contained Project may be performed automatically using the Copy X-referenced Entities into Project utility under the ModelBuilder's Tools menu. This operation cannot be reversed automatically.

**Figure 2.5. Creating a self-contained Project**



## Automatic loading of Projects

The list of Projects to be automatically loaded whenever ModelBuilder is started can be specified as part of the user's ModelBuilder Preferences. This feature is particularly useful for library Projects.

The ModelBuilder Preferences dialog can be accessed from the Edit menu. Simply highlight the Projects category; and add or delete Projects in the Start-up projects list using the corresponding buttons. The file navigation dialog

---

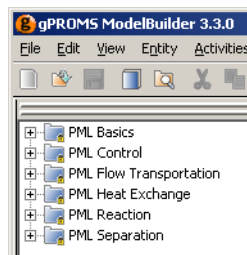
<sup>2</sup>gPROMS ModelBuilder will automatically detect and deal with any circular references. In particular, no Project is searched more than once while searching for a particular Entity.

that appears when the Add button is pressed permits the selection of any Project (.gPJ) file that is currently visible to the user - including those that reside on shared file systems.

## Read-only Projects

In order to protect against inadvertent modifications being made to a particular Project or Library project, it is recommended that the corresponding gPJ file be declared as *read-only* using standard operating system facilities prior to loading the project into ModelBuilder. In such cases, a *lock* symbol will be displayed against both the Project and any Entity that it contains in the ModelBuilder's navigation tree. Any operation that may result in a modification of this Project is automatically disabled.

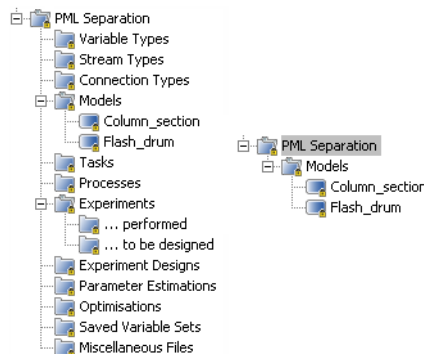
**Figure 2.6. Read-only Projects**



## Display of empty groups in Project tree

The Properties dialog can also be used to control the appearance of the Project in ModelBuilder's navigation tree. In particular, the display of empty folder groups (e.g. the TASK group in a Project that does not actually contain any TASK Entities) can be enabled or disabled by ticking the relevant check-box. The same effect can be achieved by ticking the Display empty groups entry in the Project's context-sensitive menu.

**Figure 2.7. Display of empty groups in Project tree**



## Library projects

Project libraries are projects marked by the user as such to indicate that they contain potentially re-usable components. Project libraries are marked in green and always appear at the top of the Project tree. Their purpose is to enhance the management, usability and operability of generic Models and Tasks. They can contain all entity types and can cross-reference other libraries/projects. The gPROMS Process Model Library (PML) makes extensive use of this new concept.

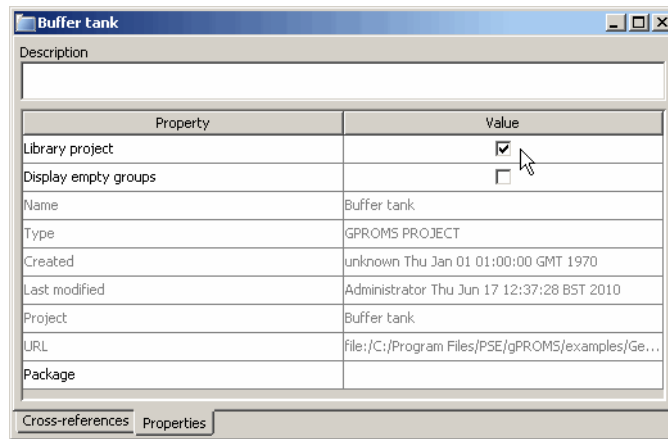
Entities in a library project can be used by another project opened in ModelBuilder if:

- the library is open in ModelBuilder, and
- the library is cross-referenced by the project

*Creation of libraries:* Any project can be converted into a library and vice versa. To create a library from a project:

- Open the Properties tab of the project,
- Check the library project checkbox to convert to project into a library<sup>3</sup>.

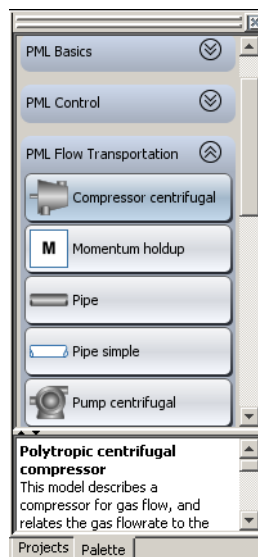
**Figure 2.8. Creating Libraries**



## Model palette

The Model palette graphically displays the Model Entities contained in open Projects (and Library projects) and provides an alternative view to the Project tree. All Models that have an icon defined are shown on the Model palette<sup>4</sup>. The Models are grouped by the Projects to which they belong. The Model palette provides a graphical view of models that can be used to build flowsheets.

**Figure 2.9. Model palette**



## Displaying the Model palette

To display the Model palette:

- click on View in the ModelBuilder top bar menu

<sup>3</sup>Similarly, a library can be converted into a project by following un-checking the library projectcheckbox.

<sup>4</sup>Models will only appear on the palette if the Show in palette option is checked on their Properties tabs.

- select Palette in the scroll-down menu (a check next to the option marks if it is on or off)

The model palette window will appear on the left hand side. If the Project tree is also open, a tab at the bottom of the window will allow the user to toggle between the Project tree and the Model palette.

The arrow to the left of the Project name allows the palette for each Project to be collapsed and expanded. Selecting a Model on the Palette displays the description of the Model Entity.

## Workspaces

Workspaces aim at simplifying the workflow when a certain set of Projects, Libraries and Cases is repeatedly loaded in ModelBuilder. When a Workspace is first saved, it combines all currently open Projects, Libraries and Cases in a single file with the extension `.gWS`<sup>5</sup>.

Workspaces can be saved or opened through the File menu.

---

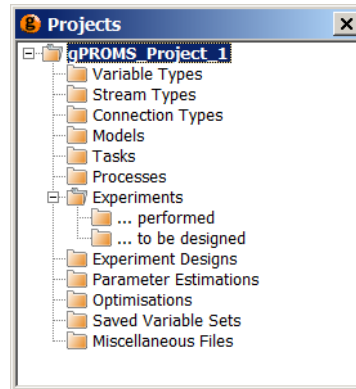
<sup>5</sup>Note that only links to the files are stored, and not the files themselves. When a Workspace file is opened, all files referenced in the Workspace are loaded.

---

# Chapter 3. gPROMS Entities

Each entry on the project tree represents a group of gPROMS *Entities*. Each entity type represents a fundamental gPROMS concept. ModelBuilder provides a customised *entity editor* for working with each entity type.

**Figure 3.1. Entities in project tree**

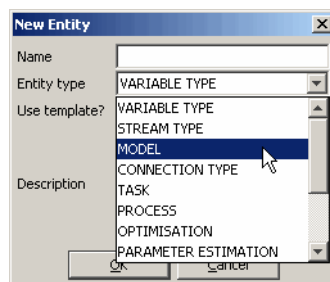


## Entity creation and deletion

To create a new entity<sup>1</sup>:

- place the pointer on the desired folder
- click the right mouse button
- select New entity from the shortcut menu
- give a name to the new entity
- select the type of entity in the drop-down menu
- fill in a description of the entity if desired and click OK

**Figure 3.2. Entity creation**



To delete an entity<sup>2</sup>:

- select the entity
- click the right mouse button
- select Delete from the shortcut menu
- confirm the delete by selecting Yes (alternatively the delete can be cancelled)

---

<sup>1</sup>Alternatively, select the desired project, select New entity from the Entity menu.

<sup>2</sup>Alternatively, select the entity press “Delete” in the keyboard.

*Note:* select several entities simultaneously with the **Shift** and/or **Ctrl** key to delete all.

To rename an existing entity<sup>3</sup>:

- place the pointer on the desired entity
- click the right mouse button
- select Rename from the shortcut menu
- fill in the desired name in the dialog window and click OK

## Allowable entity names

All entities in gPROMS (Models, Tasks, Parameters, Variables etc.) are subject to the same restrictions on the names they can be given.

- Names may only contain letters, numbers and the underscore character "\_"
- All names must begin with a letter
- Upper and lower case letters may be used but all names are case insensitive: that is, VariableOne, variableone and VARIABLEONE will all be considered the same name.

## Opening and closing an entity editor

To open an entity editor<sup>4</sup>:

- place the pointer on the desired entity
- click on the right mouse button
- select Open on the shortcut menu

ModelBuilder supports both *multiple* and *single editor* modes which can be selected from the Window menu. When working in *multiple editor* mode, an entity editor can be closed by clicking on the cross in the top right corner.

## Entity editors

ModelBuilder provides customised editors for each type of entity. Each editor provides a number of tabs, allowing the user to view or alter the different aspects of the entity's behaviour.

This section will cover the two tabs which are present on most of ModelBuilder's entity editors – the gPROMS language and Properties tabs. Tabs specific to individual entities will be described in later in this guide.

## The entity properties tab

All Entity editors in gPROMS ModelBuilder have an Entity Properties tab that includes the following information:

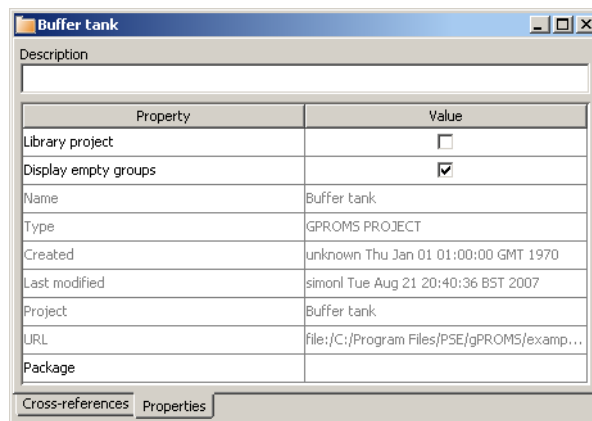
- description of the Entity; this is an arbitrary text provided by the Entity developer(s) for future reference.
- Entity creation and last modification information, including the user who performed these actions, and their times and dates; this information is read-only as it is constructed and maintained automatically by ModelBuilder.
- a list of all other Entities on which this Entity depends; again this is read-only information that is constructed and maintained automatically by ModelBuilder.<sup>5</sup>

---

<sup>3</sup>Alternatively, select the desired entity; select Rename from the Entity menu.

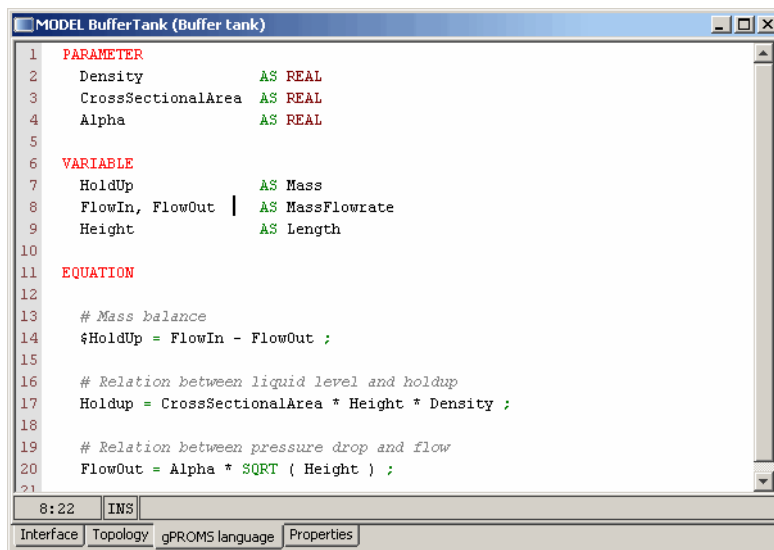
<sup>4</sup>Alternatively, double clicking on an entity will open its editor.

<sup>5</sup>The appearance of a referenced Entity in this list simply indicates that it is required for the correct operation of the current Entity; it does not necessarily imply that the referenced Entity actually exists and/or can be located at this point in the ModelBuilder session.

**Figure 3.3. Entity properties**

## The gPROMS language tab

Almost all Entity editors in gPROMS ModelBuilder have a tab that displays and allows the editing of the representation of the Entity in the gPROMS language. For an introduction to the gPROMS language - refer to the ModelDeveloper Guide.

**Figure 3.4. The gPROMS language tab in an Entity Editor**

## Syntax highlighting

gPROMS ModelBuilder automatically employs syntax-sensitive highlighting of the gPROMS language to support the creation and modification of each Entity.

A default set of colours is assigned for different types of gPROMS constructs. The user can overwrite these default settings, or indeed switch syntax highlighting off, by going to the ModelBuilder Preferences dialog and highlighting the *Entity editor* category.

## Syntax checking

gPROMS ModelBuilder will automatically check the syntax in any of the Entities that have been written. In addition, a wide range of semantic checks are performed, e.g. flagging the use of unidentified local variables.<sup>6</sup>

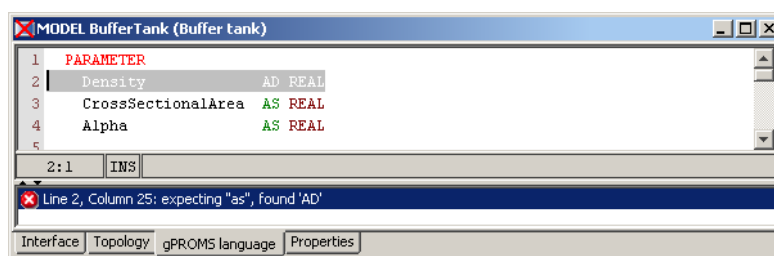
<sup>6</sup>The semantic checking performed is currently local within the current Entity. ModelBuilder does not attempt to validate any cross-referenced Entities.

Syntax and semantic checking is invoked by a number of different methods:

- When the user saves the Project.
- By clicking on the check syntax button just under Tools on the top toolbar.
- By selecting Check Syntax from the Entity menu.
- Right clicking on the Entity and selecting Check syntax.
- By using the keyboard short-cut (**F4**).

If ModelBuilder finds an error, a small pane appears just underneath the text editor reporting the error. Double-click the error message in the error pane and the cursor will automatically go to the corresponding line number to show where the syntax error is.

**Figure 3.5. Syntax error reporting**



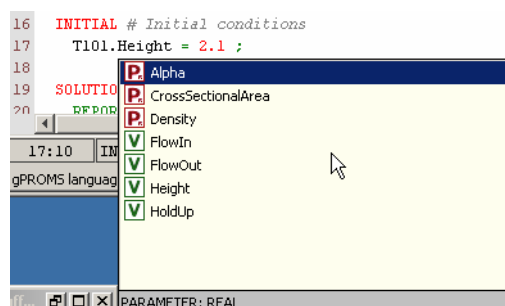
You will also see that the error is highlighted in the text editor window and that a red cross appears through the icon for the Entity in the Project tree. Correcting this error will then cause this cross and the error pane to disappear.

## Assisted pathname completion

Most gPROMS entity editors support assisted pathname completion. The pathname completion applies to all Units, Ports, Variables, Parameters, Selector names and values, Variable Types and Distribution Domains declared in a specific Model or Process and all its sub-models, as well as cross-referenced entities.

This feature is invoked by pressing **Ctrl-Space**. If there is only one possibility for the name currently being typed, the remainder of the word will be entered automatically. If there are multiple choices, a list box will appear showing all the possible Units and Variable names for completion (see the figure below). Typing additional characters will narrow down the selection, and the user may use the cursor keys to navigate up and down the list box to select an item. Pressing the **Space** or **Enter** key will complete the partially entered name using the selected item in the list. Pressing the dot key (.) will complete a Unit or Port name and will automatically show a new suggestion list for the next item in the path. Pressing the **Escape** key will close the list without making any changes.

**Figure 3.6. Automatic pathname completion**



The suggestion list will include gPROMS keywords, if at least one character has been typed before pressing **Ctrl-Space**.



The suggestion list will automatically appear after typing a Unit name followed by a dot, and waiting for a short period. The user may disable this feature, or change the delay, by changing the defaults in the Entity Editor category in the ModelBuilder Preferences menu.

The size of the suggestion list can be adjusted by clicking and dragging the grey area at the bottom of the list.

## Word Match

This is similar to automatic pathname completion in that it completes the current word being typed. However, it applies to any word that may have previously been typed (not just Units, Ports, Variables, Parameters, Selector names and values and Distribution Domains).

Word match is invoked manually by pressing:

- **Ctrl-K** to find the previous word that begins like the current word and complete it so that they match.
- **Ctrl-L** to find the next word that begins like the current word and complete it so that they match.

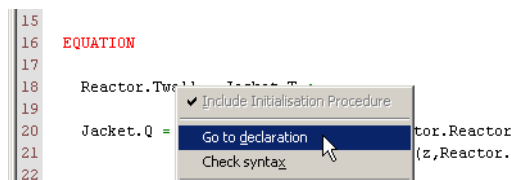
As well as completing the word, word match will also match the case. This is particularly useful for gPROMS language keywords where it is usually preferred to have the whole word in upper case.

## Locating declarations

The declaration of Units, Variables, Parameters, Selector names and values, Variable Types and Distribution Domains declared in a specific Model and all its sub-models, as well as in cross-referenced entities, can quickly be accessed by right-clicking on an element in an Entity editor.

In the example below, a right-click on *Twall* directly opens the declaration of the Variable *Twall* in the *Reactor* Model.

**Figure 3.7. Locating declarations**



# Chapter 4. Constructing flowsheet Models

gPROMS ModelBuilder allows users to build composite Models graphically. This functionality is designed to enable one quickly to construct process flowsheets that are suitable for use in any Model based activity (e.g. steady-state and dynamic simulation, parameter estimation and optimisation studies). Recall the standard gPROMS language and Properties tabs that are used to describe Model Entities: the Topology tab is used to put flowsheets together graphically and to specify the information needed for successful Model based activity.

**Figure 4.1. The Topology tab**



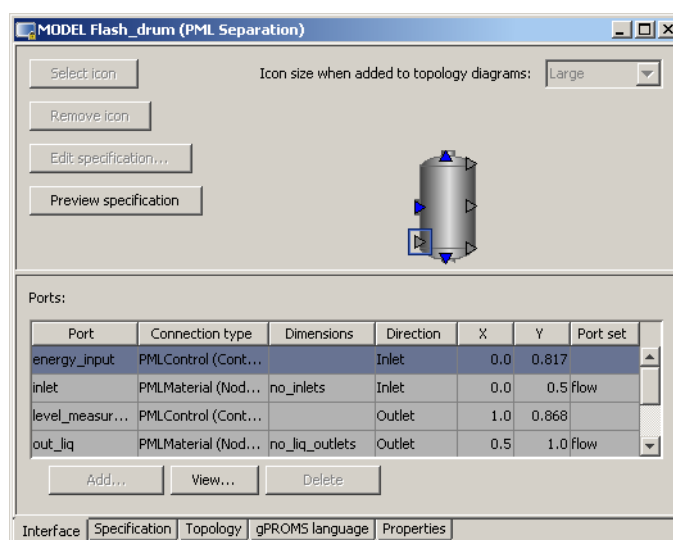
## Component (library) Models

A component Model is primarily a set of equations<sup>1</sup> (viewable in the gPROMS language tab) that describe the physical and chemical behaviour of the unit which it is representing. Typically the component Models are taken from an existing corporate library, the users own collection or the standard gPROMS Process Model Library (PML).

A flowsheet is a composite Model made up from constituent component Models that represent (part of) a process made up of connected unit operations. This ModelBuilder Guide outlines how to construct flowsheets graphically using stream connections on the Topology tab, although they can also be put together directly in the gPROMS language (refer to the Model Developer Guide for instructions on how to do this).

The Interface tab for a model shows how it will appear when used on the Topology tab of a composite Model. Model Ports are shown on the icon: they determine how connections are made to/from this component Model.

**Figure 4.2. The Model Interface tab**



Ports properties are shown in the table underneath the icon, detailing their Connection Type, Dimensions and Direction<sup>2</sup> as well as where they appear on the Model icon.

<sup>1</sup>please refer to the Model Developer Guide for more coverage of Model equations in the gPROMS language

<sup>2</sup>Ports are instances of Connection Type Entities.

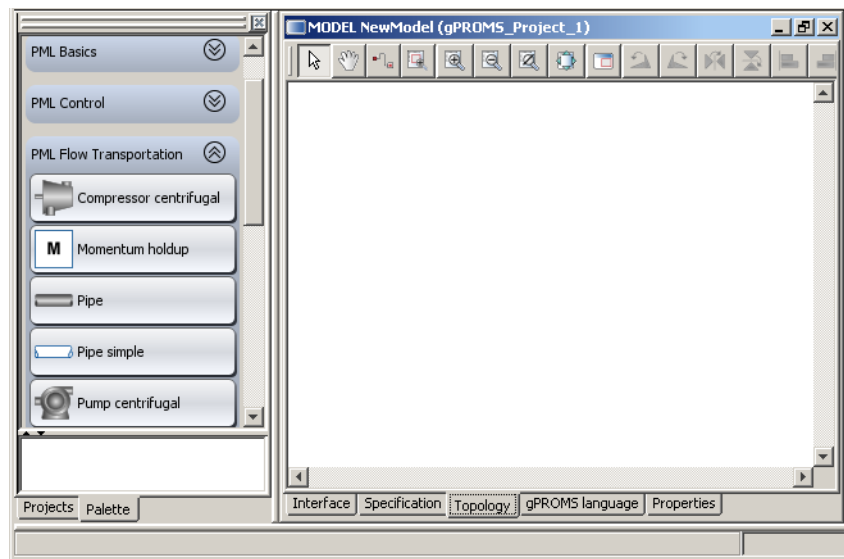
The Preview specifications button allows the user to preview the Model specification dialog associated with the Model.

## Constructing the flowsheet

Composite Models can be constructed by dragging (with the mouse) component Models from the Project tree, or Model palette, and dropping them onto a graphical topology editor. The connectivity of the composite Model can also be specified by connecting the Ports of the component Models.

The figure shows the Topology editor for a blank Model along with the Project tree displaying the Process Model Library.

**Figure 4.3. Model topology**



## Instances of component Models (Units)

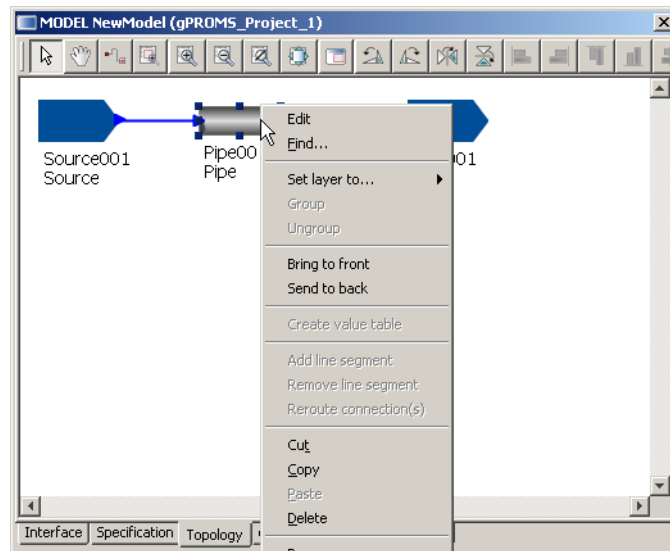
To add a Model instance (a *Unit*) to the topology editor:

- select the Model in the Project tree or from Model palette,
- drag it to the desired position on the Topology editor and drop it by releasing the mouse button

If the name of the Model is `MyModel`, the new Unit is automatically named `MyModel001`. If a second Model of the same type is dragged onto the Topology, the new Unit will be automatically named `MyModel002`, and so on.

## The Unit shortcut menu

Once positioned on the topology tab, changes can be made to a Unit using its shortcut menu. To display the shortcut menu, left-click on the Unit to select it and then right-click anywhere in the topology editor to activate the menu. The menu will apply only to the currently selected unit, indicated by the black squares surrounding it, even if you right-click on another unit (or connection). The figure below illustrates the shortcut menu for the selected pipe unit. The mouse pointer shows where the right click occurred.

**Figure 4.4. Unit shortcut menu**

The shortcut menu is divided into a number of sections. They may apply to:

- Units only
- Connections only (see Connection shortcut menu)
- Any type of object in the topology flowsheet
- The topology flowsheet itself

The sections that apply to Units are as follows.

- Edit

Left -clicking on the Edit command activates the Unit Specification Dialog. This has the same effect as double-clicking on the Unit.

- Layers and Groups



Units can be assigned to different Flowsheet Layers or combined to form Groups using these controls.

- Cut/Copy/Paste/Delete

This section contains the standard controls found in the Edit menu of most Windows applications. The commands behave in exactly the same way as most applications. Their functions are listed here for completeness.


- Cut # Copies the selected Unit(s) to the clipboard and then deletes it/them.
- Copy # Copies the selected Unit(s) to the clipboard, leaving it/them unchanged.
- Paste # Pastes the Unit or Units that are in the clipboard into the Flowsheet.
- Delete # Deletes the selected Unit(s).

Note that if a cut or deleted Unit is connected to any other Units, then these connections will also be deleted. A Unit can also be deleted by selecting the Unit on the topology and pressing **Delete**.

Note that it is possible to undo/redo any command or action using the  and  buttons on the ModelBuilder tool bar menu.

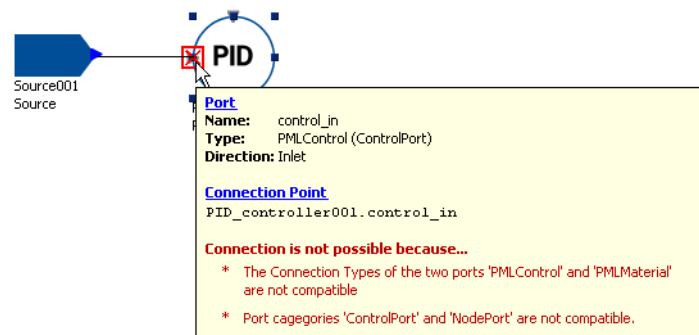
## Defining the connectivity of a composite Model

Once more than one Unit has been added to a composite Model, the connectivity of the system can be defined by connecting the Ports of each Unit. This can be done in the following manner:

- select the  button on the Topology editor tool menu: this will switch the pointer to connection mode
- select the starting Port
- move the pointer to the destination Port
- select the destination Port

If the line becomes solid then the connection has been verified. If the connection is invalid, ModelBuilder will issue a pop-up message giving an explanation.

**Figure 4.5. An invalid connection**



ModelBuilder applies the following rules to determine the validity of a connection:

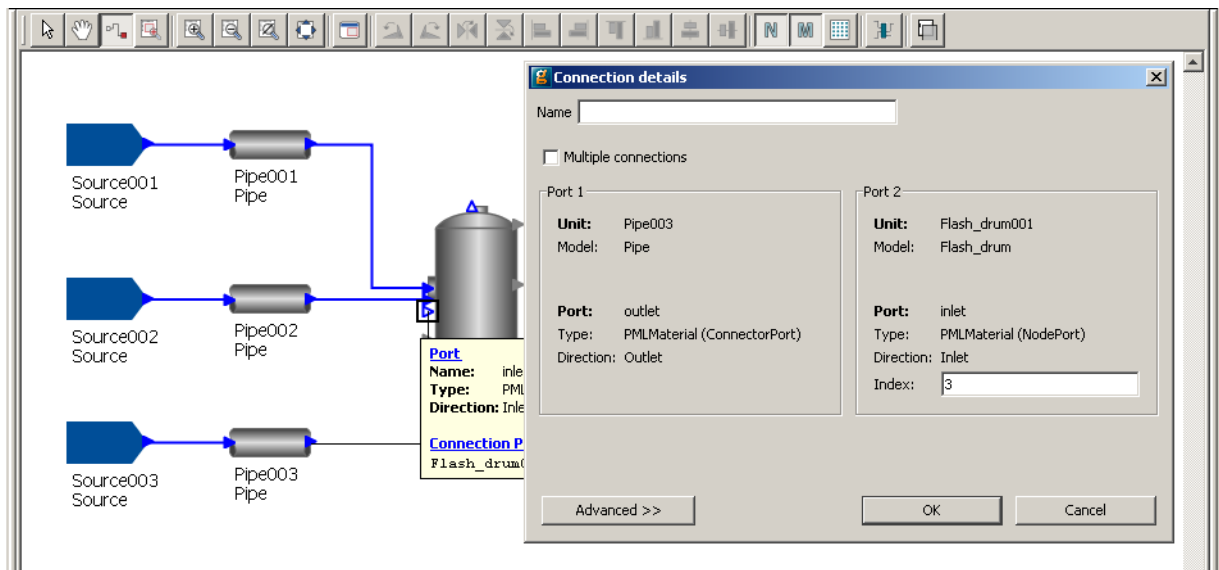
- Both Ports must be instances of the same Connection Type. Typically this will mean that they will carry the same type of information.
- The *Port categories* of the two Ports must permit the connection, as defined for the Ports' Connection Type (see Model Developer and Process Model Library Guides for further details).
- The directionality of the two Ports must be consistent:
  - An *inlet* Port can be connected to an *outlet* or a *bi-directional* Port;
  - An *outlet* Port can be connected to an *inlet* or a *bi-directional* Port;
  - A *bi-directional* Port can be connected to an *inlet*, *outlet* or another *bi-directional* Port.

To delete a connection: select  the connection and press **Delete**.

## Arrays of Ports

gPROMS supports Arrays of Ports. This functionality allows multiple streams to be connected to a single Port (see figure). An Array Port is indicated by a hollow Port icon.

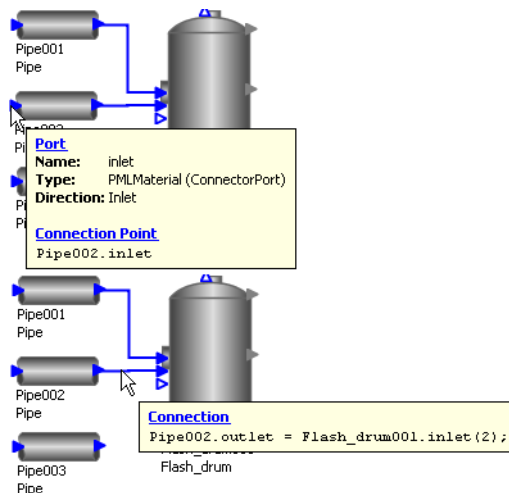
When connecting to (or from) an Array Port a Connection details dialog box prompts the user to confirm to which Port index the connection should be made. In most cases the user can simply accept the default index and press the OK button. However, the Port index can be modified if desired and multiple connections can be made in one action by checking the Multiple connections box and specifying the first and last index for the connections. Finally, a name can be given to the connection: this will be used to label the columns of stream tables and can also be used to find the connection using the shortcut menu.

**Figure 4.6. Array of Ports**

## Additional information display

The Topology editor provides additional information via a pop-up message approach. The following information is available:

- *Connection information*: placing the mouse pointer over a connection displays the corresponding gPROMS language for the connection equation.
- *Port information*: placing the mouse pointer over a Port displays the Port name, type and direction.

**Figure 4.7. Additional information display**

## Topology connections and the gPROMS language

Each Unit on the Topology corresponds to a declaration of a Model instance. The Unit declarations are shown (and can be edited) on the gPROMS language tab in the UNIT section.

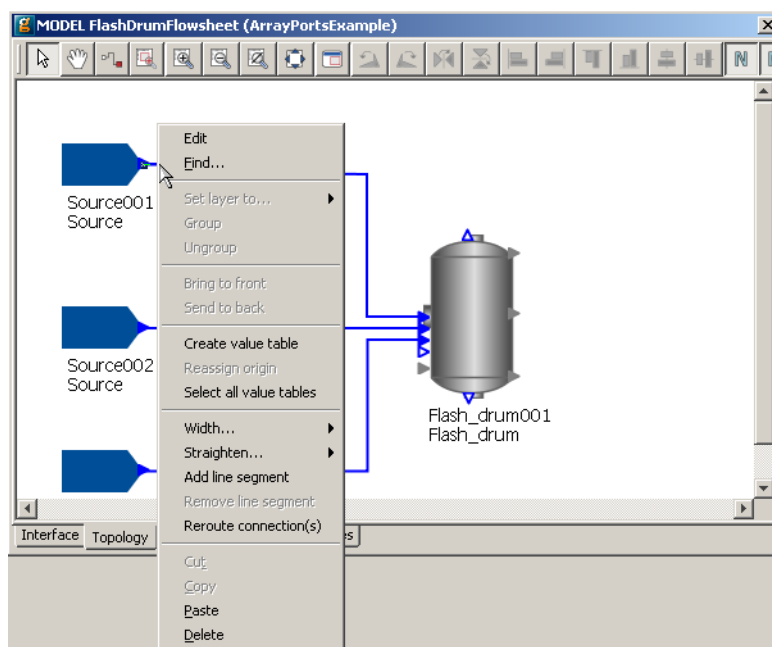
Each connection on the Topology corresponds to an equation that equates the two Ports<sup>3</sup>. Like the Unit declarations, the equations are shown (and can be edited) on the gPROMS language tab in the TOPOLOGY section.

<sup>3</sup>More explicitly, gPROMS expands the Port definition to equate all the variables in the two Ports.

## The connection shortcut menu

The appearance of connections on the Flowsheet can be customised using the Connection shortcut menu. As with the Unit short cut menu, the connection shortcut menu can be activated by left clicking on a connection to select it and then right clicking anywhere within the Flowsheets window.

**Figure 4.8. Connection shortcut menu**



The sections of the shortcut menu relevant to connections are:

- Edit

The edit command launches a dialog that allows you to edit the connection properties. For a connection that links two Models with scalar Ports, the only property that can be modified is the connection name. For more complicated connections, the Connection details dialog also allows the index of any array Ports to be modified.

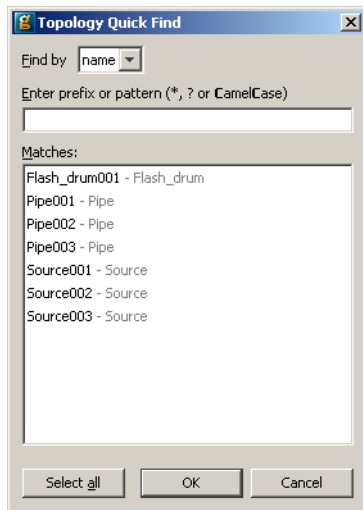
The connection name is subject to the same restrictions as other names in the gPROMS language (i.e. the same restrictions as names for Models, Parameters etc. — see Allowable entity names) is used to label and sort columns in stream tables.

- Find...

Selecting this option launches the Topology Quick Find Dialog, shown in the figure below. This dialog allows one to find (and select, if desired) a particular unit or connection by typing a search string into the box provided. All entities in the Topology whose name or type matches the search string will be listed at the bottom (if no string is entered then all entities are listed). The Find by listbox is used to specify whether the search lists the entity types or names matching the search string.

The search string may contain \* characters, which will match any string, or ? characters, which will match any single character. CamelCase allows you to match just the capital letters in an entity name: e.g., typing PS as the search string would match the unit names PipeSimple and PumpSimple.

Left clicking on one of the search results then selects that unit in the Topology. Multiple selections can be made by using the **CTRL** and **SHIFT** keys (individually or in combinations, as in Windows) or by pressing the Select all button to select all of the matching units.



- Connection properties

- Width...

By moving the mouse pointer over the Width... menu item, a secondary menu will appear that lets you select the width of the line used to draw the selected connection(s). Left clicking on one of the numbers will change the width of the line(s). The default connection width is 2.

- Straighten...

The Straighten... menu item is used to make the selected connection a straight line. gPROMS will need to move one of the two Units attached to the connection in order to make it straight. The secondary menu lets you choose which of the units to move. In the example above, the secondary menu contains Move unit on the left and Move unit on the right. For vertical connections, the menu items are Move unit at the top and Move unit at the bottom.

- Add line segment

When two Units are connected together, ModelBuilder uses an algorithm to determine the best line routing between them using a number of straight-line segments, typically 1 to 3. These arrangements can be overridden by adding or removing line segments. If you want to add more line segments to the connection, place the mouse pointer where you want the connection to be rerouted, right click to enable the shortcut menu and select Add line segment. ModelBuilder will then add extra line segments to route the connection via that position on the Flowsheet. The connection can be further customised by left-dragging the handles on the lines, in a similar manner to resizing objects.


This function may only be performed on a single connection, and is therefore greyed out if more than one connection is selected.

- Remove line segment

Connections between Units can also be simplified by removing line segments. Right click on the part of the selected connection that needs to be removed and select Remove line segment. ModelBuilder will then remove nearby line segments to give a more direct connection between the two Units.

This function may only be performed on a single connection, and is therefore greyed out if more than one connection is selected.

- Reroute connections

Selecting this option will perform an orthogonal routing algorithm on the selected connection(s). This is the same as pressing the  button on the Topology editor tool bar, but only applies to the selected connection(s).



- Delete

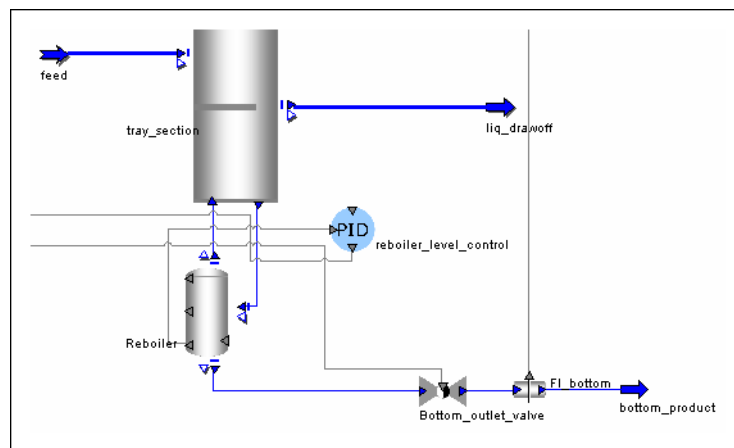
Of the commands in the standard Edit menu, only Delete has any relevance to connections. The Cut, Copy and Paste commands are all disabled. The selected connection(s) can be deleted either by left clicking on Delete or by pressing the **Delete** key.

## Hierarchical Model construction

Most of the Models in the gPROMS Process Model Library (PML) are component Models, i.e. non-composite Models. However, in gPROMS there is no limitation with respect to the number of levels in the hierarchical Model decomposition: composite Models with a defined topology can themselves be used to build even more complex composite Models (see for example the PML distillation column).

To use a composite Model on the Topology tab of a higher level Model the user must simply define the Interface [14] for the composite Model (i.e. introduce an icon and add Ports). The Ports of a composite Model appear on this Model's own Topology tab: these can be connected to any other valid Port on the topology as shown in the figure. Refer to the Model Developer Guide for further details.

**Figure 4.9. Hierarchical Model construction**

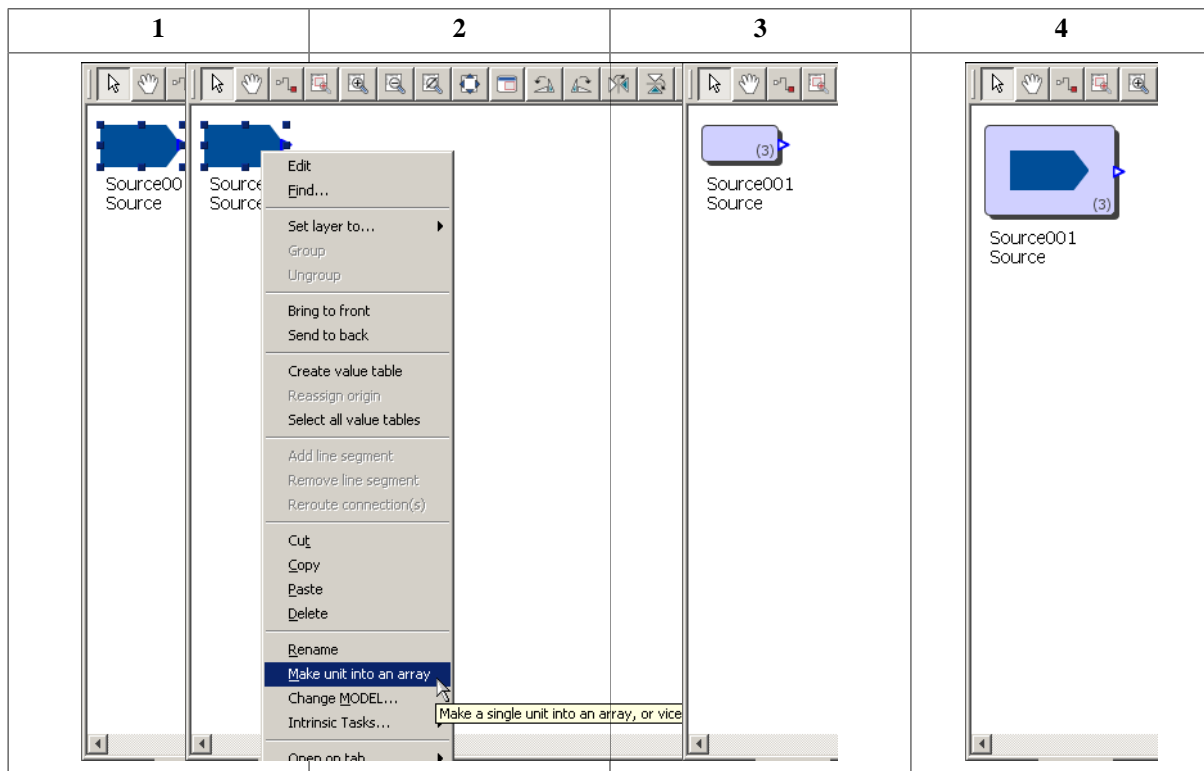


## Making a Unit into an Array

The Topology editor also supports Arrays of Units. Making a Unit into an array is easily done from the Unit shortcut menu:

- select Make Unit into an array in the popup menu,
- fill in the size of the array in the different dimensions separated by commas, and click **OK**.

Arrays of Units are distinguished on the topology editor by a coloured rectangle surrounding the Model icon, with the size of the array displayed in the bottom-left corner. Some small icons may not be visible at first, but resizing the rectangle will enable them to be shown. However, the Model type will always be shown below the rectangle (assuming this has not been disabled using the Topology Editor Toolbar). The procedure is shown in the figure below.

**Figure 4.10. Creating an Array of Units**

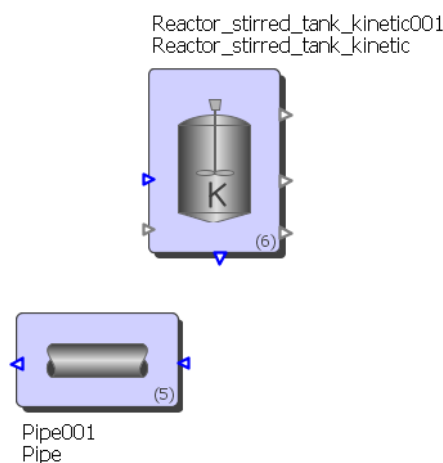
## Connecting Array of Units

Once a number of Arrays of Units are present in the Flowsheet, these will need to be connected to other Units or other Arrays of Units. It is also possible to connect an element of an Array to another element in the same Array.

It is easiest to describe the process of connecting Arrays of Units using an example. Here, we consider developing a Flowsheet comprising a series of CSTRs using Models from the Process Model Library. We wish to simulate a sequence of six CSTRs (using the `Reactor_stirred_tank_kinetic` Model), interconnected by Pipe Models. The Flowsheet will also need an inlet Source and an outlet Sink.

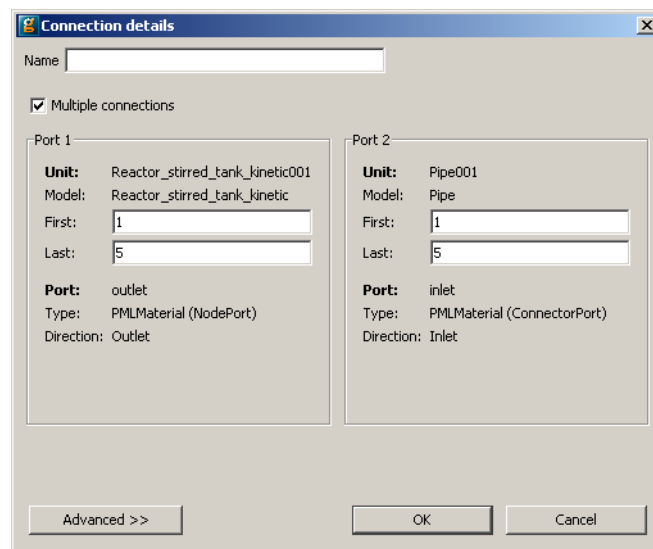
We begin by adding a Unit of a `Reactor_stirred_tank_kinetic` Model and a Unit of a Pipe Model and converting them to Arrays, as shown in the figure below.

**Figure 4.11. Flowsheet with unconnected Arrays of `Reactor_stirred_tank_kinetic` and Pipe**



Now we need to connect the outlet of the first CSTR to the inlet of the first Pipe, the outlet of the 2nd CSTR to the inlet of the 2nd Pipe and so on up to the 5th CSTR and Pipe. To do this, use the Connection tool, left click on the outlet of the CSTR Array and then left click on the inlet of the Pipe Array. This will activate the Connection details dialog, where the details of connection can be specified. gPROMS first assumes that a single connection will be made, and therefore asks for just one index for each Array, but we need to make an Array connection: in order to do this, the box labelled Multiple connections must be checked. Once checked, the dialog is updated and now the first and last index of each Array can be specified. Since we want the first CSTR outlet to go to the first Pipe inlet and so on up to the fifth element of each array, we just need to specify 1 as the first index and 5 as the last index for both Arrays. The correct specification is shown in the figure below.

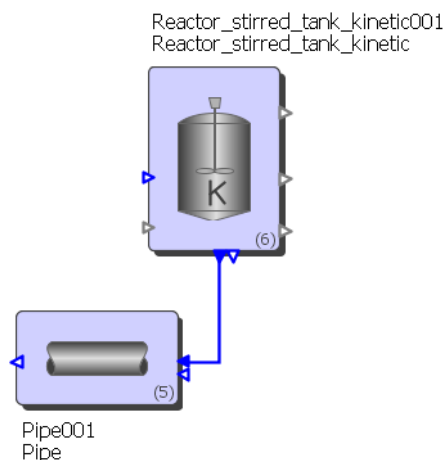
**Figure 4.12. Connection details dialog for the outlet of the CSTR Array to the inlet of the Pipe Array**



Note that gPROMS automatically sets the first index to 1 and the last index to the size of the Array. More often than not, this is what will be desired, but in this case we needed to change the 6 on the left to a 5 because there is no Model to which the last CSTR can connect.

Clicking on the Advanced >> button will change the dialog so that gPROMS language can be entered. However, this is not usually necessary and pressing the OK button will add the connections to the flowsheet as shown below.

**Figure 4.13. Flowsheet with outlet of CSTR Array connected to inlet of Pipe Array**



The gPROMS language associated with this connection is automatically generated and the whole model of the Flowsheet (in its present state) is shown below.

```
UNIT
    Reactor_stirred_tank_kinetic001 AS ARRAY (6) OF Reactor_stirred_tank_kinetic
    Pipe001 AS ARRAY (5) OF Pipe
SET
    # Start Dynamic Connections
    # End Dynamic Connections

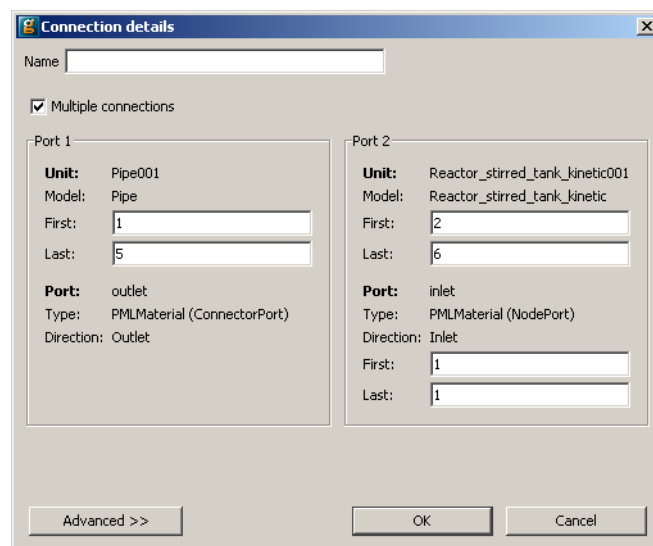
TOPOLOGY
    Reactor_stirred_tank_kinetic001(1:5).outlet = Pipe001(1:5).inlet;
```

The TOPOLOGY section in the language above defines exactly the connection we require. The outlet of the first CSTR being connected to the inlet of the first Pipe, and so on up to the fifth CSTR and Pipe. See Expressions involving arrays of Units for more details.

The next step is to connect the outlet of the Pipe Array to the inlet of the CSTR Array. This is done in more-or-less the same way: use the connection tool to connect the outlet port of the Pipe Array to the inlet port of the CSTR Array. When the Connection details dialog appears, check the Multiple connections box. This time, we need to connect the outlet of the first Pipe to the inlet of the second CSTR, and so on up to the 5th Pipe to the 6th CSTR.

The main difference this time is that the Reactor\_stirred\_tank\_kinetic Model has multiple inlets: therefore there is a second set of text boxes that allow you to specify to which inlets the connections should go. (For example, it could be that we wanted the outlet each Pipe to be connected to each of the inlets of just one of the CSTRs.) In this case, we want the outlet of each Pipe to go to the first inlet of each CSTR, so we must specify the first and last index of the CSTR inlets to be 1. The correct connection dialog is shown below, along with the gPROMS language that is automatically generated.

**Figure 4.14. Connection details dialog for the outlet of the Pipe Array to the inlet of the CSTR Array**



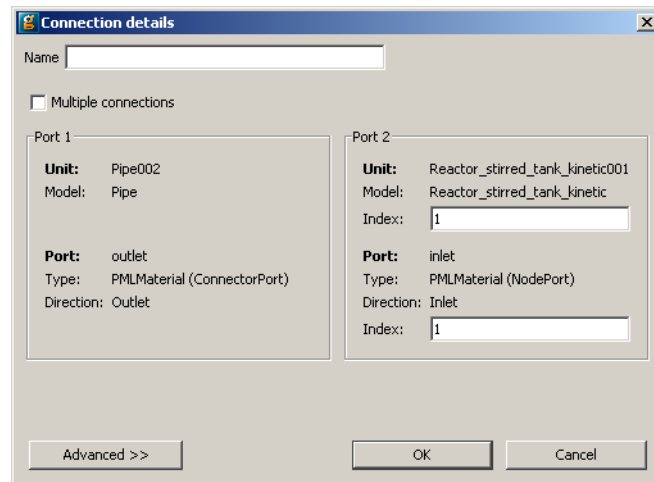
```
UNIT
    Reactor_stirred_tank_kinetic001 AS ARRAY (6) OF Reactor_stirred_tank_kinetic
    Pipe001 AS ARRAY (5) OF Pipe
SET
    # Start Dynamic Connections
    # End Dynamic Connections

TOPOLOGY
    Reactor_stirred_tank_kinetic001(1:5).outlet = Pipe001(1:5).inlet;
    Pipe001(1:5).outlet = Reactor_stirred_tank_kinetic001(2:6).inlet(1);
```

Now the CSTR train is complete, all that needs to be done to complete the flowsheet is to add the Source and connect it to the inlet of the first CSTR (using another Pipe) and similarly to connect the outlet of the last CSTR Sink.

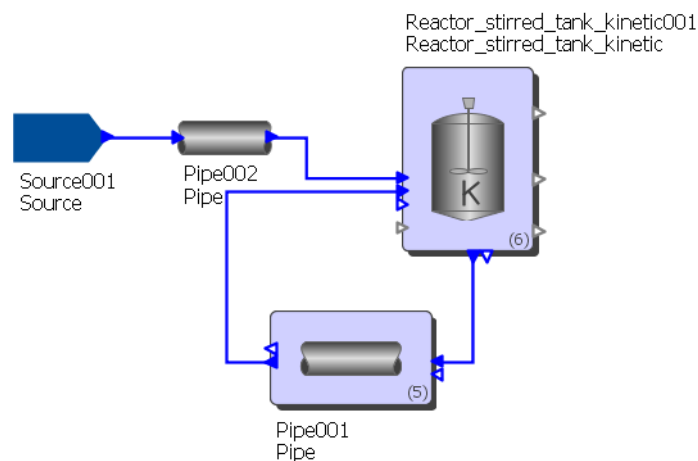
Adding the Source, the new Pipe (called Pipe002) and connecting them is straight forward. The outlet of the Pipe002 can be connected to the Array of CSTRs as before, but this time we only need a single connection so there is no need to check the Multiple connections box. Since we want to connect it to the first CSTR, we must make sure that the Unit Array index is set to 1 and the Model port index is also set to 1. The correct specification is shown below.

**Figure 4.15. Connection details dialog for the outlet of the inlet Pipe to the inlet of the CSTR Array**



Once the OK button is pressed, gPROMS creates the link and updates the gPROMS language, both of which are shown below.

**Figure 4.16. Flowsheet with Source connected to first CSTR of the Array**



```
UNIT
    Reactor_stirred_tank_kinetic001 AS ARRAY (6) OF Reactor_stirred_tank_kinetic
    Pipe001 AS ARRAY (5) OF Pipe
    Source001 AS Source
    Pipe002 AS Pipe

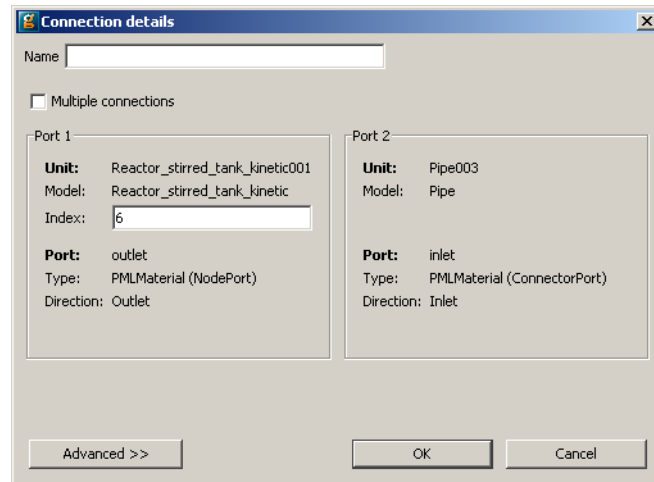
SET
# Start Dynamic Connections
# End Dynamic Connections
```

#### TOPOLOGY

```
Reactor_stirred_tank_kinetic001(1:5).outlet = Pipe001(1:5).inlet;
Pipe001(1:5).outlet = Reactor_stirred_tank_kinetic001(2:6).inlet(1);
Source001.outlet = Pipe002.inlet;
Pipe002.outlet = Reactor_stirred_tank_kinetic001(1).inlet(1);
```

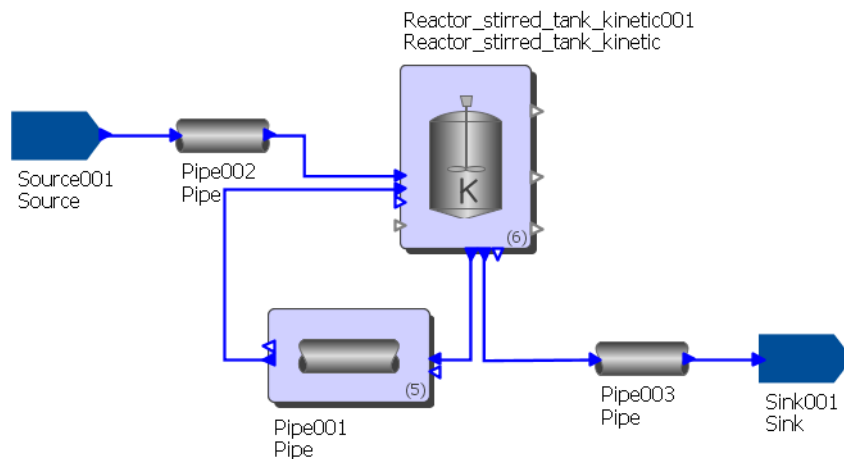
Finally, we need to add the last Pipe Unit, the Sink and connect them to the outlet of the last CSTR. This is done just as before. When connecting the outlet of the CSTR Array to the inlet of the Pipe (Pipe003), we just have to specify that it is the 6th CSTR's outlet that is connected to the Pipe, as shown below.

**Figure 4.17. Connection details dialog for the outlet of the CSTR Array to the inlet of the outlet Pipe**



The final flowsheet and gPROMS language are shown below.

**Figure 4.18. Final Flowsheet**



#### UNIT

```
Reactor_stirred_tank_kinetic001 AS ARRAY (6) OF Reactor_stirred_tank_kinetic
Pipe001 AS ARRAY (5) OF Pipe
Source001 AS Source
Pipe002 AS Pipe
Pipe003 AS Pipe
Sink001 AS Sink
```

#### SET

```
# Start Dynamic Connections
# End Dynamic Connections
```

## TOPOLOGY

```

Reactor_stirred_tank_kinetic001(1:5).outlet = Pipe001(1:5).inlet;
Pipe001(1:5).outlet = Reactor_stirred_tank_kinetic001(2:6).inlet(1);
Source001.outlet = Pipe002.inlet;
Pipe002.outlet = Reactor_stirred_tank_kinetic001(1).inlet(1);
Pipe003.outlet = Sink001.inlet;
Reactor_stirred_tank_kinetic001(6).outlet = Pipe003.inlet;

```

## Connecting Elements in the Same Array

The procedure for connecting elements in the same Array is identical to the one outlined previously for connecting different Arrays. Simply use the Connection tool to connect the outlet port of the Array to the inlet port of the same Array and then use the Connection details dialog to specify how the elements in the Array are to be connected.

Note that this is only possible with certain Models because some Models can only be connected via a *Flow Transportation* Model, cf. the CSTR example above, where the outlet of a *Reactor\_stirred\_tank\_kinetic* Model cannot be connected directly to the inlet of another *Reactor\_stirred\_tank\_kinetic* Model: they can only be connected using Models such as a Pipe, Pump or Valve.



## The Topology editor tool bar

The Topology editor provides many tools that enable the user to manipulate the look of the system Model. These tools can be accessed from the Topology editor tool bar that is shown in the figure below.

**Figure 4.19. The Topology editor toolbar**







## Display of Unit and Model names




- To toggle the display of the names of the Units in the flowsheet, click on the  button on the Topology editor tool bar. This will display or hide the names of the Units in the flowsheet.
- To toggle the display of the names of the generic Models underneath each Unit in the flowsheet, click on the  button on the Topology editor tool bar. This will display or hide the names of the Models underneath the Units in the flowsheet.

The font size of the Units and Model names is determined by the ModelBuilder preferences (accessed from the Edit menu).






## Zoom controls

- To pan the Model, click the  button from the Topology editor tool bar, select the Model and move the mouse to pan.
- To zoom in on a specific area of the flowsheet, click the  button from the Topology editor tool bar and draw a rectangle covering the area of interest: to do this, click, hold and drag a rectangle and then release the mouse button.
- To zoom in and out the flowsheet with respect to the centre of the Topology editor window, click the  or  buttons from the Topology editor tool.<sup>4</sup>


<sup>4</sup>Alternatively, to zoom in and out the flowsheet with respect to the centre of the Topology editor rotate the mouse wheel.

- To return the flowsheet to its original size, click the  button from the Topology editor tool bar.
- To fit the Model to the current topology editor view size, click the  button from the topology editor tool bar.
- To display a small "overview" window showing the whole Flowsheet, click the  button from the topology editor tool bar. (This button does not affect the main Topology Editor window.)


## Moving and rotating Units

- To move a Unit, or a set of Units: first select  the Unit, or the set of Units, then drag the selection with the mouse to a new location.
- To rotate a Unit, or a set of Units: select the Unit, or the set of Units, then click on the  or  buttons on the Topology editor tool bar.
- To flip a Unit, or a set of Units: select the Unit, or the set of Units, then click on the  or  buttons on the Topology editor tool bar to flip.
- To align a number of Units: select the Units then click on one of the following buttons on the Topology editor tool bar
  - to align the left-hand edges of the Units
  - to align the right-hand edges of the Units
  - to align the top edges of the Units
  - to align the bottom edges of the Units
  - to align the centres of the Units horizontally
  - to align the centres of the Units vertically


## Displaying a grid

- To display a grid over the Flowsheet, press the  button. Pressing it again removes the grid.

## Automatic layout routing

- To reroute all of the connections between all of the units, using an orthogonal routing algorithm, press the  button.

## Setting up Flowsheet layers

- Press the  button to activate the Layers dialog. Layers are explained in more detail later.

## Groups of Units

All of the tools for editing Topologies described so far have been concerned with modifying single Units. It is also possible to form groups of Units and apply many of the tools to the group as a whole.

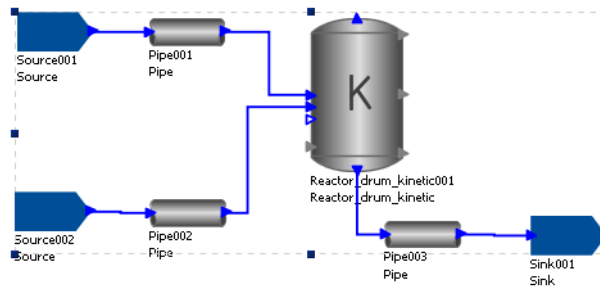
First, a group must be defined by holding down the **SHIFT** or **CTRL** key and left clicking on all of the Models that are to be included in the group. You may let go of the **SHIFT** or **CTRL** key at any time to scroll or resize the window and hold the key down again to continue selecting more Units. Once you have selected all of the Units



to be included in the group, right-click anywhere within the Topology window to activate the Short Cut menu and left-click on Group.

When you next select one of the Units within the group (having deselected the group by left-clicking on an object not included in the group or on the Topology background), the whole group will be selected. This is indicated by the a dashed box surrounding the group. As with individual Units, there are sizing handles on the corners and edges of the box. This is illustrated in the figure below, where all units are grouped together.

**Figure 4.20. A selected group**



A group may also contain another group (to any level of nesting). Simply create the group, as usual, by selecting all of the Units and existing groups that you want to include in the new group, right-clicking to activate the Unit short cut menu and clicking on Group.

To break up a group, simply select it, right-click to activate the Unit short cut menu and left-click on Ungroup. Any groups that were included in the top-level group will remain grouped; gPROMS remembers all of the sub-groups at all levels of nesting.

Once you have created all of the groups you like, you can manipulate them just as with individual Units. These include:

- Moving

To move the whole group, select it by left clicking on any Unit in the group, then left-click and drag any of the Units within the group to its desired location. The other Units will move by the same amount. You can combine these two actions in one by simple left-clicking and dragging one of the Units in one go (without releasing the left mouse button).

If the group is already selected, then simply left-click and drag one of its Units.

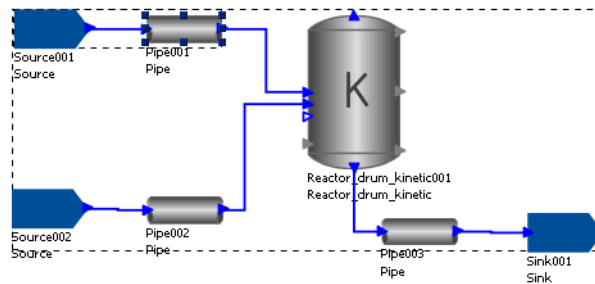
- Resizing

To resize the group, select it and then left drag one of the sizing handles in the corners or on the edges. Dragging a corner handle allows you to resize the height and width at the same time; dragging an edge handle changes the height or width only. All Units and groups within the group are resized by the same proportions.

If you hold down the **SHIFT** key, while resizing the aspect ratio will be set and maintained at 1 (i.e. a square).

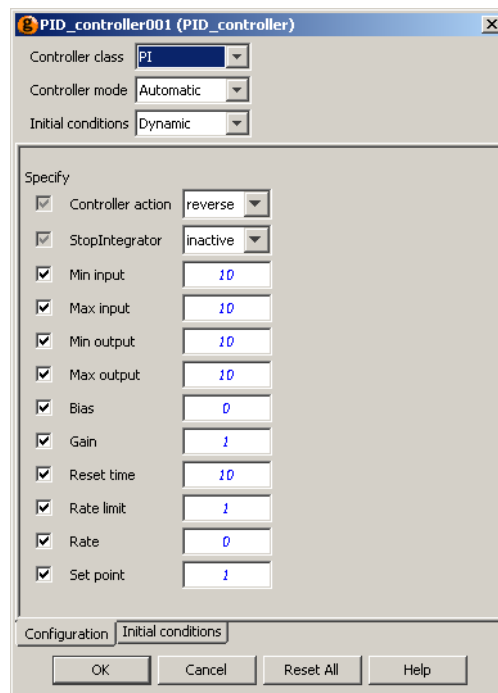
In addition to performing tasks on the group as a whole, you may still manipulate the individual objects within the group. First, select the group by left-clicking on one of its constituent objects. Then left-click on the object you want to manipulate. You may now move or resize this object relative to all of the others within the group, without affecting them. You may also perform any of the tasks allowed by the Short Cut menu, such as assigning the object to a Layer (as long as it is not a group itself).

Although there may be many layers of nested grouping, you may still select individual Units by repeatedly selecting groups down the hierarchy and then finally selecting an individual Unit. The example below illustrates the selection of a Unit belonging to a group that also belongs to a group. It requires 3 left-clicks to select this unit (assuming nothing is initially selected): first select the primary group, then the embedded group and finally the Unit. Note that the dashed boxes appear around both groups and the selected Unit is identified by the sizing handles.

**Figure 4.21. A selected Unit within a nested group structure**

## Making Model specifications

Specifications are made using Model specification dialogs that appear when double clicking a component Model icon on the Topology tab of a composite Model. Giving specifications using these dialogs corresponds to Assigning values to the gPROMS Parameters, Variables and Selectors declared as part of this component Model<sup>5</sup>.

**Figure 4.22. A dialog box of a PID controller**

Specification groups allow the Model user to select Model mode and choose from different specification sets. The PID controller example in the figure above contains two group selectors, determining the controller class [Proportional, Proportion Integral, Proportional Integral Derivative] and the controller mode [automatic, manual, cascade]. Changing the group may change the number of allowable specifications available to the user.

Specifications tabs can be used to segregate specifications logically. In the PID controller example, initial conditions have their own tab — these specifications only apply at the start of the simulation, whereas the configuration specifications apply throughout the time horizon. A dialog box may have 0, 1 or more tabs depending upon its design.

Specification properties that are mandatory have greyed-out check boxes (controller action in the example) whereas those that are optional have enabled check boxes. In many cases, the Model developer will have ensured the default number of specifications satisfy the degrees of freedom to ensure successful simulation.

<sup>5</sup>see the Model Developer Guide for more details on which gPROMS quantities need Assigning, how these specifications are classified and for construction and customisation of dialog boxes

Hovering the mouse over the specification labels will usually result in a tool tip provided by the Model developer appearing.

Specifications that involve array quantities appear as tables with the array dimension labelled vertically up the left hand side. Care must be taken to give the correct number of array specifications as they may not all fit in the window.

Specification bounds on certain specifications may have been imposed by the Model developer. If values are entered outside these bounds, the border turns red indicating that an illegal specification has been made. Hovering mouse pointer over the values themselves should display the bounds in another tool tip.

**Figure 4.23. Specification bounds violation**



Dialog box buttons:

- OK - confirms changes made to specifications
- Cancel - cancels any changes made
- Reset - returns all specifications to their default values
- Help - brings up the help for that Model

## Flowsheet layers

For complex flowsheets with many model instances, it may be desirable to maintain and print flowsheets at various levels of detail. For instance, one might like to have 3 flowsheets that show:

1. only the main processing operations
2. the main processing operations plus ancillary processing units
3. all entities, including controllers etc.

This may be particularly useful when developing Model-Library flowsheets, where connections between the key processing units are themselves Model entities, such as pipes etc.

Since the flowsheet determines the connectivity of the underlying gPROMS Model, there can only be one flowsheet defined but simplified versions of the flowsheet can be displayed and printed using *layers*. Another useful application of layers is a mechanism for assigning common attributes to a number of entities. In the case of gPROMS flowsheets, one can assign groups of similar Model entities to a layer and then set display properties for the whole group.

In the above example, we could create 3 layers and assign the main processing operations to the first layer, the ancillaries to the second layer and all other Model entities to the third. Then we can print or display flowsheets at the three levels of detail by specifying which layers to print or display.

A layered flowsheet can be developed in two ways:

1. develop the full flowsheet on a single layer, create additional layers and then assign Model entities to the new layers; or
2. create all of the layers first and assign Model entities to them as you develop the flowsheet.

Both of these methods make use of the same controls in ModelBuilder. These are:

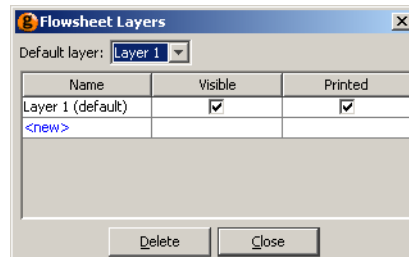
- Adding, renaming or deleting a layer
- Assigning a Model entity to a layer

- Setting the layer attributes

## Adding, renaming or deleting a layer

To add, rename or delete a layer, click on the Layers control in the Topology editor tool bar. This will activate a dialog, which shows the existing layers and their properties. For new projects, only one layer is defined:

**Figure 4.24. The Flowsheet Layers Dialog**



To add a new layer, simply left-click on the cell displaying the text <new>, type in a name for the layer and press the **Return** key (or selecting another row with the mouse has the same effect). The new layer will be created and automatically ordered alphabetically. Note that the names of the layers are case sensitive.

To rename an existing layer, simply left-click on the name of the layer and type in the new name, pressing **Return** as before. This will replace the existing name with the text that you type in. If you only want to change part of the name, then double left-click on the name to enable edit mode. You can now use the keyboard and mouse to edit the name. Again, pressing **Return** or selecting another layer saves the changes made to the name.

To delete a layer, simply select the layer by clicking on any cell in its row and then press the Delete button. You cannot delete a layer if any entities are assigned to it. The easiest way to identify which entities are assigned to a layer is to check and un-check the Visible property (see Specifying layer attributes) and observe which entities change on the Flowsheet. You can then re-assign these entities to other layers (as described next) and delete the layer. (You need not close the Layers dialog while re-assigning entities.)

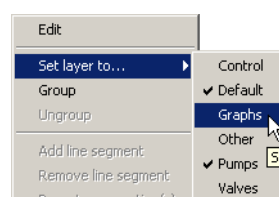
Once you have set up all the layers you require, you may specify the layer to which new entities added to the Flowsheet will be assigned. Left click on the Default layer control and select the name of the layer you want to be made default.

## Assigning a Model entity to a layer

To assign an entity to a layer, select the entity by left clicking on it and then right click anywhere on the Flowsheet to bring up the Unit shortcut menu. Now hold the mouse pointer over the Set layer to... menu item. This will pop up a list of available layers, with a tick next to the layer currently containing the entity. Simply left click on another layer to assign the entity to this new layer.

Multiple entities can be assigned to a layer in one go by selecting all of them and then right clicking to enable the Unit shortcut menu. Multiple entities can be selected by left clicking on a single entity, holding down the Shift key and then left clicking on one or more other entities while the Shift key is held down. Now all of the selected entities can be assigned to the same layer by right clicking and continuing as for a single entity. Note that there will be a tick next to each layer that contains at least one of the entities selected. Once you left click on a layer name, they will all be reassigned to that layer.

**Figure 4.25. Assigning a group of entites to a layer**



## Specifying layer attributes

Once all of the entities in a Flowsheet have been assigned to different layers, you can specify what parts of the Flowsheet are displayed or printed by setting the attributes of each layer. To do this, bring up the Layers dialog by left clicking on the Layers control in the Topology editor tool bar.

In addition to adding, renaming and deleting layers, the Layers dialog allows you to set the properties of the layer. These are:

- Visible — defines whether the layer will be shown in the Flowsheet editor
- Printed — defines whether the layer will be shown when the Flowsheet is printed

Each layer has a checkbox for each property. By default, both properties are enabled for new layers. Simply left click on a checkbox to toggle its value. Any time you want to display or print the Flowsheet with a different level of detail, simply return to the Layers dialog and change the properties of the layers you want to show or hide.

These properties apply both to the actual Model Flowsheet (i.e. the one in the gPROMS Project) and to all instances of the Flowsheet in a gPROMS Case.

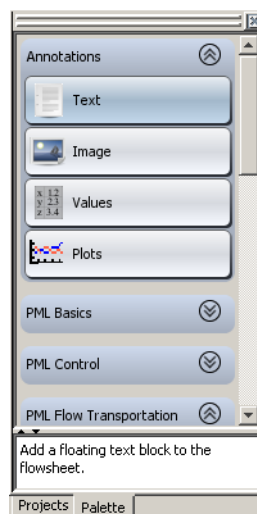
## Adding graphs and other annotations to the flowsheet

In addition to the Model entities and their connections, a number of other annotations may be added to the flowsheet, some of which are updated dynamically as gPROMS performs a simulation. These are:

- Text — a static text label
- Images — a static image
- Values — a dynamic value or table of values of Model Variables
- Plots — a dynamic plot of Model Variables

To insert an annotation into the Flowsheet, left click on the Palette tab in the left-hand pane of ModelBuilder. You will now see a list of projects from which you can select Models to add to the Flowsheet. At the top will be a section named Annotations. Click on the triangle button to expand this section, which will result in a window similar to that shown below.

**Figure 4.26. The Annotations Palette**



To add an annotation to the Flowsheet, simply left drag one of the items in the annotations palette onto the Flowsheet. You can then move and resize it as with ordinary Model entities. Furthermore, annotations can be

assigned to layers just as Model entities. This way, for example, you can set up the Flowsheet to display real-time results while the simulation is running (they can also be replayed after a simulation) but none of the results would appear in the printed Flowsheet.

More details on the individual annotations can be found by clicking on the links above. How gPROMS ModelBuilder presents real-time results on the Flowsheet is discussed in Viewing results on the Flowsheet during and after simulation.

## Viewing results on the Flowsheet during and after simulation

Having added value or plot annotations to a Flowsheet, and specified which variables they are to display (see value annotations), these values will be displayed automatically during a gPROMS simulation or can be played back after the simulation has been completed.

To see the results, open a Case Project, open the Trajectories branch and double click on a Model entity that contains a Flowsheet with value or plot annotations. If the Case is currently being simulated by gPROMS, then you will automatically see the values and graphs being updated dynamically as the simulation progresses. If the simulation has ended, the values shown will correspond to the end of the simulation.

Above the standard Topology editor tool bar, will be a results-playback toolbar that enables the simulation to be reviewed in a number of ways. This toolbar is shown below.

**Figure 4.27. The Playback Toolbar**







The toolbar comprises a text box, a slider and four buttons: to stop the playback and rewind to the start, to stop the playback at the current time, to begin playing back from the current time and to speed up the playback.

To set the values displayed to a specific time, left click in the text box and enter the desired time. Playback will stop as soon as you click on the text box. Press **Return** to set the displayed values to the time entered. Note that because gPROMS only stores values at regular intervals (set by the ReportingInterval SolutionParameter) or whenever there is a discontinuity (e.g. because of a RESET Task), gPROMS can only show values at the closest time to the one entered in the text box. The text box will be updated automatically to show the actual time used. Of course, if the simulation is still running, you cannot enter a time later than the current simulation time.

The time displayed can also be set by dragging the slider bar. Left click on the button and, holding down the left mouse button, drag it to a time for which you want the values shown. The time in the text box will update as you do this. Alternatively, you can left click on the slider on either side of the button, which will cause the time to move forward or backward a certain amount.

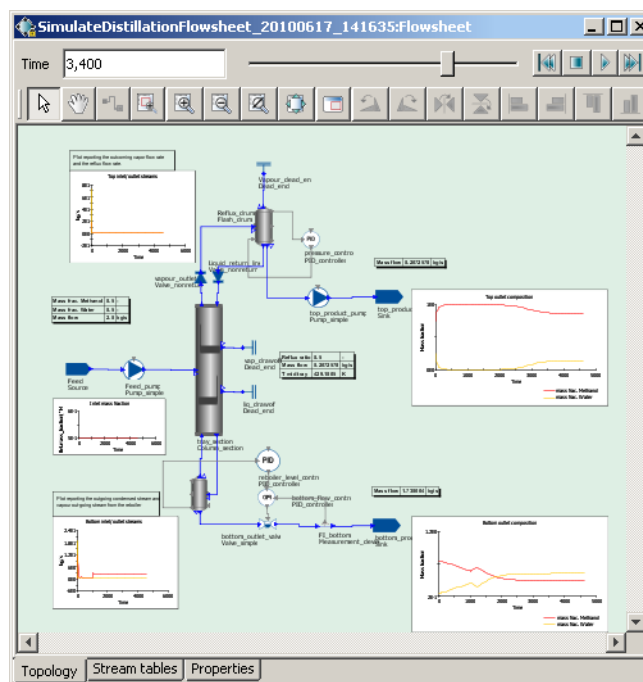
The next four buttons control the playback of the results.

- The  button will stop any playback and return the time to zero.
- The  button will stop any playback and return the time to zero. The second button will stop the playback, but the time will remain where it was when the button was pressed.
- The  button has three functions
  - To begin playback, if playback is currently stopped
  - To stop playback during normal playback (the time remains where it was)
  - To return to normal playback if playback is at a higher speed
- The  button increases the playback speed each time it is pressed (playback is reset to normal every time the playback is stopped or paused)

- If playback is stopped, pressing the fast-forward button begins playback at normal speed
- If the results are being played, the fast-forward button increases the speed by one level every time it is pressed
- If the results are showing the current values during an active simulation, playback naturally cannot be increased.

An example of a flowsheet with value and plot annotations is shown below.

**Figure 4.28. Flowsheet results with annotations and playback toolbar**



## Text annotations

Once a text annotation has been added to the Flowsheet, it can be moved and resized in the same way as Model entities. Left click on the text annotation to select it and you will see 8 control points. You may then:

- Hold down the left mouse button anywhere within the control points and drag the mouse to move the text annotation.
- Left click on one of the control points and drag it to resize the text annotation. (The mouse pointer will change when it is over a control point to indicate which way the point can be moved: vertically only, horizontally only or in any direction.)
- If you hold down the **Shift** key while resizing, the aspect ratio of the box will be set to 1 (i.e. a square).

Text annotations have a number of attributes relating to the text itself and the surrounding box. These are accessed by double clicking on the text annotation, which enables an editor dialog. This contains two tabs:

- The Editor tab: for setting the attributes of the text itself
- The Properties tab: for setting the attributes of the surrounding box

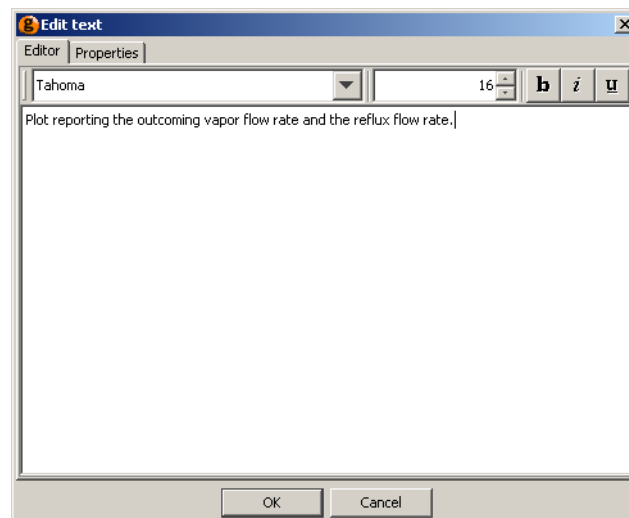
## The Editor tab

The Editor tab is divided into two sections: a set of controls at the top and the main text control at the bottom. Simply enter the desired text in the text control. Now you can use the mouse or keyboard to select parts of the text and apply attributes to them using the controls in the toolbar at the top of the pane. These are:

- Font name
- Font size
- Boldface
- Italics
- Underline

The toolbar can be undocked from the Editor dialog by dragging the *handle* just to the left of the Font combo box. To redock the toolbar, left click on the "X" button to close it, and it will return to the main Editor dialog.

**Figure 4.29. The Editor tab**



## The Properties tab

The Properties tab contains one main control and up to 5 auxilliary controls. The main Style control selects the type of line used to draw the surrounding box:

- None

Selecting None prevents a box being drawn around the text. When this is selected, no other controls are present in the tab.

- Drop Shadow

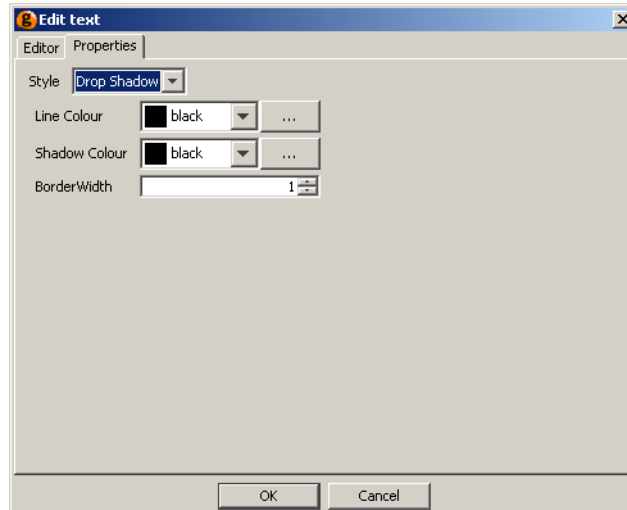
The Drop Shadow style draws a box around the text with a shadow. Five additional controls are enabled:

- A colour selector for the main box, containing some basic preset colours
  - An advanced colour picker for the main box, which allows you to pick colours from a much larger palette or to define custom colours in RGB or HSV format
  - A colour selector for the shadow box, containing some basic preset colours
  - An advanced colour picker for the shadow box, which allows you to pick colours from a much large palette or to define custom colours in RGB or HSV format
  - A border width selector, which allows you to specify the line width used for both the main box and its shadow
- Bevel



The bevel style is a single box using two line widths and two colours to give an embossed look. The line widths are fixed, with the top and left sides being slightly wider than the bottom and right, so there are only four additional controls. They are the same as the drop shadow style: two controls for selecting the colour of the top and left sides, and two for the other sides.

**Figure 4.30. The Properties tab**



## Image annotations

When an image annotation is added to the Flowsheet, it can be moved and resized just as a text annotation. However, when the image file is selected, the annotation is automatically resized to the size of the image. Therefore, it is best to select the image file first. To do this, double click on the image annotation to bring up the file-selector dialog. You can then select the desired image using the standard file browser. Supported formats are:

- .svg
- .gif
- .jpeg
- .jpg
- .png

Once the image has been selected, you can move or resize it by left clicking on the image annotation to select it and then:

- Holding down the left mouse button anywhere within the control points and dragging the mouse to move the image annotation.
- Left clicking on one of the control points and dragging it to resize the image annotation. (The mouse pointer will change when it is over a control point to indicate which way the point can be moved: vertically only, horizontally only or in any direction.)
- If you hold down the **Shift** key while resizing, the aspect ratio of the image will be set to 1 (i.e. a square).

## Value annotations

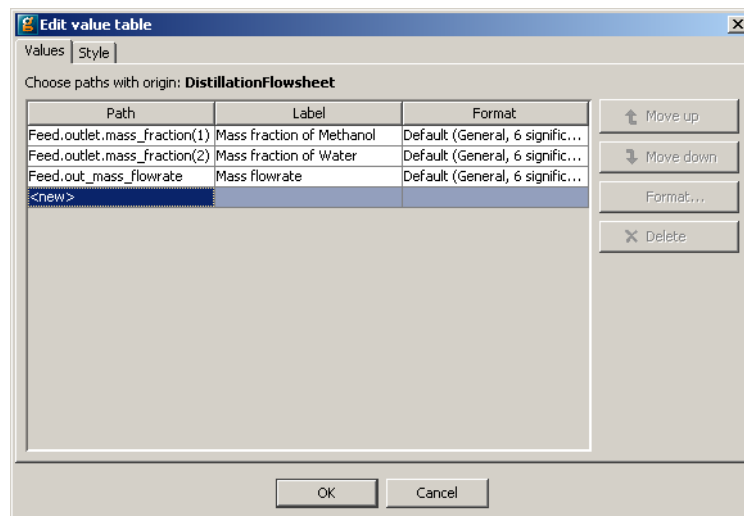
Tables of Variable values can be added to the Flowsheet in a similar way to text and image annotations. They will be updated dynamically as the simulation runs, or as a completed simulation (a Case Project) is replayed

(see Viewing results on the Flowsheet during and after simulation). Three types of Value Table can be added to a Topology Flowsheet:

- a generic Value Table, which displays the Variables from any Unit in the Topology;
- a Unit Value Table, which is specific to a particular Unit in the flowsheet; and
- a Connection Value Table, which displays all of the Variables associated with a Connection.

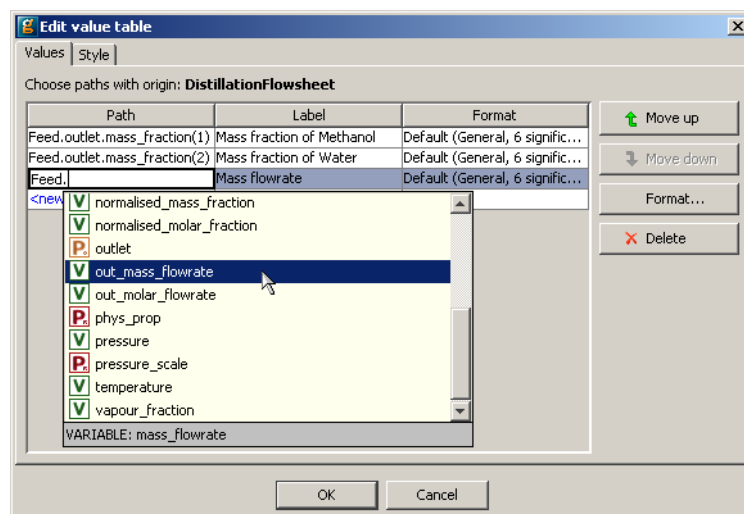
To create a generic Value Table, drag the Values annotation from the palette onto the Topology. An empty table will appear, containing the text <double-click to add variable>. Once you double click on the table, the following dialog will appear, which will allow you to choose the variables to display and configure the format of the table.

**Figure 4.31. The Value Annotation Dialog**



Adding variables to be displayed is very easy. Simply click or double click on the <new> text under the Path column and enter the full path of the Variable required. Here, ModelBuilder will display all of the available completions when you press **Ctrl+Space**, just as when using the gPROMS Language editor (see figure below and Assisted Pathname Completion).

**Figure 4.32. Path Completion in Value Annotation Dialog**



When the Variable path is complete, press **Return** or **Tab**. The Label for this variable will automatically be populated with the path name. To replace this with your own label, left click in the cell and type the new label; to edit the existing one, double click in the cell. Press **Return** or **Tab** to confirm the changes made.

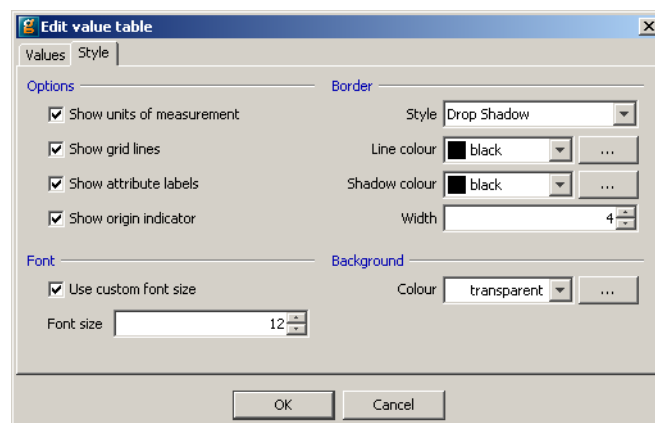
To add further Variables, repeat the above procedure using the newly created row with <new> in the Path field.

Each value can be displayed using a variety of number formats. To set the number format for a single Variable, double click on the number format for that Variable and select the desired number format from the dialog that appears. The example below the listbox indicates how the number will appear in the table. The available number formats are:

Number Format	Meaning	Examples
Default	Whatever value is specified as the default number format in the Model Builder preferences. When the default number format is changed, the numbers displayed on the topology will be automatically updated.	N/A (depends on value set)
General	The general number format is a flexible format that changes depending on the value of the number to be displayed. Normally it displays the specified significant figures without an exponent. However, an exponent is used if the exponent from its conversion is less than -4 or greater than or equal to the number of significant figures. The number of significant figures can be set from 1 to 6.	1.234E-06, 12.34, 1.234E+09 (4 s.f.)
Fixed	The fixed number format renders numbers using a fixed number of decimal places (and no exponent). The number of decimal places can be set from 0 to 9.	0.000001, 12.340000, 1234000000.000000 (6 d.p.)
Scientific	The scientific number format always renders numbers using the scientific notation with the number of decimal places specified. The number of decimal places can be set from 0 to 5.	1.2340E-06, 12.3400E+00, 1.2340E+09 (4 d.p.)

Once you have selected all of the Variables you want to display, you can then configure the Value Table by pressing on the Style tab. This displays the style options, as shown below.

**Figure 4.33. Style Options in Value Annotation Dialog**



The following options are available:

- Options
  - Show units of measurement — enables or disables the display of units column after the values
  - Show grid lines — enables or disables lines between the rows and columns of the table
  - Show attribute labels — specifies whether or not the label for each value is shown
  - Show origin indicator — HELP! I can't see what this does!
- Border
  - Style — sets the style of the border of the table, which can be None, Drop Shadow or Bevel
  - Line colour — specifies the colour of the border line (Drop Shadow or Bevel)
  - Shadow colour — specifies the colour of the shadow (Drop Shadow only)
  - Width — specifies the width of the shadow (Drop Shadow only)
- Font
  - Use custom font size — specifies a specific font size for the text in the table (when the check box is enabled, an extra control will appear to allow you to specify the font size)
- Background
  - Colour — specifies the colour of the table background

Press the OK button to confirm all of the changes made since the last time the dialog was activated, or Cancel to abandon them.

To add a Unit Value Table, right click on a Unit and select Create value table from the context menu. The rest of the procedure is identical to generic Value Tables, except that you can only choose Variables from the Unit first chosen. To indicate that the Value Table is a Unit value table, a dotted line will join the table to the Unit whose Variables will be displayed in the table.

To add a Connection Value Table, right click on a Connection and select Create value table from the context menu. The rest of the procedure is identical to creating Unit Value Tables, except that the table will be automatically populated with the Connection Variables and that when editing the table, only Variables belonging to that connection can be chosen.

## Plot annotations

A dynamic graph or one or more Variables may be added to a Flowsheet by dragging a plot annotation from the annotations palette onto the Flowsheet. The annotation can be moved and resized exactly as with value annotations.

After placing and sizing the plot annotation, double click on it to activate the annotation dialog so that you can add or change what Variables are plotted.

**Figure 4.34. The Plot Annotation Dialog**

Select values to display

Plot Title: Top inlet/outlet streams

X-Axis Label: Time

Y-Axis Label: kg/s

Show Legend: ☐

Show Symbols: ☐

Path	Label
tray_section.out_vap.mass_flowrate	Out vap. mass flow
tray_section.in_liq.mass_flowrate	In liq. mass flow
<new>	

Delete

OK Cancel

Specifying Variables and their labels is done in exactly the same way as value annotations.

The additional controls on the dialog allow you to specify a plot title, y-axis label and whether or not a legend is shown. Simply left click in the text boxes to add the title and y-axis text; left click on the checkbox control to show the legend (or disable it if it is already checked).

---

# Chapter 5. Executing simulations

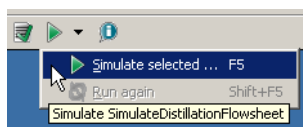
In this section we consider running simulations. For details on running other model-based activities, such as optimisation and parameter estimation, and their particulars - please refer to the appropriate guide.

## To execute a simulation

Select the Process<sup>1</sup> to be executed on the project tree and any of the following procedures can be used to start the simulation:

- navigate to the Activities drop down menu and select Simulate (keyboard shortcut - **F5**).
- press the green simulate arrow on the toolbar pane running across the top of the main ModelBuilder window.
- Right click on the process and select Simulate from the submenu.

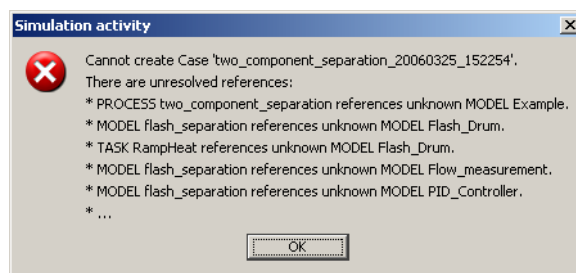
**Figure 5.1. Simulating: executing a process**



## Cross-reference check

Before attempting to execute an Activity gPROMS checks that all Entities referenced from other Entities are either present in the current Project or from a cross referenced Project. gPROMS will notify you if you have made a mistake and a referenced Entity cannot be located, for example, by referring to a Model that does not exist within a Process.

**Figure 5.2. Automatic cross-reference checking using gPROMS**



## Cases

A Case is a combined record of all the input information that defines a Model-based activity and the results generated by the execution of this activity, as well as any diagnostic messages that may have been issued during its execution. The intention is that a Case may serve as a permanent record of a particular Model-based activity that can be archived for future reference, thus providing auditability and traceability of Model-based decisions.

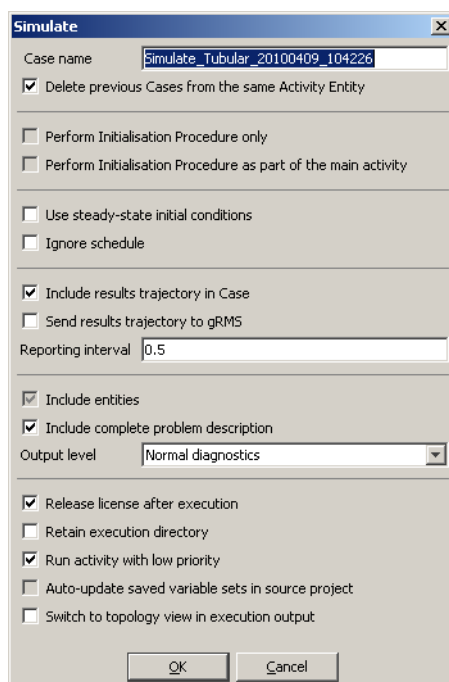
## The Case configuration and execution control dialog

A Case is created automatically by ModelBuilder at the start of the execution of any Model-based activity. An execution control dialog is presented to allow the user to configure various aspects of the Case including the following:

---

<sup>1</sup>Alternatively a model may be selected to run the simulation - this will execute a process with the same name; if one does exist, it will be created

**Figure 5.3. The execution control dialog**



- Case configuration
  - *Case name*: ModelBuilder creates a default name for the Case. This comprises the name of the Process Entity defining the simulation followed by the date and the time, the three parts being separated by underscore characters. The user may overwrite this name by editing the corresponding field in the Case dialog.
  - *Delete previous cases from the same Activity Entity*: during Model development, one typically executes the same Model-based activity (based on the same Process Entity) time and time again. In such situations, it is usually both unnecessary and undesirable for ModelBuilder to keep the Case corresponding to each and every execution. By enabling this check-box, the user instructs ModelBuilder to retain only the last Case arising from any particular Process Entity.
- Initialisation Procedures
 

Initialisation Procedures provide a set of initial guesses for the initialisation of a Process. There are two ways to do this in gPROMS:

  - *Perform Initialisation Procedure only*: This method only performs the Initialisation Procedure to generate the initial guesses but does not continue to initialise the problem or execute the Schedule. This is useful if you only want to generate the initial guesses and save them in a *Saved Variable Set* . (The generation of the *Saved Variable Set* is specified as part of the Initialisation Procedure.)
  - *Perform Initialisation Procedure as part of the main activity*: This method generates the initial guesses using the Initialisation Procedure, then immediately performs an initialisation using them. If a Schedule is present, and the *Ignore schedule* option is unchecked, then gPROMS will complete the simulation according to what is specified in the Schedule. If the Initialisation Procedure specifies that a *Saved Variable Set* should be generated, then this is done too.
- Dynamic or steady-state simulation
  - *Use steady-state initial conditions*: When this is selected, the initial conditions specified in the Process will be ignored and the Flowsheet will be initialised at steady-state.
  - *Ignore schedule*: When selected, the Schedule which is specified in the Process will be ignored and only an initialisation will be performed. The initialisation is either steady-state or dynamic as specified by the user.

- Results configuration
  - *Include results trajectory in Case*: incorporate results generated from the simulation natively into the Case in the form of tables and graphs for individual quantities, Stream Tables and Model Reports.
  - *Send results trajectory to gRMS*: send simulation results to the gPROMS Results Management System for more advanced display capability.
  - *Reporting interval*: the interval at which the values of time-varying Model quantities are reported.
- Output configuration
  - *Include entities*: copy each and every Entity that has been used for defining this Model-based activity, including any Entities that reside in cross referenced Project (e.g. libraries).
  - *Include complete problem description*: contain a complete definition of the activity in the gPROMS language comprising of the above Entities in the correct reference order - the first line of each Entity is annotated with a comment naming the Project from which the Entity was obtained.
  - *Output level*: This parameter sets the overall diagnostics level when performing an Activity. Its (integral) value may range from -1 to 9, which are described as Silent, Solver diagnostics only, Normal diagnostics, Extra - level 2 and so on up to Extra - level 9. Currently, levels 2 to 9 behave identically to level 1 but may may be used in the future to give more control of the diagnostic level. When the Execute Control dialog is launched, it reads the value of the `OutputLevel` Solution Parameter (if present) from the Process entity in order to initialise the value in the dialog. If the `OutputLevel` is not specified in the Process entity, then the default value is Normal diagnostics. If this value is changed in the dialog, then the execution is performed using the currently selected value *without* modifying the value specified in the Process. The effect of this parameter on the output of a simulation activity is summarised in the table below.

`OutputLevel` is also described in Controlling result generation and destination.
- Miscellaneous execution controls
  - *Release model after execution*: whether the license required by the simulation should be retained at the end of the execution<sup>2</sup>.
  - *Retain execution directory*: keep the temporary folder in which the "execution files" are stored.
  - *Run activity with low priority*: set the execution as a low priority activity within Windows.
  - *Auto-update source project*: whether any Entities that are used by the activity and which are modified by its execution should automatically be updated at the end of the execution<sup>3</sup>.
  - *Switch to topology view in execution output*: once the simulation experiment is completed automatically switch the view in execution output from the Output view to the Topology view (if one is defined).

The user may specify that one or more of the items above may be omitted from the Case by un-checking the corresponding check boxes.

Finally, the execution control dialog provides a Cancel button that allows the user not to go ahead with the execution of the simulation. On the other hand, pressing the OK button instructs the ModelBuilder to proceed with the execution.

---

<sup>2</sup>Retaining the license allows some interaction with the Model at the end of the execution. The license can always be released manually at any time.

<sup>3</sup>This is particularly useful for gPROMS saved variable sets that may be used to provide initial guesses for the initialisation of the activity, and which are subsequently over-written by values obtained as a result of the execution of the activity.



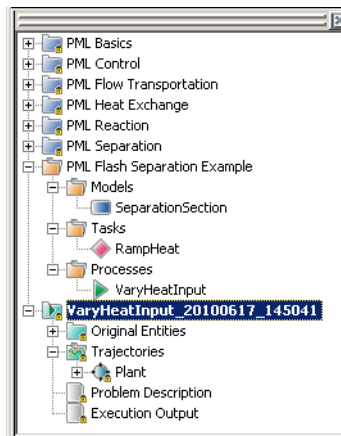
**Table 5.1. Effects of Output level on execution diagnostics**

Output Level	-1 (Silent)	0 (Solver diagnostics only)	$\geq 1$ (Normal diagnostics, Extra – level n)
Diagnostics for system construction, index reduction and structural info, schedule execution, STN switching etc.	Off	Off	On
Diagnostics of individual solvers	Off	On — according to the individual solver settings	On — according to the individual solver settings

## Management of Cases

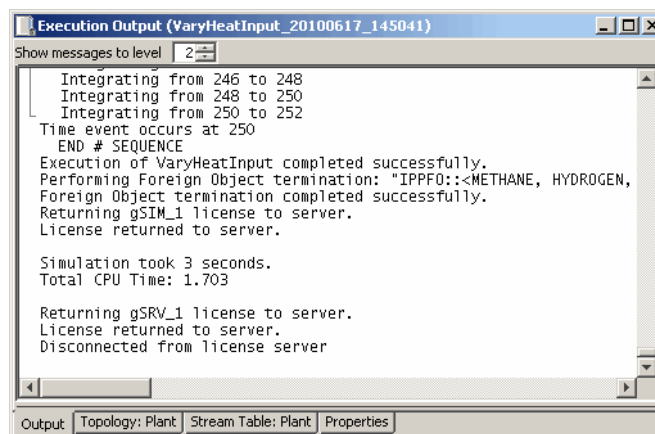
Once the user presses the OK button in the execution control dialog, ModelBuilder creates the Case. Just like a Project, a Case appears as a sub-tree of ModelBuilder's navigation tree. However, unlike most Projects, all entries in a Case are read-only, and this is indicated by a lock symbol annotating each entry in the Case sub-tree.

**Figure 5.4. Read-only Cases**



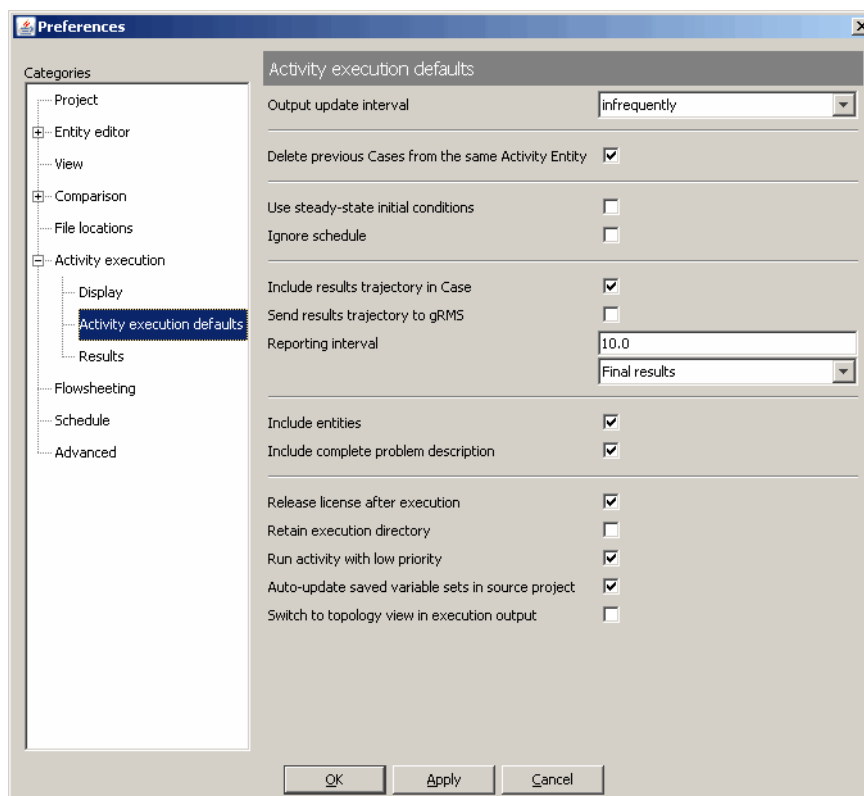
Cases may be saved to disk using standard Save and “Save as...” mechanisms from ModelBuilder's Project menu. The files used for their permanent storage have the .gCS suffix and are marked as read only. Cases may also be closed (with or without saving) at any time, and also be opened at a later stage using standard Project-management mechanisms.

**Figure 5.5. Cases and activity execution**



The default choices for the entries in the execution control dialog can be specified as part of the user's ModelBuilder Preferences. Further configuration options can be accessed by highlighting the “Case content defaults” entry.

**Figure 5.6. ModelBuilder preference dialogs relating to Case management and activity execution**

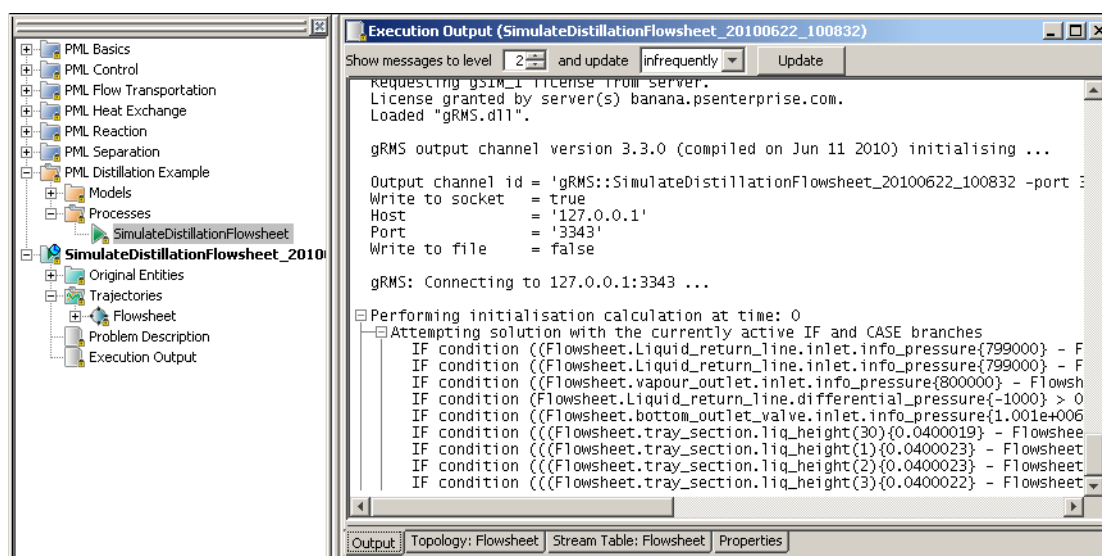


## Creating Projects from Cases

Cases are read-only records of executed Activities and therefore can not be changed or used directly to perform Activities again. They can, however, be converted into Projects by performing a right-click on the Case and selecting Create gPROMS project from the popup-menu. When this is selected, a new Project is created and all Entities which have been saved in the Case are copied into the new Project. It is therefore required that the option *Include entities* is selected in the Execution Dialog when executing an activity.

## Interacting with executing simulations

The execution of an activity is initiated by the user pressing the OK button on the execution control dialog. At this point the ModelBuilder creates the Case. It also requests and obtains the license necessary for execution; this license is associated with the Case until it is released either automatically at the end of the execution or manually at a later stage if the user has disabled this automatic license release mechanism. The fact that the Case is holding (“tying down”) a license is indicated by a “clock” symbol attached to the Case's name in the ModelBuilder's navigation tree. The symbol automatically disappears as soon as the license is released.

**Figure 5.7. Cases and activity execution**


## Execution output

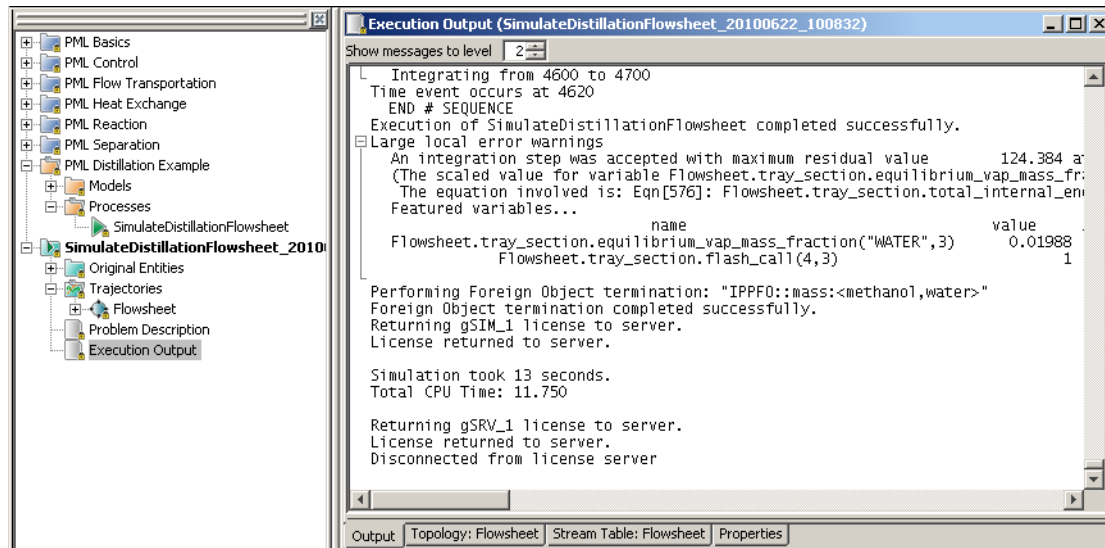
Once the necessary license is obtained, ModelBuilder creates an execution output window which displays all the messages relating to the solution of the Model-based activity. During this execution, the user can interact with the executing activity by right clicking on this window, which causes the execution interaction menu to appear. While the execution is actually proceeding, the menu has two main options that are enabled:

- The stop execution, release model option stops the execution at the earliest convenient stage and releases the license. No further interaction with this Model-based activity is possible.
- The stop execution, retain model option also stops the execution at the earliest convenient stage. However, the license is retained, and this allows the user to interact further with the executing activity via additional options in the execution interaction menu that become enabled at this stage, including:
  - Query variable allows the examination of the information available on a particular variable specified by its number as this is reported during the numerical solution;
  - Query block allows the examination of the information available on a particular block specified by its number as this is reported during the numerical solution. This option is only available when BDNLSOL is used as the non-linear solver and the block reported corresponds to the block in last initialisation/re-initialisation of the activity before it is stopped;
  - Query equation allows the examination of the information available on a particular equation specified by its number as this is reported during the numerical solution;
  - Query unit allows the examination of the information available on a particular UNIT specified by its complete pathname;
  - Create Model report creates a report on all parameters, variables and equations in the Model and incorporates this under the “Results” category of the current Case for later inspection or archiving;
  - Create saved variable set creates a gPROMS saved variable set using the current values of all variables; it then incorporates this under the “Results” category of the current Case.
  - Release license releases the license and essentially aborts the activity execution; no further interaction is possible with it.

All the options listed above available after stop execution, retain model can also be performed once the activity has stopped automatically (whether it has failed or completed successfully) if the Release model after execution option was not selected in the case configuration dialog at the start of the activity.

It is worth noting that (when in multiple windows mode) closing the execution output window by clicking on the X button at its top right corner does not cause the activity execution to be aborted or interrupted. In fact, the activity proceeds as before and all messages issued by it continue to be recorded within the Case. The execution output window can be made to re-appear at any time simply by double-clicking on the Execution Output entry in the Case sub-tree.

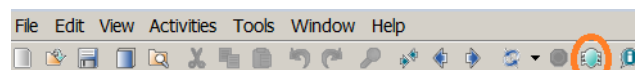
**Figure 5.8. Cases and activity execution**



## Diagnostics console






The diagnostics console provides an alternative window for interacting with the model - it can be added to the execution activity window with the diagnostic toolbar button.


**Figure 5.9. Diagnostics console Toolbar button**



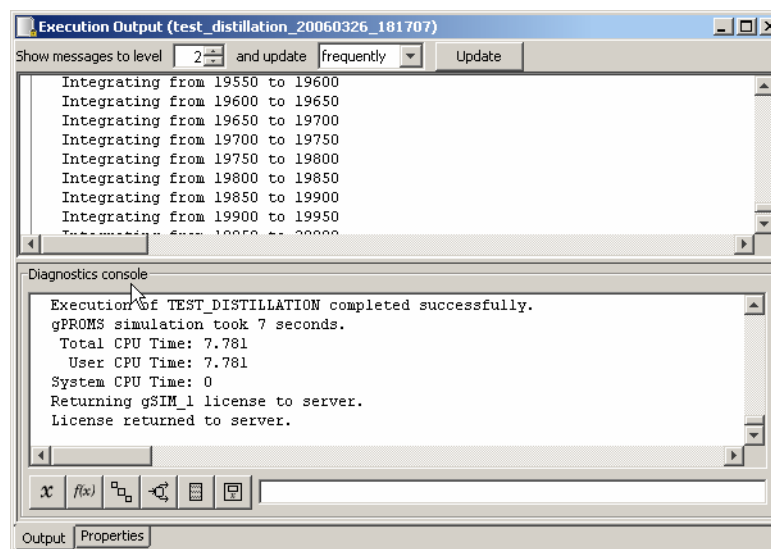
It displays the same information as the execution output during the simulation and can be interacted with in the same way by right clicking on the window. The diagnostics console provides a convenient way of interacting with the model whilst still being able to view the execution output in a separate window. It also provides a number of shortcuts for performing the various interactions with the models.

As well right clicking on the window interactions can be performed using:

- buttons at the bottom of the diagnostics console. The buttons available for interaction are:
  -  – query variable;
  -  – query equation;
  -  – query block;
  -  – query unit;
  -  – create model report;

-  – create saved variable set.
- commands in the command line at the bottom of the diagnostics console. The commands available are:
  - help [<command(s)>]
  - write [-v|--verbose] [-e|--edit] [<filename>]
  - save [-i] [-a] [-s] [-e|--edit] [<filename>]
  - qv <variable number(s)>
  - qe <equation number(s)>
  - qb [-b|--brief] <block number(s)>
  - qu <unit path(s)>
  - release – to release the license.

**Figure 5.10. Diagnostics console**



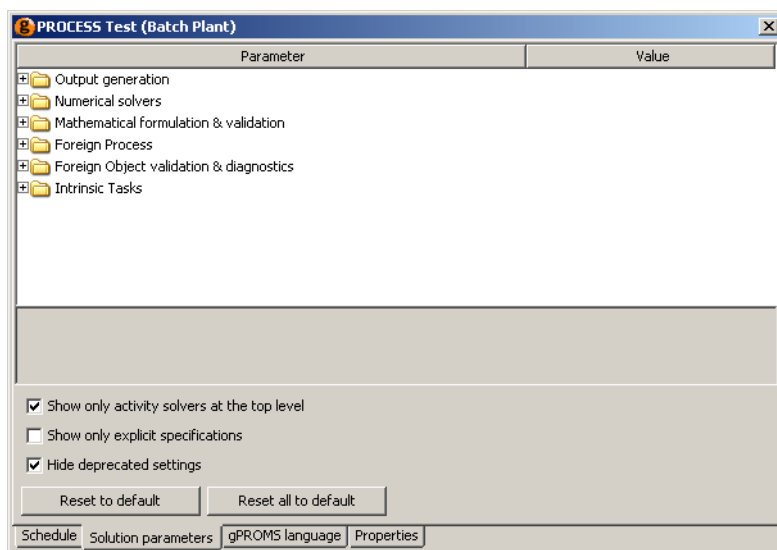
## Specifying Solution Parameters



When performing any of the activities associated with a Model, gPROMS uses a number of optional Solution Parameters. These are mainly used to specify options for the numerical methods applied and how to output the results of the activity. Although these Solution Parameters can be specified in the gPROMS language tab of a Process, it is more convenient to specify them using a graphical interface.

When the graphical method of specifying Solution Parameters is used, the SOLUTION\_PARAMETERS section of the gPROMS language tab of the Process is modified (or created if there is no section present). For this reason, it is important to make sure that any existing SOLUTION\_PARAMETERS section is free from syntax errors; otherwise, the section will be replaced and the data already present may be lost.

The Solution Parameters are specific to each Process and are accessed by selecting the Solution parameters tab.

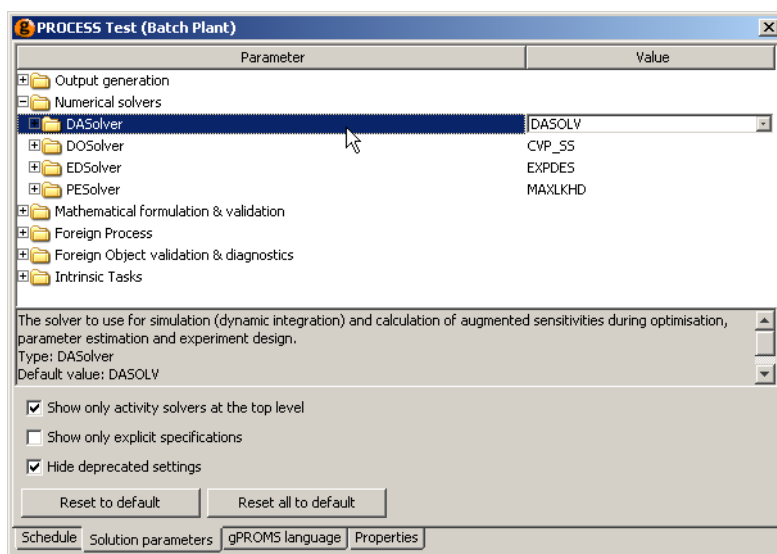
**Figure 5.11. Solution Parameters tab**



The top pane of the Solution parameters window is split into two columns which contain a tree view of all of the Solution Parameters and their values. The Solution Parameters are organised into groups of related parameters. Each group is represented by a "folder" icon in the tree view in the left-hand column. You can expand a group by left-clicking on the  symbol next to it, or by double clicking on it; left-clicking on the  symbol will collapse it. Each group may contain a number of Solution Parameters and subgroups. Solution Parameters are indicated by a bullet icon, with their value shown in the column on the right. Some Solution Parameters contain a value and also a set of subparameters. These are indicated by a folder icon along with their value in the right-hand column. These Solution Parameters can be expanded and collapsed just like the groups.

The image below illustrates the three types of entity in the tree view. Numerical Solvers is a group of related Solution Parameters; DASolver is a Solution Parameter containing further Solution Parameters, some of which also contain their own sub-parameters (such as LASolver) and some of which are simple Solution Parameters, such as EffectiveZero.

**Figure 5.12. Solution Parameters tab**



The figure above also illustrates that once a Solution Parameter is highlighted (by left-clicking on it), the middle pane then displays a brief explanation of the Solution Parameter, along with its default and allowed values.

To modify the value of a Solution Parameter, simply left click on the value. Depending on the type of Solution Parameter, you may be required to enter a new value or select one from a list. Once the Solution Parameter has been set a value other than the default, this is indicated by the text changing to bold face.


At the bottom of the tab are a number of controls that allow you to filter out some of the Solution Parameters shown in the tree view or reset the Solutions Parameters to their default values. See: filtering and resetting Solutions Parameters.

For further information, see:

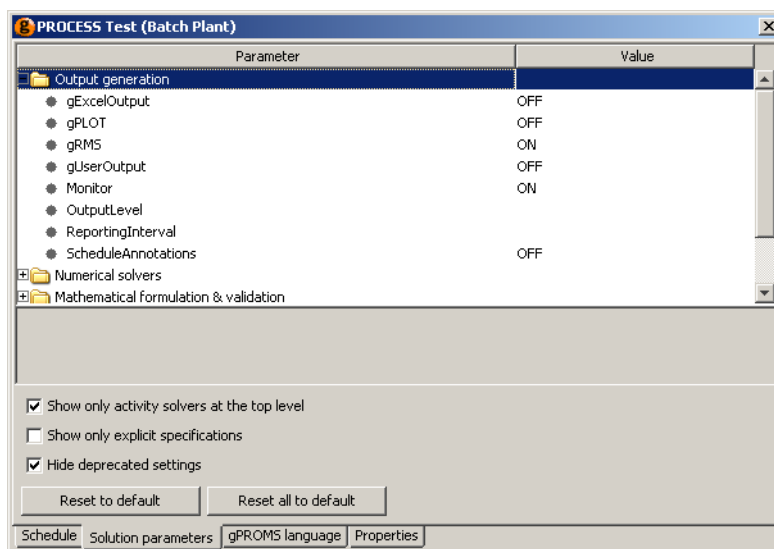
- Examples of setting Solutions Parameters
- Global specification and inheritance of Solution Parameters
- Filtering and resetting Solutions Parameters

## Examples of Solution Parameter Specifications

To illustrate how to set Solution Parameters using the graphical interface, consider the cases of setting the output reporting interval and specifying a file for the Excel output channel.

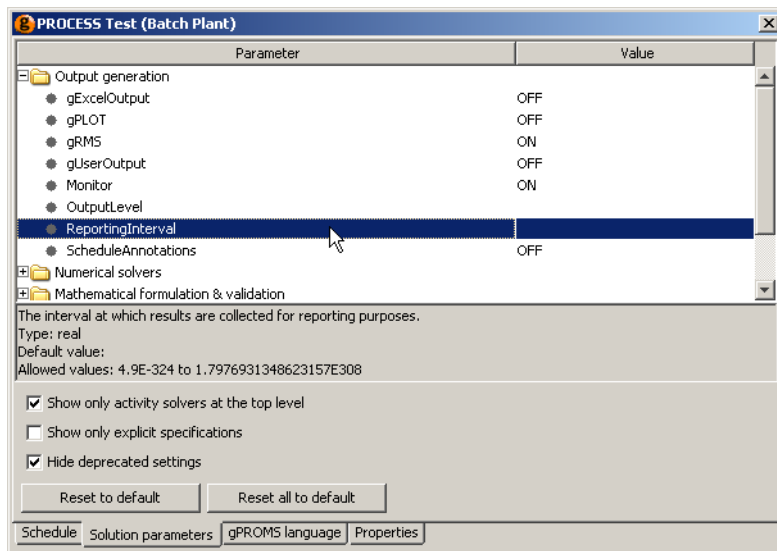
The `ReportingInterval` and `gExcelOutput` Solution Parameters come under the Output generation section, so left click on the  icon or double click on the Output generation "folder" to expand this section:

**Figure 5.13. Opening the Output generation section**



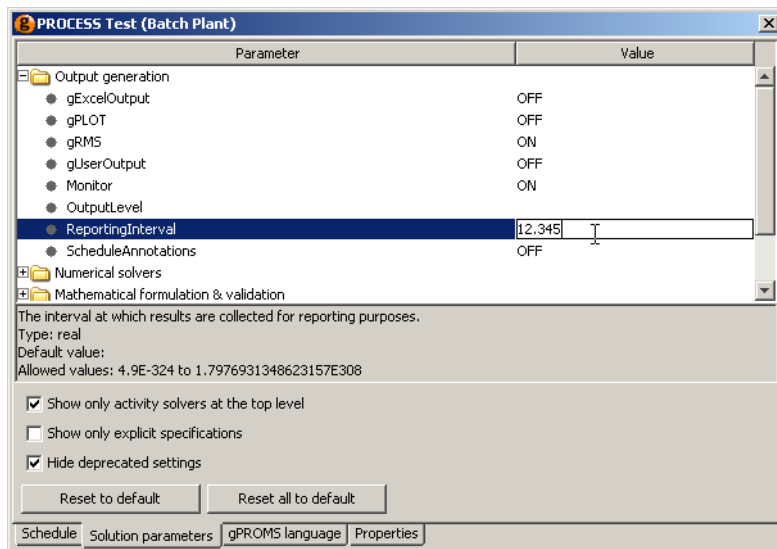
Now we can see the two Solution Parameters that we want to specify. Left click on `ReportingInterval` (or the place where the value will appear), and the following will be shown.

**Figure 5.14. Selected Solution Parameter**



We can see below the main view that it is of type real, with no default value and it can take values between  $4.9 \times 10^{-324}$  and about  $1.8 \times 10^{308}$ . To enter a value, double click on the Value place holder and enter a value:

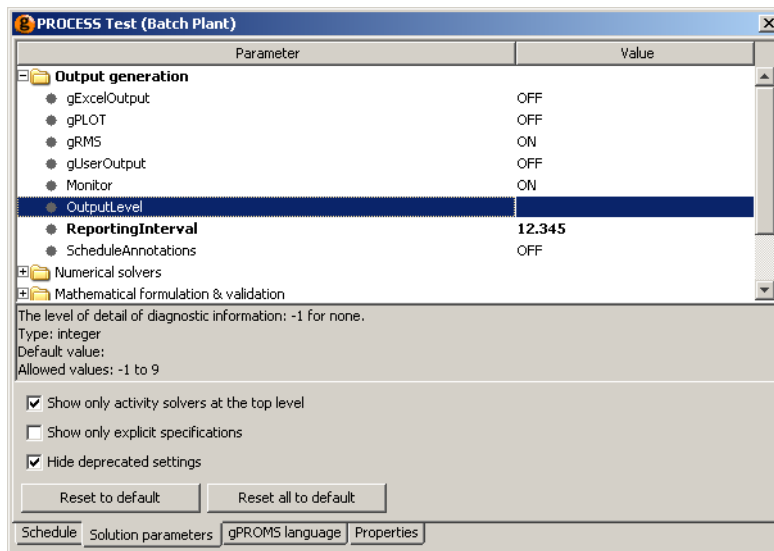
**Figure 5.15. Entering a Value**



And press return:

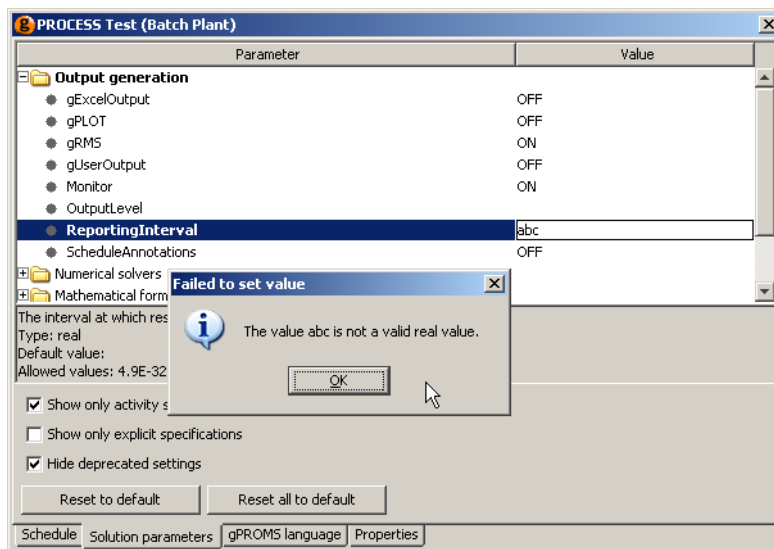


**Figure 5.16. Solution Parameter changed**



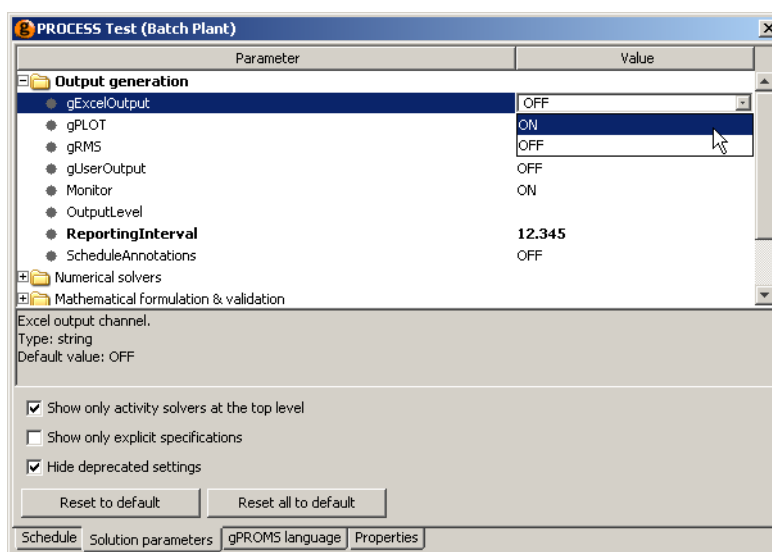
So we can now see that the value have been changed and the text has changed to bold face to indicate that it is no longer at the default value. Note that gPROMS will not allow invalid values to be entered, as shown below.

**Figure 5.17. Invalid entry of Solution Parameter values**



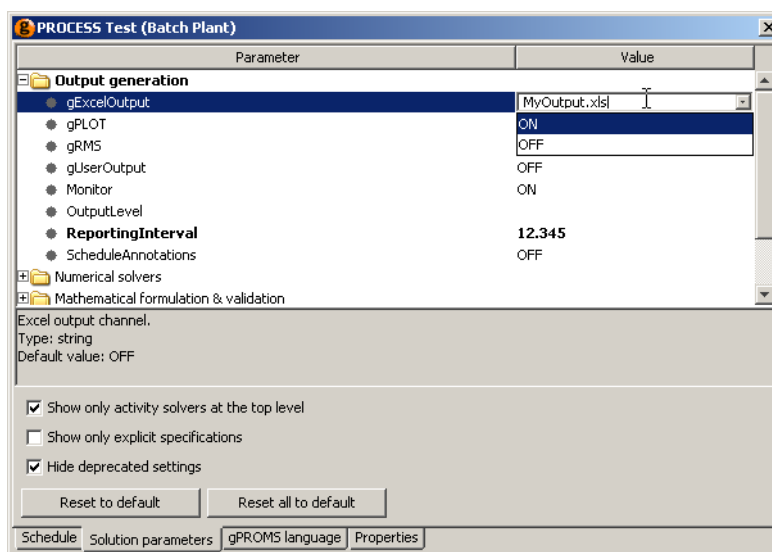
Now try double clicking on the value of gExcelOutput. You will now be presented with a choice of values: ON or OFF.

**Figure 5.18. Multiple choice value specification**



In this case, the value can take any string, so you have a choice of selecting one of the two predefined values or of entering a value yourself. (The behaviour of this Solution Parameter is described in The SOLUTION PARAMETERS Section.) So, to enter a value for the output channel file name, simply double click on the value again and enter a file name:

**Figure 5.19. Entering a value instead of the two provided**



and finally, by pressing the return key, gPROMS will accept the value. As before, the text will be rendered in bold face to indicate the non-default value.

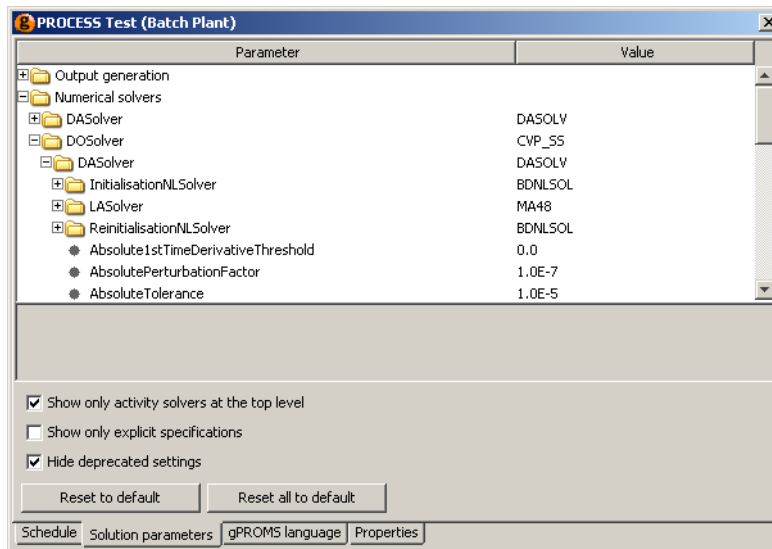
Any other Solution Parameter can be specified in just this way, by browsing the tree on the left to find the desired Solution Parameter and entering a value by double clicking on the value. The details of all Solution Parameters are given in The SOLUTION PARAMETERS Section.

Whenever a Solution Parameter is changed in this way, the gPROMS language tab is automatically updated. Similarly, any changes in the gPROMS language will be reflected in the Solution Parameters tab.

## Global Specification and Inheritance of Solution Parameters

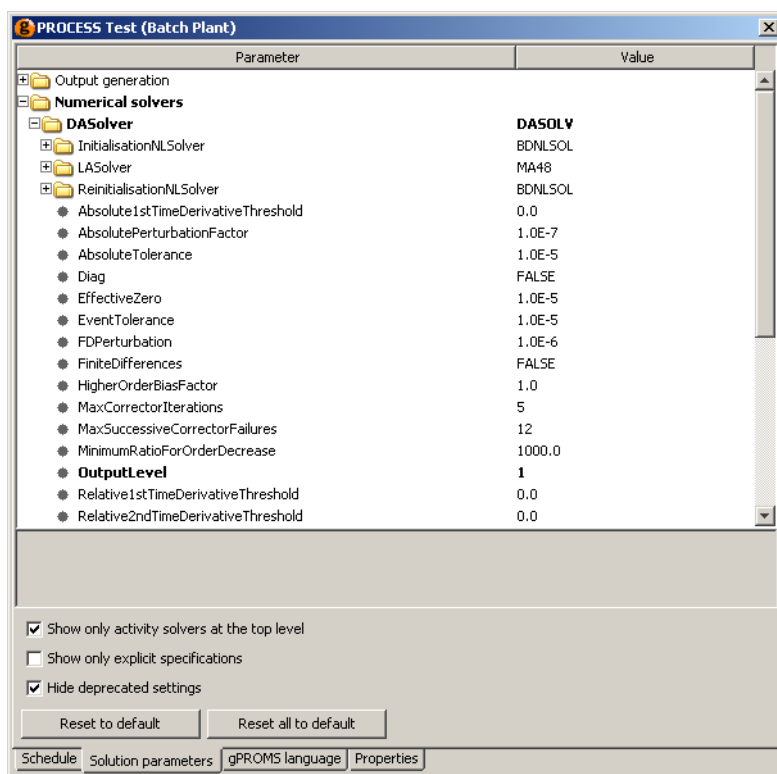
In order for gPROMS to perform any activity, it must apply a particular type of solver. Each solver typically decomposes the problem into a number of simpler problems that each require a different type of solver. Consider the Solution Parameters for the Dynamic Optimisation activity as shown below.


**Figure 5.20. Solution Parameters for Dynamic Optimisation**

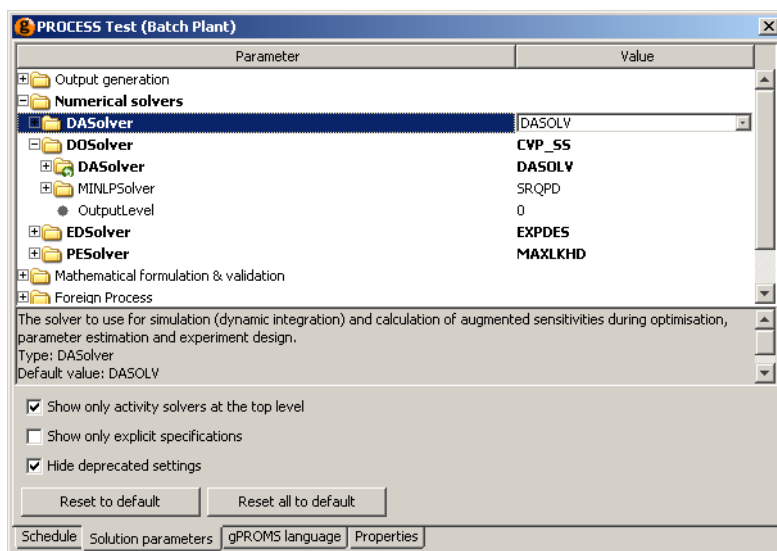



The top-level solver for Dynamic Optimisation is specified in the `DOSolver` Solution Parameter (in this case, its value is `CVP_SS`). The `DOSolver` Solution Parameter then contains a `DASolver` parameter, for solving differential-algebraic equations. This `DASolver` has three sub-solvers: two non-linear solvers for initialisation and reinitialisation and a linear-algebra solver. The non-linear solvers also make use of a linear-algebra solver, and therefore contain their own `LASolver` parameter.

Clearly, if one wanted to change the specification of one of the solver parameters for all occurrences in an activity, then setting all of these individually would be time consuming and prone to error. For this reason, when a solver parameter is specified at the highest level, its value is inherited by all of the same solver parameters at a lower level. To see this, let us change the `OutputLevel` of the `DASolver` to 1 and see what happens to the other solvers that have `DASolver` parameters.

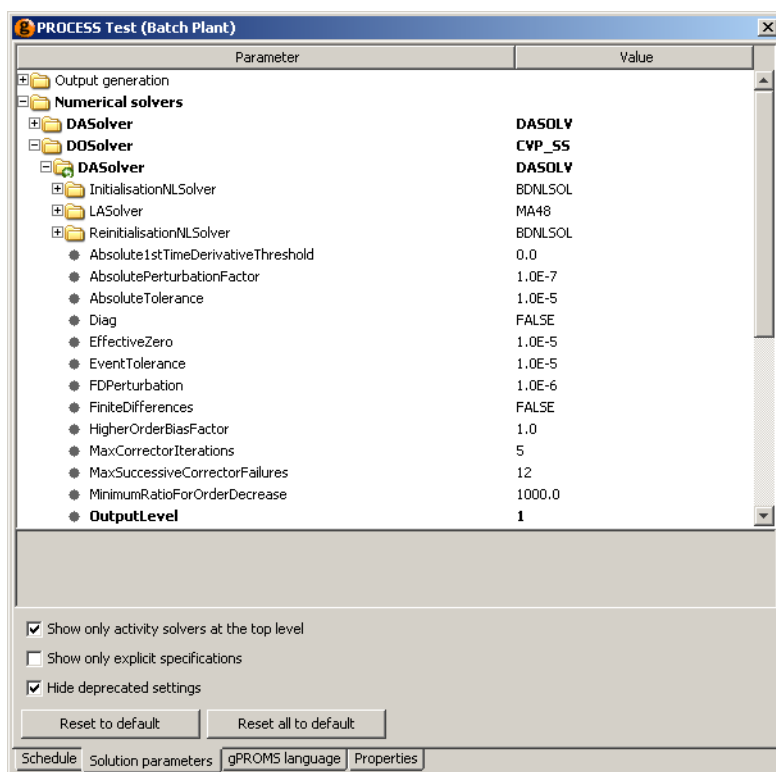
**Figure 5.21. Changing top-level DASolver OutputLevel**


First, we can see that the Numerical solvers Solution Parameter is now displayed in bold face, indicating that one of the Solution Parameters therein has been specified a non-default value. Similarly with the DASolver parameter, whose OutputLevel we have just changed to 1. Now, if we hide the details of the DASolver parameter, by left clicking on the  symbol, the following will be shown.

**Figure 5.22. Inheritance of OutputLevel in DASolver parameter**

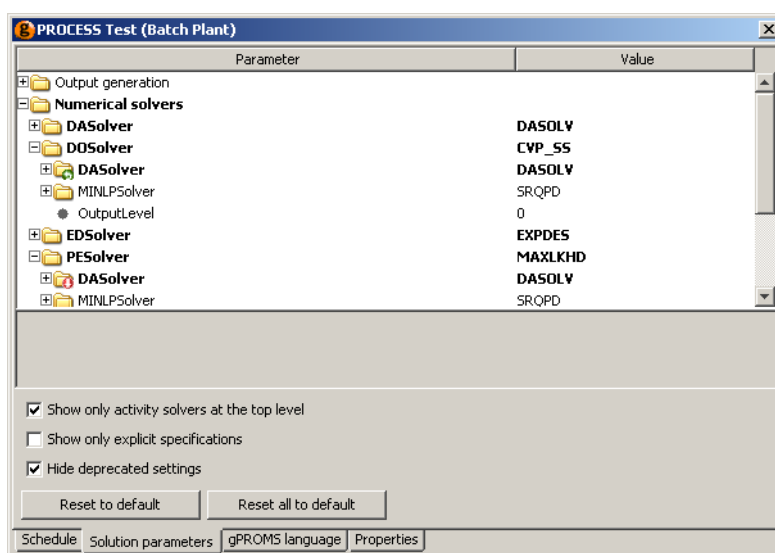
So we can see that the effect of specifying OutputLevel in DASolver has been to modify Solution Parameters within the solver specifications for Dynamic Optimisation, Experiment Design and Parameter Estimation, since all of these make use of a DASolver. The fact that the DASolver in DOSolver has inherited values is indicated by a  icon.


If we expand the DASolver parameter (inside the DOSolver section), we can see that the OutputLevel has changed there:

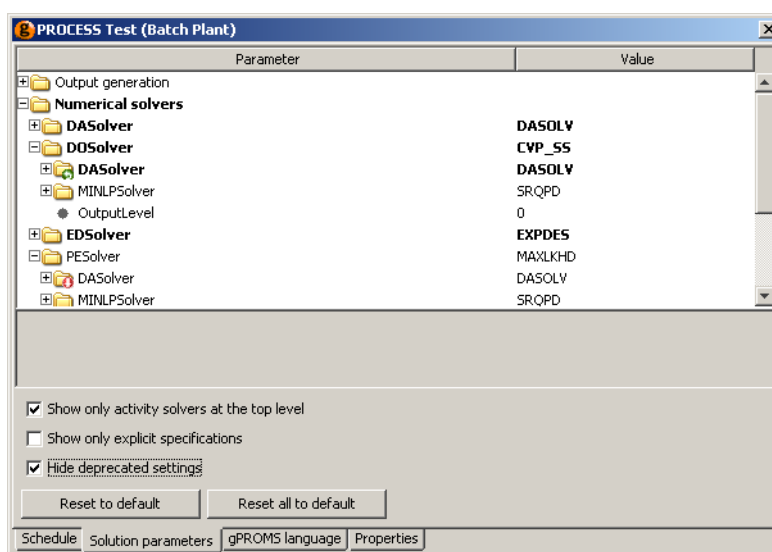


Now it might be that we don't want the value to be propagated to the DASolver parameters in the other activities. In this case, we can override the inheritance directly by going to the DASolver parameter and changing it. In the image below, we have changed the value of OutputLevel (in the DASolver of the PESolver parameter) to 3 and so the fact that the value overrides the inherited value is indicated by a  icon; the fact that the value is still altered from the default is shown, as usual, by the bold text.

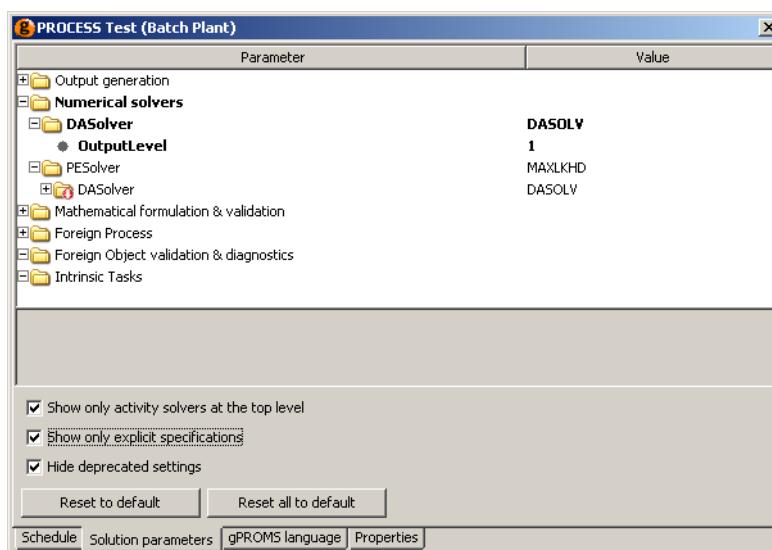
**Figure 5.23. Inheritance of OutputLevel in DASolver parameter**



Finally, if we change the value back to its default, then text returns to the normal weight to show that the parameter is at its default value but the  icon remains because it is overriding the setting at the higher level.

**Figure 5.24. Inheritance of OutputLevel in DASolver parameter**

One can imagine that once a lot of specifications have been made, it will become extremely difficult to track where an inherited value has come from. This task can be made much easier by checking the Show only explicit specifications box, as shown below.

**Figure 5.25. Inheritance of OutputLevel in DASolver parameter**

So in the figure above, we can see that the only specification that has been made is in the top-level `DASolver`. None of the other Solution Parameters is visible because they have not been specified explicitly. The `DASolver` parameter under `PESolver` is shown to indicate that it is overriding the earlier specification.

## Filtering the Display of Solution Parameters and Resetting Default Values

The Solution Parameters tab contains five controls in the bottom left-hand corner. These are for filtering out some of the Solution Parameters shown in the tree view and for resetting Solutions Parameters to their default values.

The three checkboxes allow you to:

- Show only activity solvers at the top level

In addition to specifying solver parameters for each type of activity, it is also possible to specify solver parameters that will apply to all activities. It is not recommended to do so, but if you want to specify solver parameters *globally* in this way, then uncheck the Show only activity solvers at the top level checkbox. The solver parameters will then appear under the Numerical solvers group along side the four activity Solution Parameters. These issues are discussed in detail in the section on global specification and inheritance of Solution Parameters.

- Show only explicit specifications

Because of the way that some specifications are propagated down to other Solution Parameters (see Global Specification and Inheritance of Solution Parameters), it can sometimes be difficult to identify where a particular specification was made. Checking the Show only explicit specifications box will remove all default and inherited Solution Parameters from the tree view, leaving only the Solution Parameters that were specified explicitly, simplifying the tree view considerably.

- Hide deprecated settings

Some projects developed using an old version of gPROMS may contain Solution Parameters that are no longer used or that are in a different format to the current version. When these projects are opened and the Solution Parameters tab is first opened, the Solution Parameters section will be converted to the new format. Any Solution Parameters that are no longer used will be deleted; those that are still used but are now listed under a different name or within a different section of the Solution Parameters will be shown separately under the Deprecated Settings heading. By default, this section is hidden, so if you have imported an older project, then uncheck the Hide deprecated settings box and browse the Deprecated Settings part of the tree to identify any Solution Parameters that can be kept. You will have to locate the new Solution Parameter and specify its value manually (see The SOLUTION PARAMETERS Section for a description of all of the Solution Parameters).

The two command buttons at the bottom left of the tab allow you to reset Solution Parameters to their default values. The two buttons are:

- Reset to default

This button will reset the currently selected Solution Parameter to its default value. When applying this to a Solutions Parameter that has inherited its value from a higher-level specification, then only this value will be reset and a "red" arrow icon will indicate that it is overriding the higher-level specification.

- Reset all to default

This button simply resets *every* Solution Parameter to its default value. Be sure that you really want to do this, as gPROMS will not ask for confirmation.

---

# Chapter 6. Viewing results

There are many ways of viewing simulation results from gPROMS ModelBuilder both as 2D or 3D plots:

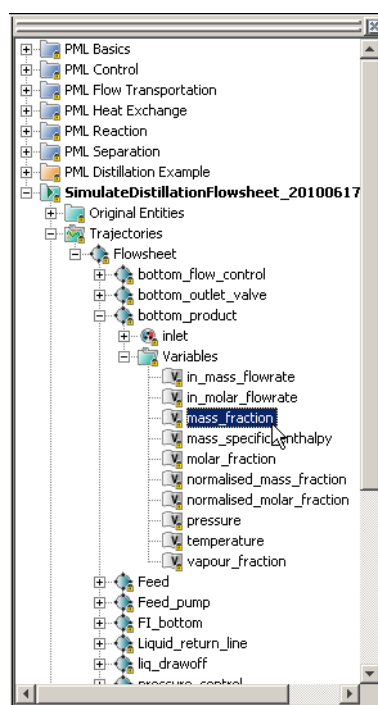
- Basic results can be viewed from within the Cases themselves as explained in the section on Inspecting results for an individual variable.
- gRMS (general Results Management Service) offers more powerful facilities for configuring plots.
- The Microsoft Excel™ Output Channel allows to view and process results in Microsoft Excel™.
- The gPLOT Output Channel exports the result as text files which provides full flexibility in postprocessing data in custom tools.

Flowsheet results stored in Cases provide direct access to Stream Tables or customised Reports.

## Inspecting results for an individual variable

gPROMS ModelBuilder stores the results of a simulation activity in the Case<sup>1</sup>.

**Figure 6.1. Finding a single variable in the project tree**



To access the results for a particular Variable:

- Expand the *Results* entity group in the desired Case
- Expand the Unit tree
- Expand the Variables folder under the desired unit
- Open the desired Variable by double-clicking it

---

<sup>1</sup>If enabled from the Execution Control dialog



- Results may be viewed in tabular form or as simple 2D/3D graphs by choosing the appropriate tab.

**Figure 6.2. Simulation results: table**

	Time			
	0.0	0.5	1.0	1.5
Axial 0.0	625.0	625.0054	625.0055	625.0055
0.06	625.0	632.3802	632.4321	632.4322
0.12	625.0	638.5396	639.0143	639.0148
0.18	625.0	642.4163	644.4617	644.4661
0.24	625.0	643.15	648.6482	648.6718
0.3	625.0	640.9314	651.4781	651.5695
0.36	625.0	637.0645	652.8927	653.1653
0.42	625.0	633.0383	652.9085	653.5682
0.48	625.0	629.7809	651.6422	652.9863
0.54	625.0	627.5641	649.3071	651.6854
0.6	625.0	626.2499	646.1938	649.9335
0.66	625.0	625.5583	642.6402	647.9587
0.72	625.0	625.2306	638.9932	645.9277
0.78	625.0	625.089	635.5609	643.9421
0.84	625.0	625.0324	632.5703	642.0464
0.9	625.0	625.0112	630.1451	640.2419

The figure above shows the Variable Results in tabular form for a one-dimensional array. In this case, there are many rows due to the large number of values reported during the simulation. One can easily navigate through this table by left clicking on a cell and then using the cursor keys to move the selected cell. The table will scroll automatically when you move the selection outside the visible part of the table. You can also use the mouse wheel to scroll the table more quickly and easily.

The value of a particular element of a Variable can be copied to the clipboard (and therefore pasted into another application) by right-clicking on the desired cell and selecting Copy from the context menu. You can also copy a value by moving the selection with the cursor keys, then pressing **CTRL+c**. Ranges of cells can also be copied. Either:

- Select the first cell by using the cursor keys, then hold down the **SHIFT** key and move the cursor to the last cell of the range. Finally, with all of the cells in the range selected, press **CTRL+c**.
- Left click on the first cell and drag the mouse pointer (keeping the left mouse button pressed) to the last cell of the range. Then right click and select Copy from the context menu.

Ranges are indicated by the grey box, as can be seen in the image above. The current cursor position is also highlighted in dark grey.

When there are variables that depend on more than 2 domains, all but 2 of the domains must be fixed in order to show the data in the table. The values of fixed variables can be specified in the box to the right of the domain. Values can be typed directly into the box or selected from the set of available values using three methods:

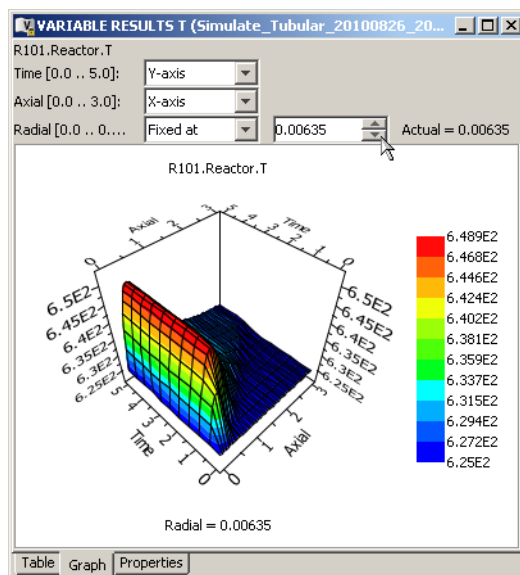
- left click on either of the spinner buttons (indicated by the mouse pointer in the image below) to move up or down in the list (holding the left mouse button down cycles quickly through)
- left click on the text in the box and
  - use the up and down cursor keys to change the selected value
  - use the mouse wheel to change the selected value

When there are many possible values (such as a Distribution Domain with a large number of intervals), cycling through all of the values can take some time. The process can be sped up by holding down the **SHIFT** and or

**CTRL** keys. When the **SHIFT** key is held down, one press of a cursor key or one click of the mouse wheel will move the selection by 10 points. The **CTRL** key skips 100 points, and holding down both keys skips 1000 points.

This method also applies to the fixed values of graphs.

**Figure 6.3. Simulation results: graph**



Note that this capability is complementary to the other gPROMS output channels. These include gRMS (gPROMS Result Management System - used for more advanced plotting capability)<sup>2</sup> and the gPROMS Excel Output Channel which provide additional ways of viewing the results of a Simulation activity.

## Exporting the results

The data for a single Variable can be exported as single Variable to a .csv file, which can be opened for example by Microsoft Excel™ and Microsoft Word™. To do this, right click on the results table for a Variable to activate the Results shortcut menu and select Export table.... Alternatively, the table data (or a subset of them) can be copied to the clipboard to be pasted into another application: to do this, highlight the range of cells that you would like to copy by left clicking on a cell and dragging the mouse pointer to another cell (or simply left click on one cell if you only need one value), then right click on the table and select Copy from the Results shortcut menu.

If you need to export a large amount of data, then it is more convenient and flexible to use the facilities described in Exporting Data to CSV Files.

The complete results set for a simulation can be exported to a .gRMS file to be read by gRMS. To do this select the Simulation Trajectory entity group and select Export.

## Printing the results

The graph of a single Variable can be printed by right-clicking on the graph to activate the Results shortcut menu and select Print.... This activates the Print Preview dialog with just that graph selected for printing. There, you may view how the graph will be printed, change any of the page settings by pressing the Page Setup button and then send the graph to the printer by pressing the Print... button.

The complete results set for a simulation can be printed in one go using the File menu. Full details of this, including further information on the Print Preview, Page Setup and Print dialogs, can be found here.

<sup>2</sup>Sending results to gRMS can also be configured from the Execution Control dialog.

## Viewing stream tables

After the simulation has run, values of quantities within the streams that make up the flowsheet at different times can be viewed via stream tables. They can be viewed in a number of ways:

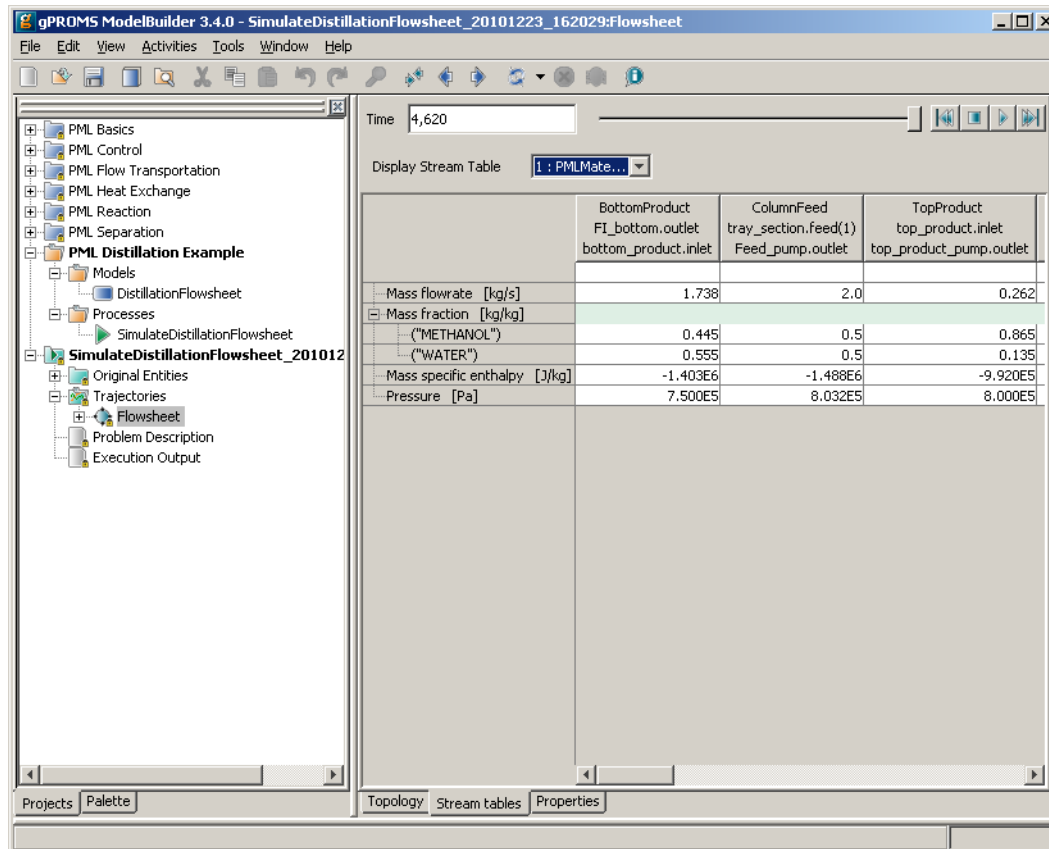
1. Double click on a Model that contains connections or the Execution Output, then select the Stream Tables tab (the Execution Output may contain more than one tab: one for each model with connections)

This will show a single stream table showing all connections in the Model. Each connection occupies one column of the stream table, with the column heading containing the connection name (if given), the name of the Unit connected upstream and the name of the Unit connected downstream. So in the screen shot below, the first column is the stream named `BottomProduct`, which connects the outlet of the flow sensor called `FI_bottom` to the inlet of the sink called `bottom_product`. The connections that have names will be shown first, in alphabetical order, and the unnamed connections will be shown last.

2. Stream tables containing a subset of the available connections can be created by:
  - a. double clicking on a Model that contains connections or the Execution Output and then selecting the topology tab (again, the Execution Output may contain more than one topology tab)
  - b. right clicking on a stream<sup>3</sup> and selecting Create stream table.

This will open a new pane at the bottom of the Topology window. Each time a stream table is created, it will be added as a new tab in the bottom pane. Each tab can then be undocked from the window by right clicking and selecting either Detach window, which opens the stream table in a window outside ModelBuilder, or Float Window, which opens the stream table in a window inside ModelBuilder (to redock the window, right click and select Dock window).

**Figure 6.4. Stream tables tab**



<sup>3</sup>Multiple streams may be selected by use of **CTRL**

Figure 6.5. Docked stream tables

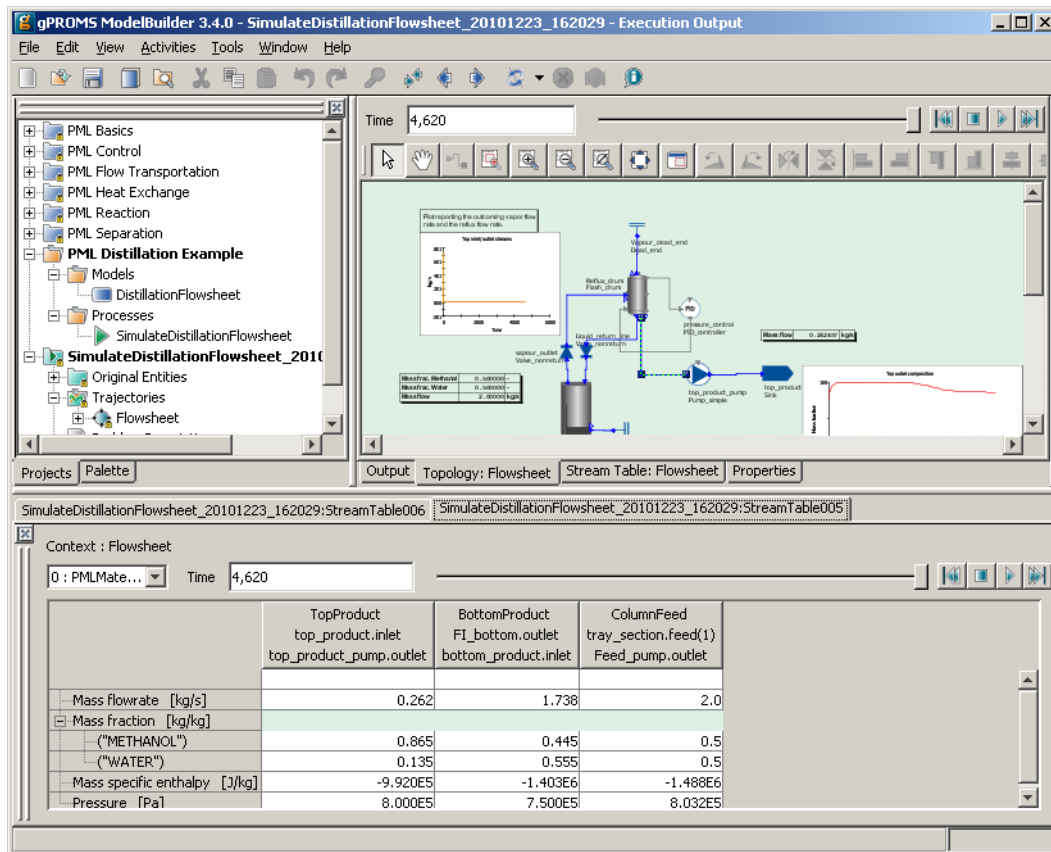
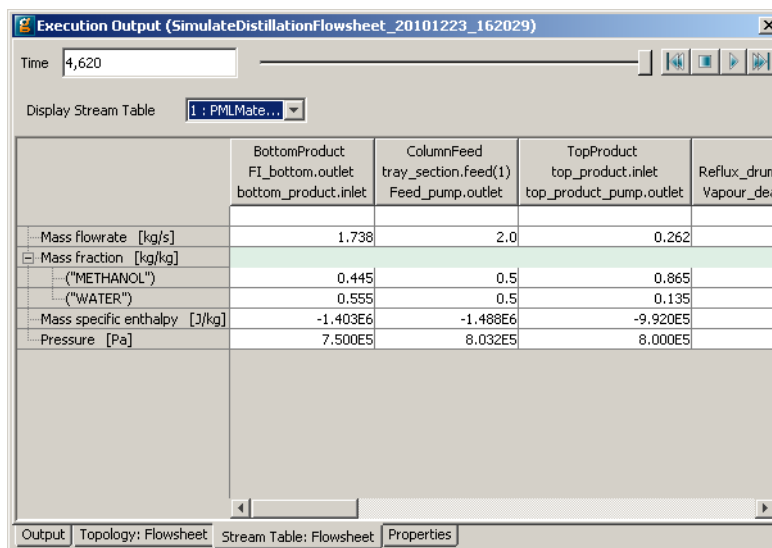



Figure 6.6. Detached stream table



A docked stream table tab may be closed by left clicking on the  button at the top left of the pane.

The Playback Toolbar at the top of the window/pane allows you to change the time for which the results are displayed (see Viewing results on the Flowsheet during and after simulation).

Finally, stream tables display only connections of the *same type*. To specify the type of connections shown, select an option from the Display Stream Table listbox. The examples above show all connections of type PMLMaterial; the other alternative in this case is PMLControl, which shows the signals in the control connections.

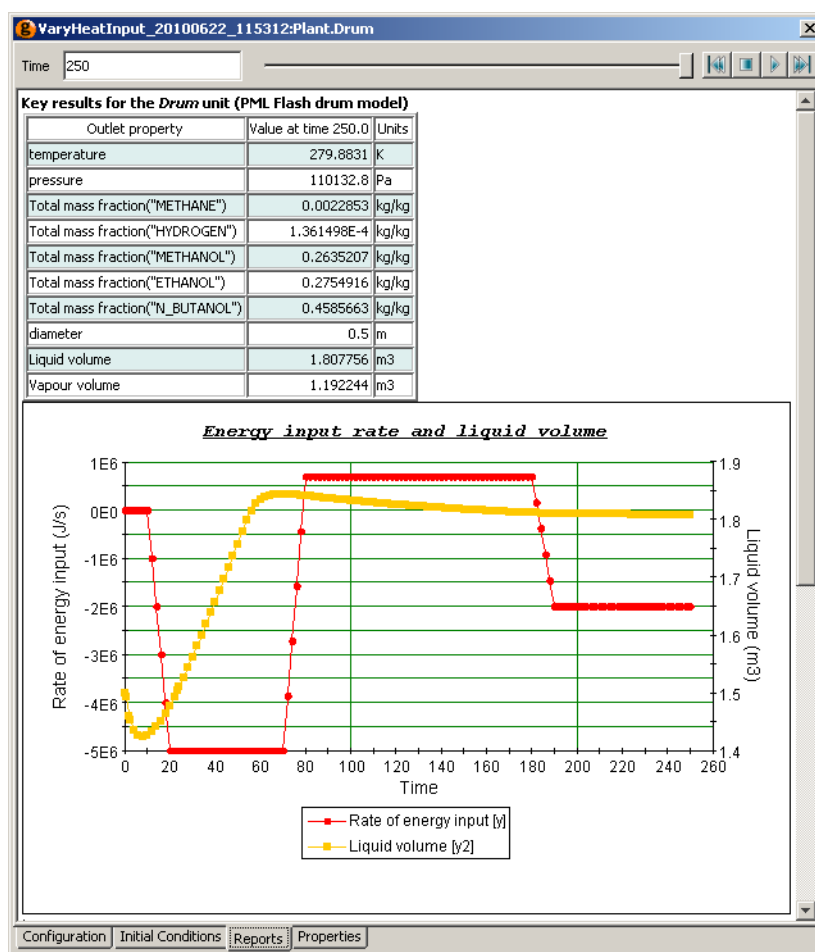
## Exporting stream tables

The data of an entire stream table can be exported to a .csv file, which can be opened for example by Microsoft Excel™ and Microsoft Word™. To do this, right-click on the results table for a Variable to activate the context menu and select Export table. Alternatively, the stream table data can be copied to the clipboard to be pasted into another application by selecting Copy from the context menu.

## Viewing Model reports

A model developer may have written HTML report templates for Models in the post-simulation topology. These can be accessed by double clicking on the Model in the flowsheet (or right clicking and selecting the Show report option).

Figure 6.7. A model report



The scrollbar across the top will again show the variables changing with time.

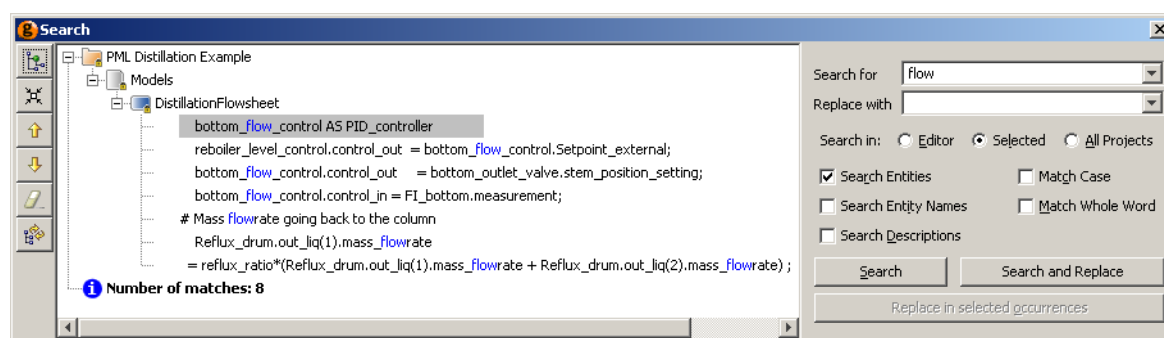
# Chapter 7. Modelling Support Tools

gPROMS ModelBuilder provides a set of powerful tools to support the user in developing and maintaining complex models. These are accessed from the Tools menu. These include productivity tools that allow the user to perform search and replace operations on the whole project tree and a comparison tool. The variable browser provides a convenient view of the different variables in a particular Process. Files can be imported into or linked too ModelBuilder and various export operations from ModelBuilder can be performed.

## Global Search-and-replace

The search-and-replace tool can be used to search for a text string within selected Entities or Projects, and optionally to replace it with another string. The tool can be accessed via ModelBuilder's Tools menu and appears as a separate window, normally located at the bottom of the main ModelBuilder window. The user has to specify a string to be searched for, and optionally a replacement string. The search can take place over all Projects that are currently open in ModelBuilder, or be limited to one or more Entities selected by left-clicking<sup>1</sup> on their names in ModelBuilder's navigation tree. Various other aspects of the scope of the search<sup>2</sup> can be specified by enabling or disabling the check-boxes provided.

Figure 7.1. The search and replace tool



Clicking on the Search button causes the search results to appear in the form of an occurrence tree, organised by Project, Entity group and Entity, in the left part of the search-and-replace window.

Clicking on the Search and Replace button has a similar result, with the additional effect that all occurrences of the search string are changed to the specified replacement string. A replacement may be applied selectively by selecting one or more entries in the occurrence tree and clicking on the Replace in selected occurrences button. In this case, the selection moves on automatically to the next occurrence. This allows the replacement to be applied one-at-a-time in a controlled fashion.

The navigation buttons situated on the left-hand side help the user to manage the search results. These include:

- Expand all nodes in the search tree
- Collapse the search tree
- Select the next match
- Select the previous match
- Delete selected nodes from the search tree
- Select all visible replaceable matches in the search tree

Double-clicking on any Entity name in the occurrence tree will cause the corresponding Entity editor to be displayed in the main ModelBuilder window. Double-clicking on a particular occurrence will have a similar effect, with the cursor located at the precise position within the Entity editor.

<sup>1</sup>Standard Shift-left Click and Ctrl-left Click mechanisms can be used to select multiple Entities.

<sup>2</sup>e.g. whether the string should be an entire word or a sub-string within another word.

The effects of any “replace” actions affected by ModelBuilder on a particular Entity can be reversed by opening the relevant Entity and pressing “undo” (Ctrl-Z) a sufficient number of times.

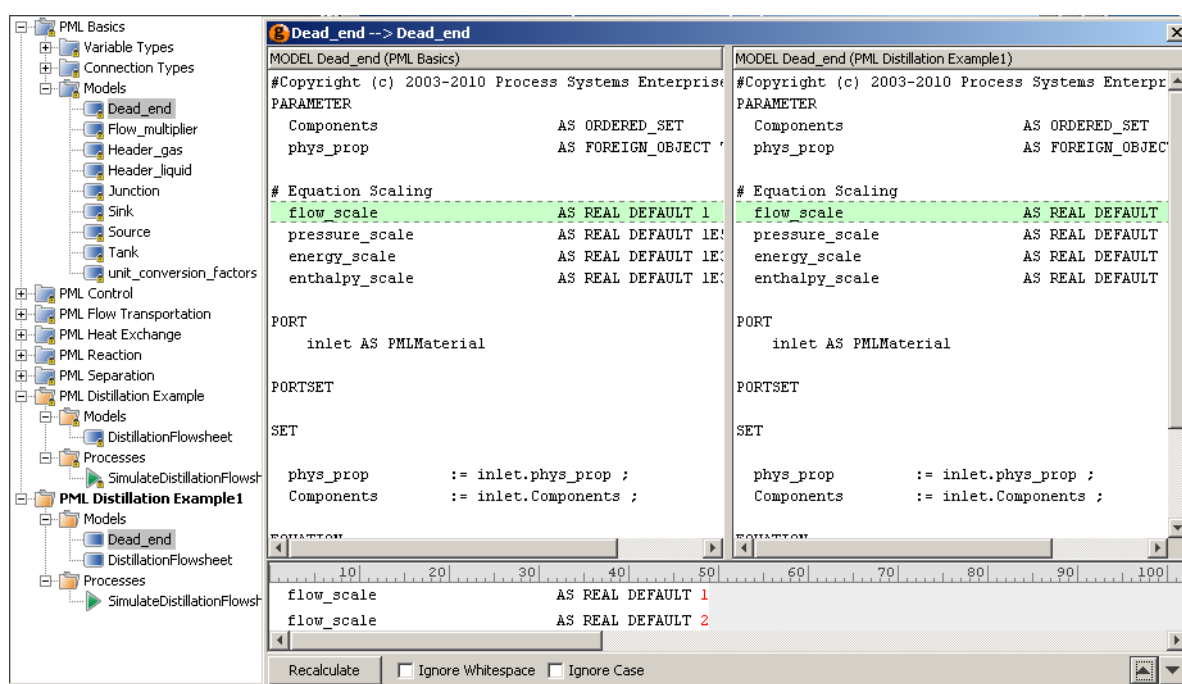
## Project and entity Compare

gPROMS ModelBuilder provides a comparison tool that can be used to compare any two selected Entities or Projects. The tool can be accessed via ModelBuilder's Tools menu, provided two comparable entries (e.g. Entities or Projects or Entity groups) have been selected in ModelBuilder's main navigation tree.

### Entity comparison

When two Entities are compared, a comparison window will pop-up highlighting the differences between them (e.g. any lines that are different in the two Entities, or which have been added or deleted). The colours used for the highlighting can be configured as part of the user's ModelBuilder Preferences by highlighting the “Text comparison” entry in the Preferences dialog.

**Figure 7.2. Comparison of two Entities**



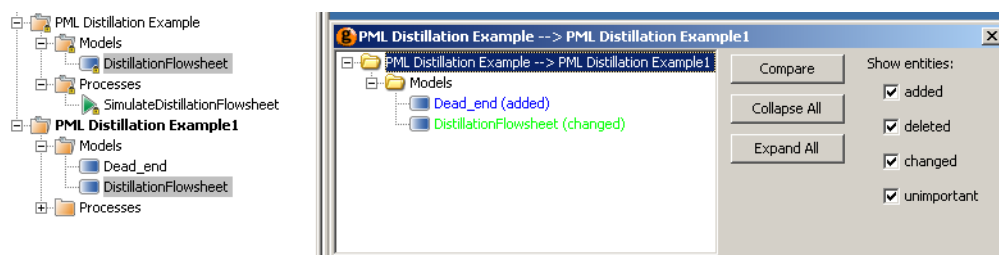
Please note that the comparison can be configured to ignore white spaces or differences in capitalisation. If enabled, only differences in actual content will be found and displayed. The same holds true for Project comparison.

### Project comparison

The comparison tool may also be applied to two entire Projects. In this case, the results of the comparison will appear in a separate window, normally located at the bottom of the main ModelBuilder window. In general, all Entities which occur in one of the two Projects but not in the other, or which occur in both Projects but in different versions, are presented in a comparison tree form in the left part of the comparison window.

Different colours are used to denote various types of difference (e.g. addition, deletion, modification) in the comparison tree. These colours can be configured as part of the user's ModelBuilder Preferences by highlighting the Group comparison entry in the Preferences dialog<sup>3</sup>.

<sup>3</sup>The content of the comparison tree can be controlled via the check-boxes provided in the right part of the comparison window.

**Figure 7.3. Comparison of two Projects**

Double clicking on an entry in the comparison tree corresponding to a deleted or added Entity will cause the corresponding Entity editor to pop up in the main ModelBuilder window. Double clicking on an entry corresponding to a modified Entity causes the Entity comparison window to appear.

The comparison tool may also be used to compare two Cases or a Case and a Project. This is particularly useful when attempting to compare a current model with one that was used in the past to produce certain archived results.

## Entity group comparison

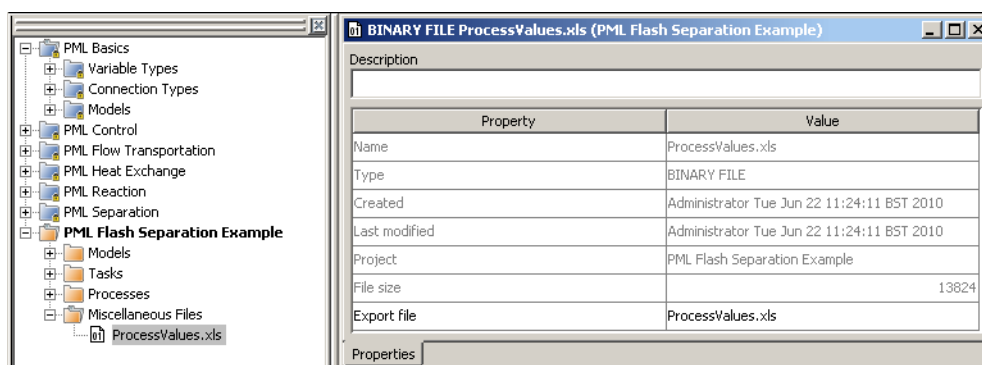
The comparison tool may also be used to compare corresponding groups of Entities (e.g. all Model Entities) residing in two different Projects. The behaviour of the comparison tool in group comparison mode is identical to that in the Project comparison mode.

## Import files

In addition to text files containing gPROMS language descriptions of various Entities<sup>4</sup> as well as other plain text files, ModelBuilder allows importing non-text (binary) files into a Project. These are held within the current Project under the *Miscellaneous files* category.

This useful capability allows any type of file that is necessary for the correct operation of a Project (e.g. a MS Excel spreadsheet used in conjunction with gPROMS' Excel Foreign Object/Foreign Process<sup>5</sup>) to be incorporated within the Project, thus making the latter completely self-contained in this respect.

Double-clicking on a binary miscellaneous file brings up a Properties dialog similar to the Properties tab of the standard Entity editors. A useful feature of this dialog is the “Export filename” property. This allows the user to specify the export filename as well as the sub-directory of the gPROMS execution into which this external binary file should be copied during execution of any gPROMS model-based activity. For example, consider a gPROMS model making use of a MS Excel Foreign Object described by the MS Excel file `crystalliser.xls`. Suppose, further, that the PROCESS Entity describing a simulation activity SETs the value of this Foreign Object as “ExcelForeignObjects\crystalliser.xls”. In this case, the export filename for this binary file should be set to “ExcelForeignObjects\crystalliser.xls”.

**Figure 7.4. Binary files imported in Project**

<sup>4</sup>i.e. .gPROMS, .gOPT, .gEST, .RUN and .gSTORE files compatible with gPROMS v1.x.

<sup>5</sup>see Foreign Objects and Foreign Processes.

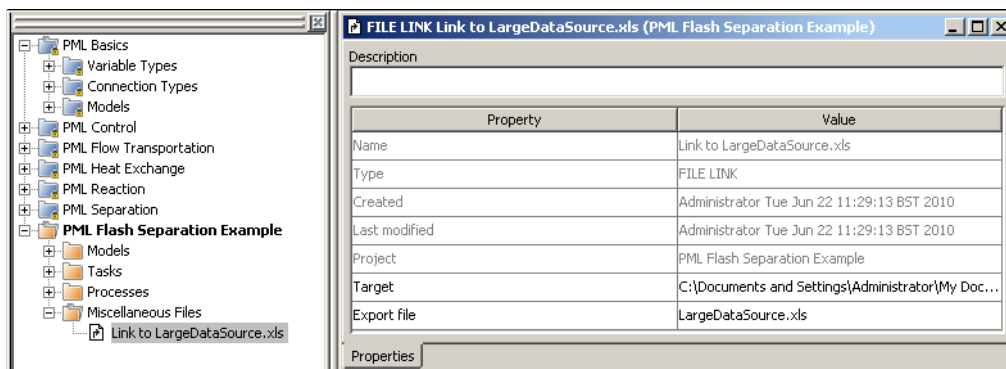


## Create links to external files

It may sometimes be impractical and/or undesirable to import all files needed for the execution of a Project's model-based activities within the Project. This is likely to be the case, for example, when such files are very large or when they are shared files that may need to be updated centrally from time to time. On the other hand, it may still be desirable to inform the Project explicitly of the association with these files and the need for copies of them to be present in certain sub-directories during execution.

The above requirements can be addressed by establishing a link to the external file without physically importing it into the Project. ModelBuilder supports this activity via the Create links to external files tool provided under the Tools menu. The links established using this tool are displayed under the Project's Miscellaneous Files category .

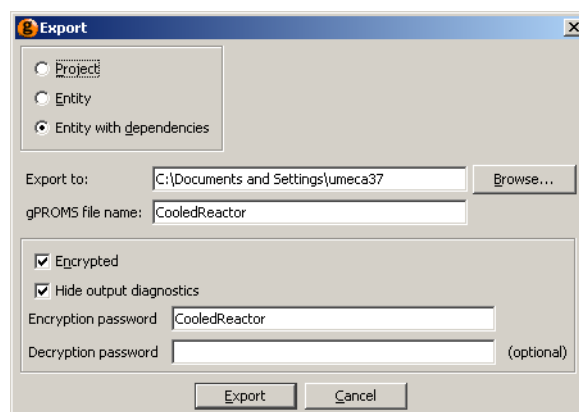
**Figure 7.5. Link to external files**



## Export

ModelBuilder allows the contents of a Project or an Entity to be exported as a plain text file containing the corresponding gPROMS language description. This type of operation, which is useful for exporting models for use with gPROMS-based applications (e.g. gO:Matlab), can be performed using the Export tool provided under the Tools menu.

**Figure 7.6. The export tool**



## Export Entity with dependencies

When exporting an individual Entity, the user may request that all other Entities on which it depends (via cross-referencing) should be exported with it.

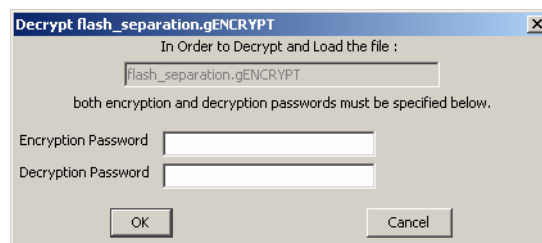
This is achieved automatically by ModelBuilder irrespective of whether these Entities reside in the same Project as the Entity being exported, or in other Projects being cross-referenced by the current Project. The gPROMS

language representation of each Entity in the export file is annotated with a comment stating the Project from which this Entity was extracted.

## Encryption

The user can also request that the exported text file is encrypted. The encryption is performed using standard gPROMS file encryption facilities and makes use of a pair of passwords supplied by the user. The first password must be supplied for the use of the exported file within any gPROMS-based application. The second (optional) password is needed to decode the file and have access to its contents<sup>6</sup>. If this password is given then the encrypted file can be opened in ModelBuilder. To open an encrypted file, use the normal open file methods (File>Open, the open tool bar button, etc) and change the Files of type field in the open dialog to “Encrypted gPROMS input files (\*.gENCRYPT)”. Browse for and select the encrypted file to be opened and select open. A dialog then appears asking for the encryption and decryption passwords to be specified.

**Figure 7.7. Open encrypted file dialog**



## Hide output diagnostics

When the exported entity is run, it may not be desirable to have it display diagnostic output. In order to suppress this, ensure that the Hide output diagnostics box is checked. This will then set the `OutputLevel` (see The SOLUTIONPARAMETERS Section) for this entity to -1. If Hide output diagnostics is unchecked, then the `OutputLevel` is set to the value of `OutputLevel` in the top-level entity; if this is unspecified, then the default value of 0 is used for the exported entity.

## Export to ModelBuilder v2.3 Project

ModelBuilder v3.0 uses a different file format for storing its projects to that used by previous versions of ModelBuilder. In the interests of forward compatibility, ModelBuilder v3.0 projects may be exported in v2.3 format using the Export to ModelBuilder v2.3 Project tool in the Tools menu. Note that not all gPROMS v3.0 concepts are recognised by earlier versions of gPROMS ModelBuilder (for example icons, ports).

## Export to Simulink

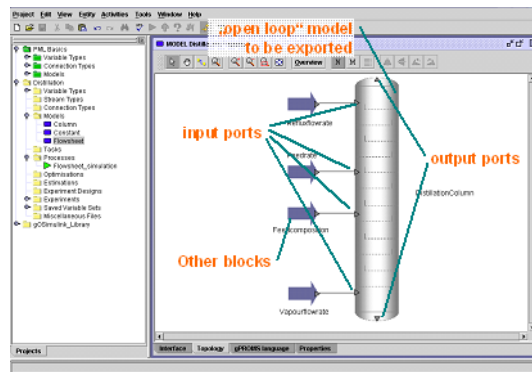
ModelBuilder can export gPROMS models to be used as Simulink®<sup>7</sup> blocks, see gO:Simulink User Guide for information on how to use the exported model.

To export a gPROMS model as a gO:Simulink block, a working Process that uses a flowsheet containing the model must first be created in ModelBuilder. The model that is to become a gO:Simulink block (“open loop” model) must contain ports, and these become the inputs and outputs of the block in Simulink®.

---

<sup>6</sup>Leaving the second password blank in the export dialog will ensure that the contents of the exported file cannot be viewed in ModelBuilder; only used within an model-based activity.

<sup>7</sup>Simulink is provided by Mathworks Inc. [<http://www.mathworks.com/products/simulink/>]

**Figure 7.8. gPROMS Model with Ports**

A gSTORE file is needed to provide good initial guesses for the model and must be present in the project. Once the desired working Process is created the following steps should be followed to export it to gO:Simulink:

- Highlight the Process,
- Select Tools from the menu bar,
- Select Export to Simulink....

This will pop-up a dialog box with various options to configure the exported files.

**Figure 7.9. Export to Simulink dialog**

The fields in this dialog box are described in more detail in the following table:

**Table 7.1. Fields in Export to Simulink Dialog**

Export Directory	The directory in which the files exported will be located in.
Dictionary directory	The directory in which dictionary files are located for models in gPROMS that can be replaced by standard Simulink® models.
Simulink model (mdl) file	The name of the Simulink® file to be created.
gPROMS based Simulink block	The unit in the Process that is to be the gO:Simulink block.

Use assignments from saved variable set	This selects the gSTORE file from which the values for the input and output ports are taken from to allow the open model to be run.
Overwrite Simulink model (mdl) file	The export will write over the existing mdl file of the same name if it already exists.
Include generated open loop process and model in the current project	A model and corresponding process to run it will be created in the project file that consists of just the gPROMS unit that was exported as the gO:Simulink block (open model).
Encryption password	During the export, ModelBuilder creates an encrypted file which is then used by gO:Simulink. The encryption password must then be used to enable gO:Simulink to run the encrypted file.
Decryption password (optional)	If a decryption password is provided the exported files can be re-imported into ModelBuilder, e.g. for debugging. Not providing an decryption password will encrypt the files without giving the option to decrypt them in ModelBuilder, thus fully protecting the content from the end-user of the exported files.

## Export to CAPE-OPEN

ModelBuilder can export gPROMS models to be incorporated as steady-state unit operations in CAPE-OPEN compliant process modelling environments (PMEs). The gPROMS model needs to be a steady-state model, a dynamic model which initialises at steady-state is not sufficient.

A gPROMS model that is intended for use as a unit operation should declare its ports using the standard PMLMaterial connection-type and use the Public Model Attribute mechanism to identify variables that will be presented to the PME as CAPE-OPEN parameters.

Once the desired Model is created the following steps should be followed to export it to gO:CAPE-OPEN:

- Highlight the Model,
- Select Tools from the menu bar,
- Select Export to CAPE-OPEN.

This will pop-up the 'Export CAPE-OPEN Unit Operation Wizard' that will guide you through the required steps:

- Basic properties
- Port mappings
- Parameter mappings
- Additional files
- Advanced options
- Entity generation options
- Export options

## Basic properties

The 'Basic properties' page is used to specify a name, description, and other identification information for the exported unit operation.

**Figure 7.10. Export CAPE-OPEN Unit Operation Wizard: Basic properties page**

**Export CAPE-OPEN Unit Operation Wizard**

**Overview**

1. Basic properties
2. Ports
3. Parameters
4. Files
5. Advanced
6. Entity generation
7. Export

**Basic properties**  
Specify name, description, author, version, etc. information about the unit operation

Name: MethanolSynthesisReactor

Author: thwill

Date: Thu Jun 05 15:51:18 BST 2008

Version: 0.1

Description:

Steady state tubular reactor for MethanolSynthesis b  
PML tubular reactor model (PML Reaction::Reactor\_tub

Errors/Warnings

Previous Next Last Finish Cancel

## Port mappings

Ports on the gPROMS model are mapped to CAPE-OPEN ports on the exported unit operation using the 'Ports' page of the export wizard.

**Figure 7.11. Export CAPE-OPEN Unit Operation Wizard: Ports page**

**Export CAPE-OPEN Unit Operation Wizard**

**Overview**

1. Basic properties
2. Ports
3. Parameters
4. Files
5. Advanced
6. Entity generation
7. Export

**Ports**  
Map gPROMS ports to CAPE-OPEN ports

Name	Type	Direction
feedStream	MATERIAL	INLET
productStream	MATERIAL	OUTLET

gPROMS Id: feedStream

Basis: MASS

Description:

This is port 'feedStream'

Errors/Warnings

Previous Next Last Finish Cancel

- Each gPROMS port is mapped to a CAPE-OPEN material port on the exported unit operation;

- gPROMS ports must be of the PMLMaterial connection-type; a model containing ports of any other connection-type cannot be exported.
- gPROMS ports must not be bi-directional; a model containing bi-directional ports cannot be exported.
- gPROMS inlet ports are mapped to CAPE-OPEN material inlet ports.
- gPROMS outlet ports are mapped to CAPE-OPEN material outlet ports.
- Each port can be provided with a unique name and description that will be used by the CAPE-OPEN interface; the default name is the same as the gPROMS Id.

## Parameter mappings

Public Model Attributes (PMAs) from the gPROMS model's default configuration mode are mapped to CAPE-OPEN parameters on the exported unit operation using the 'Parameters' page of the export wizard.

**Figure 7.12. Export CAPE-OPEN Unit Operation Wizard: Parameters page**

Name	Type	Mode	Default	Units
Bed porosity	REAL	INPUT	0.48	
Catalyst density	REAL	INPUT	1700.0	
Enthalpy datum	OPTION	INPUT	molecular	
Heat input rate	REAL	INPUT	-1.1E7	J/s
HTC	REAL	INPUT	1000.0	W/m.K

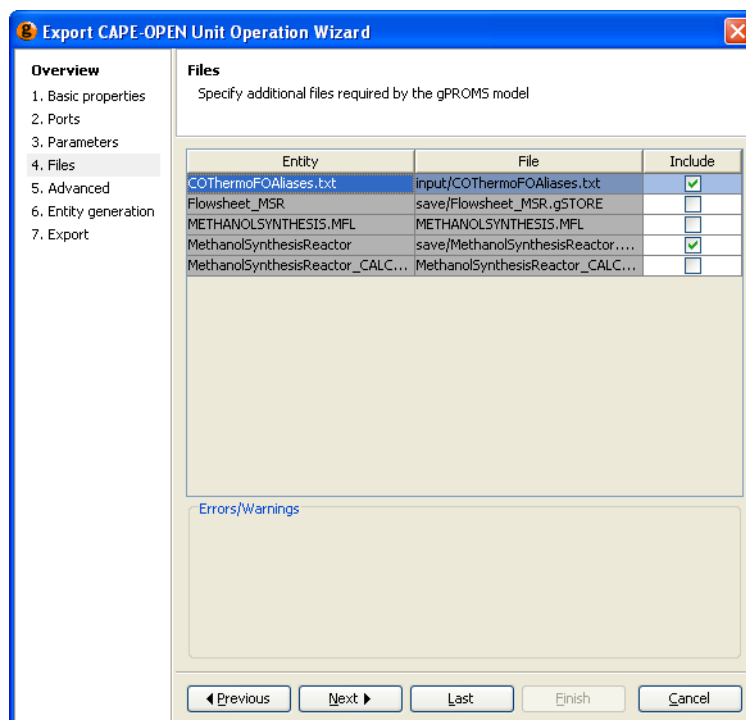
- Each gPROMS Public Model Attribute is mapped to a CAPE-OPEN parameter on the exported unit operation.
- PMAs corresponding to scalar gPROMS VARIABLES and REAL PARAMETERS are mapped to CAPE-OPEN REAL parameters.
- PMAs corresponding to distributed gPROMS variables are supported and mapped to CAPE-OPEN ARRAY of REAL parameters provided that:
  - the distribution is discrete (i.e. a gPROMS ARRAY)
  - the distribution is 1-dimensional
  - the upper bound of the distribution is a constant
- PMAs corresponding to gPROMS INTEGER PARAMETERS are mapped to CAPE-OPEN INT parameters.

- PMAs corresponding to gPROMS LOGICAL PARAMETERS are mapped to CAPE-OPEN BOOLEAN parameters.
- PMAs corresponding to gPROMS SELECTORS are mapped to CAPE-OPEN OPTION parameters.
- models containing PMA's corresponding to gPROMS FOREIGN\_OBJECT PARAMETERS or ORDERED\_SET PARAMETERS cannot be exported.
- PMAs specified as *Obligatory* or *Required - Optional (on)* are mapped to CAPE-OPEN input parameters.
- PMAs specified as *Required - Optional (off)* are mapped to CAPE-OPEN output parameters.
- Where appropriate each parameter's default value, lower bound, upper bound and units of measurement is taken from the PMA definition.
- Each parameter can be provided with a unique name and description that will be used by the CAPE-OPEN interface; the default name and description come from the PMA definition.

## Additional files

If the gPROMS model requires additional files such as a .gSTORE file for initialisation or data files for foreign objects then they can be specified using the 'Files' page of the export wizard.

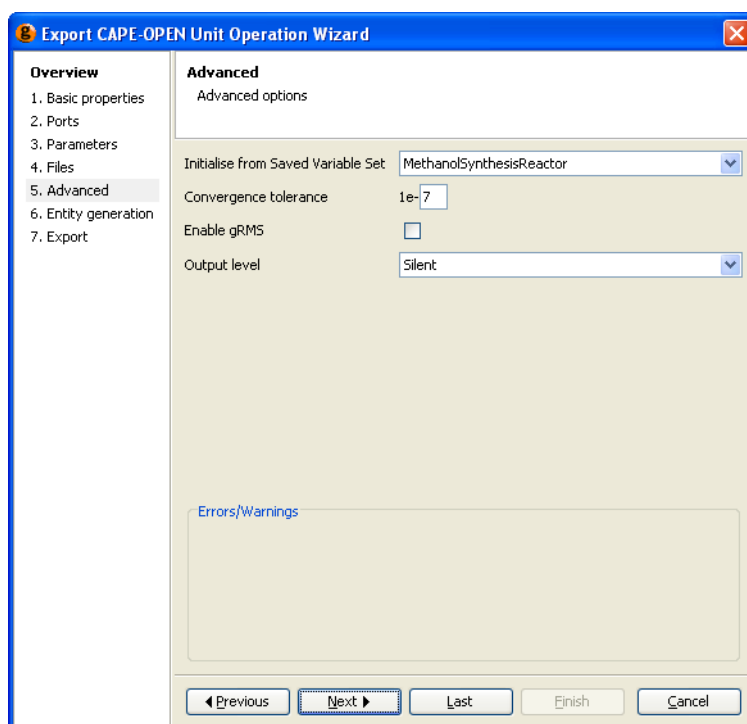
**Figure 7.13. Export CAPE-OPEN Unit Operation Wizard: Additional files page**



- The contents of the list is restricted to those files present in the project containing the gPROMS model being exported. By default all files are marked to be included except those with .gCO, .mfl and .ipp extensions.

## Advanced options

Advanced options for the exported unit operation are specified using the 'Advanced' page of the export wizard.

**Figure 7.14. Export CAPE-OPEN Unit Operation Wizard: Advanced page**

- The *Initialise from Saved Variable Set* option allows you to specify a saved variable set that will be used to initialize the gPROMS model. The list of available saved variable set's is restricted to those included as *Additional files*.
- The *Convergence tolerance* option allows you to change the accuracy with which the gPROMS model will be solved; the value can be between 1e-4 and 1e-10. For stable simulations within a PME you need the gPROMS model solved to a higher accuracy than that with which the PME solves recycle loops, but a lower accuracy than that with which the PME performs physical properties calculations.
- If checked, the *Enable gRMS* option allows the value of all monitored variables in the gPROMS model to be sent to the gRMS application (if running) when the unit operation is calculating. *Note that enabling this option exposes the gPROMS model's unit and variable structure to the end user; you may not wish to do this for commercially sensitive models.*
- The *Output level* option allows you to control what gPROMS execution diagnostics will be visible to the end user of the exported unit operation (in the "execution output" report). This is achieved by setting the top-level "OutputLevel" solution parameter of the generated gPROMS process. *Note that any value other than "Silent" has the potential to expose details of the gPROMS model's implementation; you may not wish to do this for commercially sensitive models.*

## Entity generation options

Exporting a gPROMS model as a CAPE-OPEN unit operation generates 5 entities that are included in your ModelBuilder project:

TEXT FILE <Model Name>.gCO	An XML file containing all the configuration data from the export wizard: <ul style="list-style-type: none"> <li>• when you first drop a gO:CAPE-OPEN unit operation onto a PME flowsheet you are prompted for this configuration file</li> </ul>
PROCESS CAPEOPEN_<Model Name>	A gPROMS PROCESS that is executed by gO:CAPE-OPEN to simulate the unit operation. It's main functions are to:



- initialize the phys\_prop FOREIGN\_OBJECT of each port to use an instance of COTermoFO attached to the material object connected by the PME to that port
- equate the info\_mass\_fraction and mass\_fraction, and the info\_mass\_specific\_enthalpy and mass\_specific\_enthalpy variables of each port
- use the gPROMS simpleEventFOI to initialize all the input parameter values and inlet port properties
- perform a STEADY\_STATE initialisation using the supplied saved variable set (if any)
- launch the CAPE-OPEN communication TASK

TASK CAPEOPEN\_<Model  
Name>

A gPROMS TASK that uses the gPROMS eventFPI to communicate with the PME in an endless loop that:

- GETs all the input parameter values and inlet port properties
- performs a reinitialisation
- SENDs all the output parameter values and outlet port properties

PROCESS Simulate\_<Model  
Name>\_template

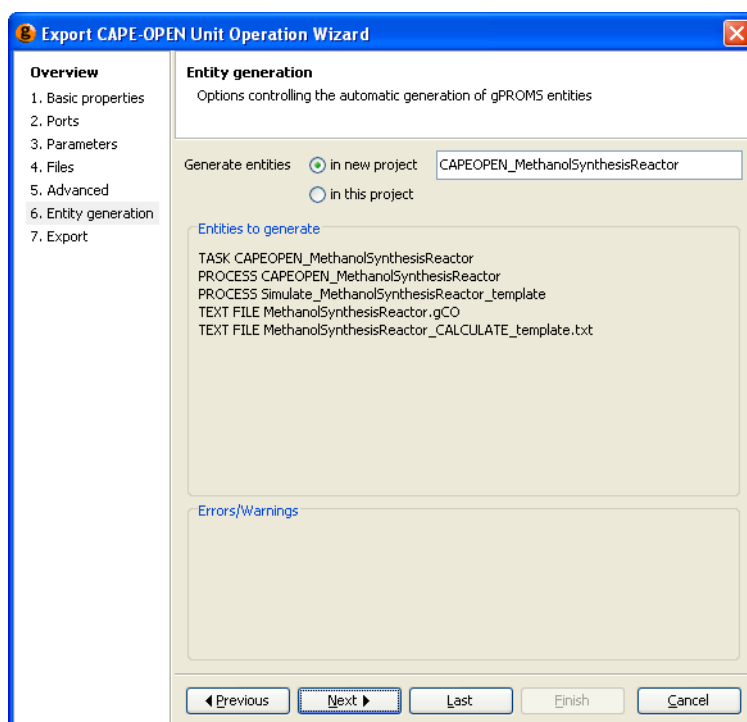
A gPROMS PROCESS that can be used to test the unit operation from within ModelBuilder. This PROCESS initialises all the exported PARAMETERS, SELECTORS and ASSIGNED variables to their default values and then performs a STEADY\_STATE initialisation using the supplied saved variable set (if any). This PROCESS is only a template and requires the following manual additions:

- initialisation strings for each phys\_prop FOREIGN\_OBJECT
- pressure, enthalpy, flow and composition for each inlet stream

TEXT FILE <Model  
Name>\_CALCULATE\_template.txt

A CALCULATE file for simulating the unit operation using the SimplePME utility. This file is only a template and requires the following manual additions:

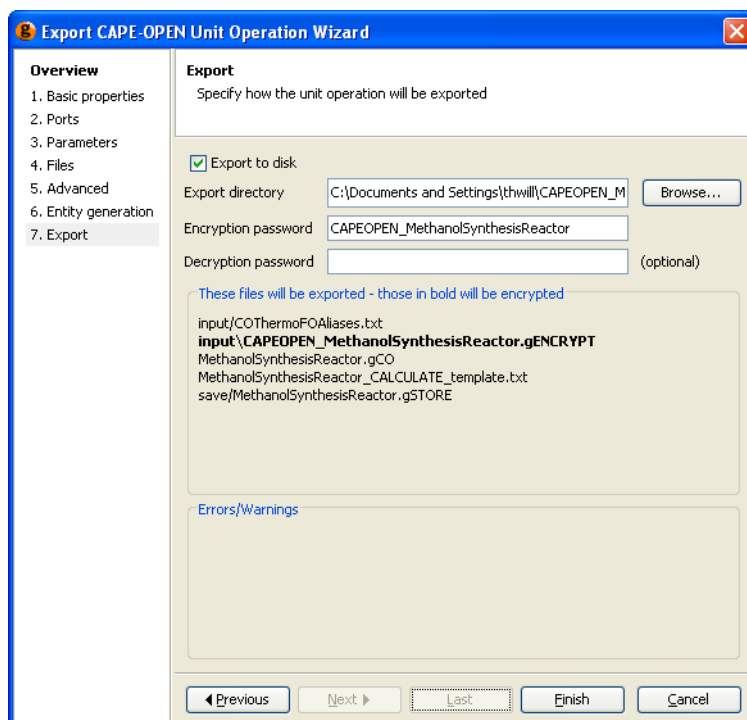
- property system/package specifications for each material port
- pressure, temperature, flow and composition for each inlet stream

**Figure 7.15. Export CAPE-OPEN Unit Operation Wizard: Entity generation page**

- These entities can either be generated in a new project (which will reference the gPROMS model in the existing project) or directly in the existing project.

## Export options

The final page of the wizard allows the destination directory and encryption/decryption passwords to be selected for the exported unit operation.

**Figure 7.16. Export CAPE-OPEN Unit Operation Wizard: Export page**

- The export directory specified will be created if it does not already exist.
- The encryption password is must be specified and will be used to enable the unit operation to be used in a CAPE-OPEN application.
- The decryption password is optional. If provided, the exported unit operation can be decrypted so that it can be opened and modified by ModelBuilder; if not, then the contents of the unit operation are permanently hidden and uneditable.
- The encryption and decryption passwords must be different and of at least 6 characters in length.
- The export process will not overwrite any files without first prompting.

## The Simple Process Modelling Environment

To help users and PSE support staff diagnose interoperability issues between gPROMS and other CAPE-OPEN compliant packages the Windows version of gPROMS comes with a command-line utility called SimplePME.exe.

- Listing available CAPE-OPEN unit operations
- Listing available CAPE-OPEN physical properties packages
- Performing a calculation on a CAPE-OPEN unit operation
- SimplePME command syntax

### Listing available CAPE-OPEN unit operations

To list all the available CAPE-OPEN unit operations registered on your computer execute **SimplePME.exe** at a command prompt.

The exact list displayed will depend on which CAPE-OPEN components are installed, but should look something like:

**Figure 7.17. Example 'unit operation list' output from SimplePME**

```
UNIT OPERATIONS
-----
0) APECS.UnitOperation.1
1) AspenCOUnit100.MixNSplit
2) COCO_COUS.CSTR.1
3) COCO_COUS.CompoundSplitter.1
...
18) COCO_COUS.Valve.1
19) ChemSepUO.ChemSep_UnitOperation.1
20) PSEUnitLibrary.EnhancedMixSplit.1
21) PSEUnitLibrary.MixSplit.1
22) PSEUnitLibrary.TestUnit.1
23) PSEUnitLibrary.gOCAPEOPEN.1

Select a unit operation (0-23):
```

Selecting a unit operation from this list (by typing the corresponding number) will cause an instance of the unit operation to be initialized and queried about its ports, parameters, report, etc. The output should look something like:

**Figure 7.18. Example 'initialization' output from SimplePME**

INITIALIZE\_RESPONSE

```
PROGID      "PSEUnitLibrary.MixSplit.1"
# CLSID     "{B12E80B6-5FBD-48AE-8AC9-FE21ADA8683F}"
NAME        "PSEMixSplit"
DESCRIPTION  "Standalone Mixer-Splitter unit op. from PSE Ltd."

#
# Parameters
#

INPUT       "BASIS"
DESCRIPTION  "Basis for thermodynamic calculations"
OPTION      "mole" : "mole", "mass"

INPUT       "MO_DIAGNOSTICS"
DESCRIPTION  "Include interactions with material objects in unit diagnostics?"
BOOLEAN     FALSE

INPUT       "SPLIT_FRACTION"
DESCRIPTION  "Fraction of flow going to first product stream"
REAL        0 : 0 : 1

INPUT       "HEAT_INPUT"
DESCRIPTION  "HEAT_INPUT"
REAL        0 : -1e+007 : 1e+007

OUTPUT      "TOTAL_FLOW"
DESCRIPTION  "Total molar flow through the unit"
REAL        0 : 0 : 1e+007

#
# Ports
#

INLET       "FEED_1"
DESCRIPTION  "First feed stream"
MATERIAL

INLET       "FEED_2"
DESCRIPTION  "Second feed stream"
MATERIAL

OUTLET      "PRODUCT_1"
DESCRIPTION  "First product stream"
MATERIAL

OUTLET      "PRODUCT_2"
DESCRIPTION  "Second product stream"
MATERIAL

#
# Reports:
#   summary
#   debug log
#
```

## Listing available CAPE-OPEN physical properties packages

To list all the available CAPE-OPEN thermodynamic property packages and systems registered on your computer execute **SimplePME.exe -pp** at a command prompt.

The exact list displayed will depend on which CAPE-OPEN components are installed, but should look something like:

**Figure 7.19. Example 'property package list' output from SimplePME**

## PROPERTY PACKAGES

```

-----
0) ProgID:      PPDS.CapeSteamPackage.1
   CLSID:       {8E9B4FC1-439C-11D5-8E2D-00D0590F7D4D}
   Name:        PPDS CO Steam Package
   Description: PPDS CO Package for properties of steam (IAPS84)

```

## PROPERTY SYSTEMS

```

-----
0) ProgID:      PPDS.CapeThermoSystem.1
   CLSID:       {032F7643-2F57-11D5-B7A8-0000E812B8B1}
   Name:        PPDS CO ThermoSystem
   Description:  PPDS set of CO Packages
   Packages:    Hydrocarbon_test_package
                Chemical_test_package
                INDISS_test_package
                MethanolSynthesis

1) ProgID:      OATS.ThermoSystem.1
   CLSID:       {4CCF55DB-E332-42F8-B685-188989F8E1EC}
   Name:        OATS (CAPE-OPEN 1.0)
   Description:  Out-of-proc Application Thermo Server: Thermo System
   Packages:    Multiflash Thermo System/METHANOLSYNTHESIS

2) ProgID:      MFCOThermoSys.MFCOSys.1
   CLSID:       {653CE81C-DAD9-434B-B878-D5947CB16AD4}
   Name:        Multiflash Thermo System
   Description:  Multiflash CAPE-OPEN v1.0 Thermo System
   Packages:    Air
                BENZENEWATER
                flash
                FluentCSTR
                FuelAir
                HEXENE
                HIDIC
                METHANOLSYNTHESIS
                METHANOLWATER
                WATER

3) ProgID:      COCO_TEA.ThermoPack.1
   CLSID:       {90DAC7FA-E0E4-40B5-A903-E0B12774D52B}
   Name:        TEA (CAPE-OPEN 1.0)
   Description:  COCO Thermodynamics for Engineering Applications
   Packages:    C1_C2
                C1_C2 (EOS)
                n-depropropanizer
                alkanes
                HDA
                Water-nButanol-UNIQUEAC
                MethanolSynthesis
                MethanolWater

```

## Performing a calculation on a CAPE-OPEN unit operation

To perform a calculation on a registered CAPE-OPEN unit operation execute **SimplePME.exe** -in *MyCalcFile.txt* at a command prompt, where *MyCalcFile.txt* is a formatted text file describing the input values to use for the calculation and the output values to include in the results output, e.g.

### Example 7.1. Example 'calculation' input file for SimplePME

```
# Perform a calculation
CALCULATE ❶

# PROGID of the unit operation to perform a calculation on
PROGID "PSEUnitLibrary.MixSplit.1" ❷

# Inputs and outputs should be on mole basis
MOLE ❸

# Set a value for the "SPLIT_FRACTION" input parameter
INPUT "SPLIT_FRACTION" ❹
0.3

# Set a value for the "HEAT_INPUT" input parameter
INPUT "HEAT_INPUT"
0.0

# Include "TOTAL_FLOW" output parameter in the results
OUTPUT "TOTAL_FLOW" ❺

# Attach a material object/stream to the "FEED_1" inlet port
INLET      "FEED_1" ❻
MATERIAL   "MFCOThermoSys.MFCOSys<METHANOLWATER.MFL>"
PRESSURE   100005
TEMPERATURE 360
FLOW        50
FRACTION    [0.3,0.7]

# Attach a material object/stream to the "PRODUCT_1" outlet port
OUTLET     "PRODUCT_1" ❼
MATERIAL   "MFCOThermoSys.MFCOSys<METHANOLWATER.MFL>"

# Attach a material object/stream to the "PRODUCT_2" outlet port
OUTLET     "PRODUCT_2"
MATERIAL   "MFCOThermoSys.MFCOSys<METHANOLWATER.MFL>"
```

- ❶ The first line of the file (other than comments and whitespace) is used to indicate the operation to perform on the unit operation:
- INITIALIZE to perform an initialisation, in which case only the PROGID specification is relevant
  - VALIDATE to perform a validation, in which case OUTPUT parameter specifications have no effect
  - CALCULATE to perform a calculation
- ❷ The PROGID specification is used to indicate which unit operation to perform a calculation on. This is optional; if not present then you will be prompted to specify a unit operation
- ❸ The MASS or MOLE specification is used to indicate the basis on which the properties for material streams are provided/displayed
- ❹ An INPUT specification is used to provide a value for an INPUT parameter, any parameters which are not specified will take their default value. The format of the specification depends on the type of the parameter:

- for scalar REAL parameters:

```
INPUT "MyRealParameter"  
3.14
```

- for scalar INTEGER parameters:

```
INPUT "MyIntegerParameter"  
5
```

- for scalar BOOLEAN parameters:

```
INPUT "MyBooleanParameter" TRUE
```

or:

```
INPUT "MyBooleanParameter" FALSE
```

- for scalar OPTION parameters:

```
INPUT "MyOptionParameter"  
"Blue"
```

- for 1D ARRAY<sup>8</sup> of REAL parameters:

```
INPUT "MyRealArrayParameter"  
REAL [0.1, 0.2, 0.3, 0.4, 0.5]
```

- for 1D ARRAY of INTEGER parameters:

```
INPUT "MyIntegerArrayParameter"  
INTEGER [1, 2, 3, 4, 5]
```

- for 1D ARRAY of BOOLEAN parameters:

```
INPUT "MyBooleanArrayParameter"  
BOOLEAN [TRUE, TRUE, FALSE, TRUE, FALSE]
```

- for 1D ARRAY of OPTION parameters:

```
INPUT "MyOptionArrayParameter"  
OPTION ["Red", "Yellow", "Blue", "Green", "Orange"]
```

- ⑤ An OUTPUT specification identifies an OUTPUT parameter which should be included in the results output
- ⑥ An INLET specification identifies a material object/stream connected to an INLET port. INLET ports without a corresponding specification are considered to be unconnected, which depending on the unit operation may not be valid;

A MATERIAL specification is used to indicate the thermodynamic property package/system to be used for the connected material object/stream. The specification can be made in 2 different ways:

- for a named package within a ThermoSystem:

```
MATERIAL "ThermoSystemProgId<ThermoPackageName>"
```

- for a standalone ThermoPackage:

```
MATERIAL "ThermoPackageProgId"
```

---

<sup>8</sup>The SimplePME does not support multi-dimensional array parameters, or arrays of heterogeneous elements.  
The properties of an INLET stream can be specified in 5 different ways:



- T, P, total flow, composition:

TEMPERATURE	360	# K
PRESSURE	100005	# Pa
FLOW	50	# mol/s
FRACTION	[0.3, 0.7]	# mol/mol

- T, vapor phase fraction, total flow, composition:

TEMPERATURE	360	# K
VAPORFRACTION	0.585	# mol/mol
FLOW	50	# mol/s
FRACTION	[0.3, 0.7]	# mol/mol

- P, H, total flow, composition:

PRESSURE	100005	# K
ENTHALPY	-15672.1	# J/mol
FLOW	50	# mol/s
FRACTION	[0.3, 0.7]	# mol/mol

- P, T, total flow, composition:

PRESSURE	100005	# Pa
TEMPERATURE	360	# K
FLOW	50	# mol/s
FRACTION	[0.3, 0.7]	# mol/mol

- P, vapor phase fraction, total flow, composition:

PRESSURE	100005	# Pa
VAPORFRACTION	0.585	# mol/mol
FLOW	50	# mol/s
FRACTION	[0.3, 0.7]	# mol/mol

- ⑦ An OUTLET specification identifies a material object/stream connected to an OUTLET port. OUTLET ports without a corresponding specification are considered to be unconnected, which depending on the unit operation may not be valid. Each OUTLET specification must be accompanied by a MATERIAL specification, but obviously there is no need to specify P,T, H, flow, fraction for an OUTLET

The output from performing a calculation obviously depends on the unit operation and the values provided for the INPUTs and INLETs, but should look something like:

**Example 7.2. Example 'calculation' output from SimplePME**

```
CALCULATE_RESPONSE
```

```
MOLE
```

```
OUTPUT "TOTAL_FLOW"
```

```
50
```

```
OUTLET      "PRODUCT_1"
MATERIAL     "MFCOThermoSys.MFCOSys<METHANOLWATER.MFL>"
PRESSURE     100005
ENTHALPY     -14510.2
TEMPERATURE  360
VAPORFRACTION 0.585425
FLOW         15
FRACTION     [0.3, 0.7]
```

```
OUTLET      "PRODUCT_2"
MATERIAL     "MFCOThermoSys.MFCOSys<METHANOLWATER.MFL>"
PRESSURE     100005
ENTHALPY     -14510.2
TEMPERATURE  360
VAPORFRACTION 0.585425
FLOW         35
FRACTION     [0.3, 0.7]
```

```
#
# Reports:
#   summary
#   debug log
#
```

**SimplePME command syntax**

```
SimplePME.exe -help
```

Display this description of the SimplePME command syntax

```
SimplePME.exe -pp
```

Output a list of the registered CAPE-OPEN property packages/systems

```
SimplePME.exe [-edit1] -gproms [progID]
```

Generate '.counitfo.cfg' and '.model.gPROMS' files for using the specified unit with the *prototype* COUnitFO

```
SimplePME.exe [-edit1] [-edit2] [-edit3] [-proxy] [-report] [-time] [-verbose] [-log
logFile] [-in inFile] [progID]
```

Parse a file to initialise, validate or calculate a CAPE-OPEN unit operation

-edit1	show the unit's <i>Edit Dialog</i> just <i>after</i> initialisation
-edit2	show the unit's <i>Edit Dialog</i> just <i>before</i> validation
-edit3	show the unit's <i>Edit Dialog</i> just <i>before</i> termination
-proxy	generate output for parsing by the COUnitProxy. <i>This option is documented purely for completeness, it is not of any use to end users</i>
-report	generate all the unit's available reports just before termination. Even if the unit defines no reports of its own the SimplePME can automatically generate a <i>summary</i> report

<code>-time</code>	output the time taken to perform the initialisation, validation or calculation
<code>-verbose</code>	include CLSID, name and description when listing registered units
<code>-log <i>logFile</i></code>	enable logging to the named file. A value of 1 sends the log to STDOUT, whilst a value of 0 (the default) causes no log to be generated
<code>-in <i>inFile</i></code>	read from <i>inFile</i> to determine the action to perform on the unit. If this is not specified then the behaviour is the same as if the file contained just the text INITIALIZE
<code><i>progID</i></code>	the ProgID of the unit to act on; this overrides a PROGID specification in <i>inFile</i> . If a ProgID is not specified in <i>inFile</i> or via this option then the user is prompted to choose from a list of the registered units

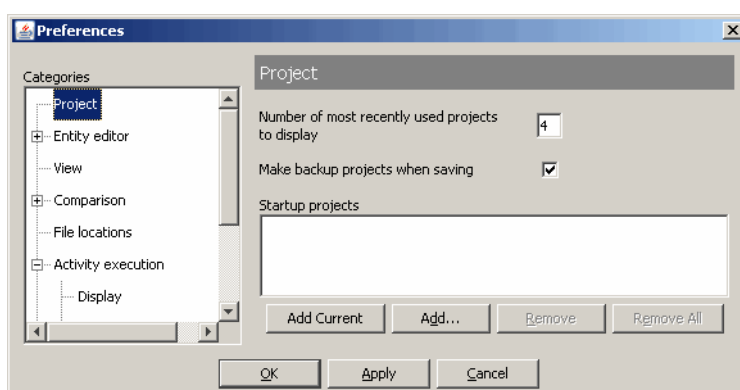
# Chapter 8. Miscellaneous Utilities

A range of other features increase the functionality of gPROMS ModelBuilder. These features include standard utilities such as printing *capabilities* and window layouts. Also, ModelBuilder has a number of shortcut keys, many of which are the same as standard shortcut keys plus some ModelBuilder specific ones.

## ModelBuilder Preferences

Many of ModelBuilder's features can be customised by the user. The ModelBuilder preferences dialog is opened from the Edit menu. There are seven preference categories, the main options for each are highlighted in the table below.

**Figure 8.1. ModelBuilder Preference dialog**



**Table 8.1. ModelBuilder preferences**

Categories	Option
Project	<ul style="list-style-type: none"><li>Specify the number of recently opened projects to appear at the bottom of the Project menu</li><li>Choose whether to make backup projects when saving or not</li><li>Specify any projects to be opened on start-up</li></ul>
Entity editor	<ul style="list-style-type: none"><li>Specify the general text editor settings, such as font properties, tab size, thresholds for displaying numbers in scientific notation</li><li>Customise the syntax highlighting attributes</li><li>Customise the settings for auto-completion in gPROMS language</li><li>Configure middle-mouse button settings</li></ul>
View	<ul style="list-style-type: none"><li>Choose the default setting for the editor mode (single or multiple editors)</li><li>Specify the look and feel of gPROMS ModelBuilder. Three options are available: Metal, Windows<sup>a</sup> and Motif</li><li>ModelBuilder messages may have been set not to be shown again, but this can be reset</li></ul>

	<ul style="list-style-type: none"> <li>• Configure if the default text size should be overridden</li> <li>• Specify the text size to be displayed in the Project tree</li> </ul>
Comparison	<ul style="list-style-type: none"> <li>• Specify the colours used for text and group comparison</li> </ul>
File locations	<ul style="list-style-type: none"> <li>• Specify the location for the temporary directory used by ModelBuilder. This is the temporary execution directory used to run model based activities</li> <li>• Specify the default starting location for the file dialog used for opening and saving projects</li> <li>• Specify a search path for any software components used by gPROMS, e.g. foreign objects, foreign processes etc.</li> </ul>
Activity execution	<ul style="list-style-type: none"> <li>• Specify whether to delete previous Cases from the same processes when executing an activity</li> <li>• Configure the look and feel of the execution output window</li> <li>• Configure the default settings for the contents of a Case</li> <li>• Choose whether or not to show data points on graphs</li> <li>• Set graphs to low-resolution mode</li> <li>• Set the default number format</li> </ul>
Flowsheeting	<ul style="list-style-type: none"> <li>• Specify the font size to be displayed in the flowsheeting tab</li> <li>• Specify the grid size when snapping object to grid</li> <li>• Choose to scale the flowsheet to fit the window whenever it is resized</li> </ul>
Schedule	<ul style="list-style-type: none"> <li>• Specify the default width of schedule elements</li> </ul>
Advanced	<ul style="list-style-type: none"> <li>• Specify whether to display the UMS text editor which offers direct access to XML code for the design of Specification dialogs</li> </ul>

<sup>a</sup>On the Windows platform, this is the default for fresh installations. It is recommended because it offers a better integration with the operating system, for instance the file dialogs.

## Number formats

gPROMS can display numbers in a variety of formats according to the specification in the Model Builder preferences. There are three types of number format: General, Fixed and Scientific.

The general format will display a number depending on its value. Normally it displays the specified significant figures without an exponent. However, an exponent is used if the exponent from its conversion is less than -4 or greater than or equal to the number of significant figures. The number of significant figures can be specified from between 1 and 6.

The fixed number format displays numbers using the specified number of decimal places, which can be set to any value between 0 and 9 inclusive. However, if more than 6 significant figures are required to display the number, then scientific notation is used instead.

The scientific number format always displays numbers using the "E" notation<sup>1</sup> and uses the number of decimal places specified, which can be from 0 to 5.

The number format for Value Tables in Case Topologies and in PMA Tables in Model Reports is specified by opening the Model Builder preferences (by selecting ModelBuilder Preferences... from the Edit menu) and choosing the desired format from the listbox in the Results section of the Activity execution preferences.

In addition to the above formatting options, one may also set the thresholds that determine whether a number is formatted using the general or scientific formats. These are set in the General appearance section of the Entity editor preferences. If a number is smaller than the lower threshold or larger than the upper threshold, then it will be displayed in scientific format. This applies to the Variable Types Table (in the gPROMS Project), in Variable Results Tables in Case Projects, the Measurements tab of a Parameter Estimation Report (also in Case Projects) and also to the legend labels of a 2D plot (Case Project: in reports and topologies).

## Text editor short-cut keys

A number of useful text editor short-cut keys have been defined in gPROMS ModelBuilder. Most commonly used short-cuts include: **F4** to check syntax and **F5** to simulate a PROCESS. This section lists the short-cut keys available in ModelBuilder and the associated action.

'Current line' refers to the line that the cursor/caret is on.

A '+' sign means hold the two keys either side of it down simultaneously. A ',' separates key presses, e.g. **Alt + U**, **U** means press the alt and U keys together, and then press the U key on its own.

## Navigation shortcuts

**Table 8.2. Shortcuts with standard navigation keys**

Keyboard shortcut	With no selected text	With selected text
<b>RIGHT</b>	Move the insertion point one character to the right.	Deselect the text and move the insertion point one character to the right.
<b>LEFT</b>	Move the insertion point one character to the left.	Deselect the text and move the insertion point one character to the left.
<b>DOWN</b>	Move the insertion point to the next line.	Deselect the text and move the insertion point to the next line.
<b>UP</b>	Move the insertion point to the previous line.	Deselect the text and move the insertion point to the previous line.
<b>SHIFT+RIGHT</b>	Select the character to the right of the insertion point.	Extend the selection one character to the right.
<b>SHIFT+LEFT</b>	Select the character to the left of the insertion point.	Extend the selection one character to the left.
<b>SHIFT+DOWN</b>	Create a text selection and extend it to the next line.	Extend the selection to the next line.
<b>SHIFT+UP</b>	Create a text selection and extend it to the previous line.	Extend the selection to the previous line.

<sup>1</sup>where  $1.23 \times 10^4$  is written 1.23E+04.

<b>CTRL+RIGHT</b>	Move the insertion point one word to the right.	Deselect the text and move the insertion point one word to the right.
<b>CTRL+LEFT</b>	Move the insertion point one word to the left.	Deselect the text and move the insertion point one word to the left.
<b>CTRL+SHIFT+RIGHT</b>	Create a text selection and extend it one word to the right.	Extend the selection one word to the right.
<b>CTRL+SHIFT+LEFT</b>	Create a text selection and extend it one word to the left.	Extend the selection one word to the left.
<b>PgDown</b>	Move the insertion point one page down.	Deselect the text and move the insertion point one page down.
<b>PgUp</b>	Move the insertion point one page up.	Deselect the text and move the insertion point one page up.
<b>SHIFT+ PgDown</b>	Create a text selection and extend it one page down.	Extend the selection one page down.
<b>SHIFT+ PgDown</b>	Create a text selection and extend it one page down.	Extend the selection one page down.
<b>SHIFT+PgUp</b>	Create a text selection and extend it one page up.	Extend the selection one page up.
<b>HOME</b>	Move the insertion point to the beginning of the line.	Deselect the text and move the insertion point to the beginning of the line.
<b>END</b>	Move the insertion point to the end of the line.	Deselect the text and move the insertion point to the end of the line.
<b>SHIFT+HOME</b>	Create a text selection and extend it to the beginning of the line.	Extend the selection to the beginning of the line.
<b>SHIFT+END</b>	Create a text selection and extend it to the end of the line.	Extend the selection to the end of the line.
<b>CTRL+HOME</b>	Move the insertion point to the beginning of the document.	Deselect the text and move the insertion point to the beginning of the document.
<b>CTRL+END</b>	Move the insertion point to the end of the document.	Deselect the text and move the insertion point to the end of the document.
<b>CTRL+SHIFT+HOME</b>	Create a text selection and extend it to the beginning of the document.	Extend the selection to the beginning of the document.
<b>CTRL+SHIFT+END</b>	Create a text selection and extend it to the end of the document.	Extend the selection to the end of the document.
<b>ALT+u , e</b>	Move the insertion point to the end of the current word.	Move the insertion point to the end of the last word in the selection.

## Navigation shortcuts - Location shortcuts

**Table 8.3. Insertion point/screen position shortcuts**

Keyboard shortcut	Action
<b>CTRL+UP</b>	Scroll the text one line up while holding the insertion point in the same position.
<b>CTRL+DOWN</b>	Scroll the text one line down while holding the insertion point in the same position.

<b>ALT+u , t</b>	Scroll the text up so that the insertion point moves to the top of the window while remaining at the same point in the text.
<b>ALT+u , m</b>	Scroll the text so that the insertion point moves to the middle of the window while remaining at the same point in the text.
<b>ALT+u , b</b>	Scroll the text down so that the insertion point moves to the bottom of the window while remaining at the same point in the text.
<b>SHIFT+ALT+t</b>	Move the insertion point to the top of the window.
<b>SHIFT+ALT+m</b>	Move the insertion point to the middle of the window.
<b>SHIFT+ALT+b</b>	Move the insertion point to the bottom of the window.

## Navigation shortcuts - Jump list shortcuts

**Table 8.4. Jump list shortcuts**

Keyboard shortcut	Action
<b>ALT+k</b>	Go to previous entry in the jump list.
<b>ALT+l</b>	Go to next entry in the jump list.
<b>SHIFT+ALT+k</b>	Go to the previous jump list entry not in the same file.
<b>SHIFT+ALT+l</b>	Go to the next jump list entry not in the same file.

## Navigation shortcuts - Miscellaneous

**Table 8.5. Miscellaneous navigation shortcuts**

Keyboard shortcut	Action
<b>F2</b>	Go to next bookmark.
<b>CTRL+F2</b>	Toggle bookmark.
<b>CTRL+[</b>	Find matching bracket.
<b>CTRL+SHIFT+[</b>	Select block between current bracket and matching one.

## Navigation shortcuts - Find shortcuts

**Table 8.6. Find shortcuts**

Keyboard shortcut	With no selected text	With selected text
<b>CTRL+f</b>	Show Find dialog.	Show Find dialog and show selected text as the text to find.
<b>F3</b>	Search for the next occurrence.	
<b>SHIFT+F3</b>	Search for the previous occurrence.	
<b>CTRL+F3</b>	Search for the next occurrence of the word that the insertion point is on.	Search for the next occurrence of the selected text.
<b>ALT+SHIFT+h</b>	Switch highlight search on or off.	
<b>CTRL+g</b>	Show Goto Line dialog.	



## Edit shortcuts

**Table 8.7. Shortcuts with standard edit keys**

Keyboard shortcut	Action
<b>INSERT</b>	Switch between insert mode and overwrite mode.
<b>CTRL+a</b>	Select all.
<b>CTRL+z</b>	Undo the previous command.
<b>CTRL+y</b>	Redo the undone command.
<b>CTRL+x</b>	Delete the selected text from the file and copy it the clipboard.
<b>SHIFT+DELETE</b>	Delete the selected text from the file and copy it the clipboard.
<b>CTRL+c</b>	Copy the selected text to the clipboard.
<b>CTRL+INSERT</b>	Copy the selected text to the clipboard.
<b>CTRL+v</b>	Insert the clipboard text into the file.
<b>SHIFT+INSERT</b>	Insert the clipboard text into the file.
<b>CTRL+e</b>	Remove the current line.
<b>CTRL+u</b>	Delete text in the following cycle (when using the shortcut successive times): first the text preceding the insertion point on the same line, then the indentation on the line, then the line break, then the text on the previous line, and so on.
<b>CTRL+w</b>	Remove the current word or the word preceding the insertion point.
<b>CTRL+k</b>	Word Match - find the previous word that begins like the current word and complete the current word so that they match.
<b>CTRL+l</b>	Word Match - find the next word that begins like the current word and complete the current word so that they match.
<b>DELETE</b>	Remove character after the insertion point.
<b>SHIFT+SPACE</b>	Insert space without expanding abbreviation.
<b>ALT+j</b>	Select the word the insertion point is on or deselect any selected text.

## Edit shortcuts - Indentation shortcuts

**Table 8.8. Indentation shortcuts**

Keyboard shortcut	With no selected text	With selected text
<b>TAB</b>	Insert tab	Shift selection right
<b>SHIFT+TAB</b>		Shift selection left
<b>CTRL+t</b>	Shift line right	Shift selection right
<b>CTRL+d</b>	Shift line left	Shift selection left

## Edit shortcuts - Capitalization shortcuts

**Table 8.9. Capitalization shortcuts**

Keyboard shortcut	With no selected text	With selected text
<b>ALT+u , u</b>	Make the character after the insertion point uppercase	Make the selection uppercase
<b>ALT+u , l</b>	Make the character after the insertion point lowercase	Make the selection lowercase
<b>ALT+u , r</b>	Reverse the case of the character after the insertion point	Reverse the case of the selected text

## Editing using external editor software

Right clicking on the selected Saved Variable Sets file and subsequently selecting Edit using external program... will bring up an application chooser window for the user to select the .exe file of the desired editor. The file can then be modified in the chosen editor and saved again to be used by gPROMS (please note that the user needs to close the editor application in order to return to ModelBuilder).

## Printing

Various aspects of a gPROMS project may be printed, from the gPROMS language itself (of Models, Tasks, Processes etc.) to Model Topologies, Graphs and so on. The following entities can be selected for printing:

- Entities in a gPROMS Project
  - Variable Types
  - Stream Types
  - Models
    - gPROMS language
    - Topology
  - Tasks
  - Processes
  - Optimisations
  - Parameter Estimations
  - Experiment Designs
  - Performed Experiments
  - Experiments to be Designed
- Entities in a Case
  - Original Entities (as above)
  - Trajectories
    - Topologies

- Reports
- Graphs
- Problem Description (the gPROMS listing for the whole problem)

There are two items on the File menu that are used for printing; these are:

- Page Setup...
- Print...

The Page Setup... menu item allows you to specify how the printed entities will appear on the page. As well as setting the paper size, orientation and margins, you can also specify headers and how Topologies are printed.

The Print... menu item opens a Print Preview dialog that lets you select which entities to print. The Print Preview dialog has a tree view on the left, showing all of the Projects and Cases open in ModelBuilder. Each entity has a checkbox that allows you to select which are printed. The right-hand side of the Print Preview dialog displays a preview of the printed output.

When the Print Preview dialog is activated, some entities will be selected automatically. Before describing the details of the Initial Print Selection, two useful situations are described below.

A whole Project (or Case) may be printed by selecting the Project (or Case) and using the Print... item on the File menu.

It is also possible to print a single Model Topology or Graph.

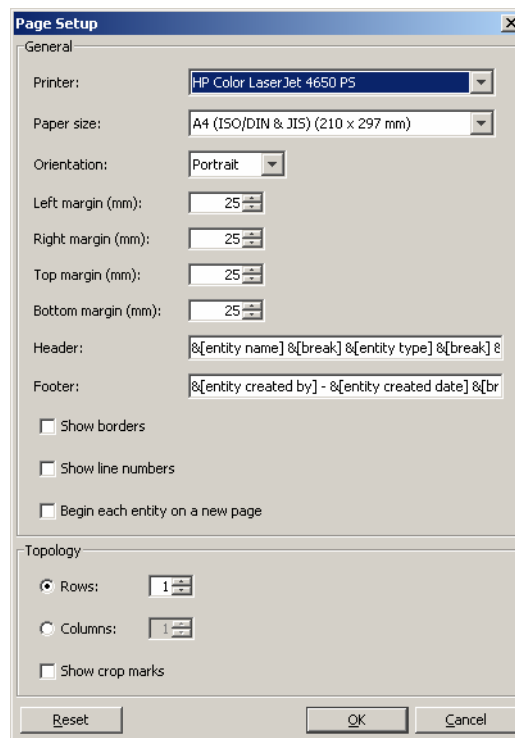
- To print a Topology
  - Double click on a Model (either in a Project or Case) and select the Topology tab.
  - Right click on the Topology to activate the context menu and select Print....
- To print a Graph
  - Double click on a Variable and select the Graph tab.
  - Right click on the Graph to activate the context menu and select Print....

Doing either of these will activate the Print Preview dialog with only that entity selected. From there, you can select further entities for printing, or just print the one selected. The Page Setup dialog is also accessible from Print Preview.

## The Page Setup Dialog

Left-clicking on the File menu and selecting Page Setup... activates the dialog shown below.

### Figure 8.2. Page Setup Dialog



The dialog is partitioned into two sections along with three command buttons. The first part sets parameters that affect all printed objects; the second concerns only the printing of Model Topologies. Each control on the Page Setup dialog is described below.

The General Page Setup controls are:

- Printer
- Paper size
- Paper orientation
- Left, Right, Top and Bottom margins
- Header
- Footer
- Show borders
- Show line numbers
- Begin each entity on a new page

Most of the controls are quite straight-forward and are found as standard in many applications. The last five merit some explanation.

Header and footer text can be provided by entering values in the two text controls labelled Header and Footer. Values entered into these boxes can be a combination of either plain text or tags. Plain text entries will be printed directly on the head or foot of each page. Tags allow project-specific parameters to be printed, such as the page number, the date and a number parameters found in the Project and Entity Properties sections. A full list of tags is given below.

&[break]	Creates a space between groups of text.
----------	---

- With no `&[break]` tags, all text is printed flush left.
- With one `&[break]` tag, the text on the left is printed flush left and the text to the right of the tag is printed flush right.
- A second `&[break]` tag causes the second group of text to be centred on the page.
- Adding further tags causes each group of text to be spaced out evenly on the page. However, as the text is always printed on one line, too many groups can cause some text to be cropped.

<code>&amp;[page]</code>	Prints the current page number.
<code>&amp;[pages]</code>	Prints the total number of pages being printed.
<code>&amp;[short date]</code>	Prints the current date in short-date format, e.g. 28/08/07.
<code>&amp;[long date]</code>	Prints the current date in long-date format, e.g. 28 August 2007.
<code>&amp;[time]</code>	Prints the current time.
<code>&amp;[project name]</code>	The name of the gPROMS Project (i.e. its filename minus the .gPJ extension).
<code>&amp;[project url]</code>	The location of the gPROMS Project in URL format.
<code>&amp;[project created by]</code>	The user id of the person who created the Project.
<code>&amp;[project created date]</code>	The date the Project was created.
<code>&amp;[project modified by]</code>	The user id of the person who last modified the Project.
<code>&amp;[project modified date]</code>	The date the Project was last modified.
<code>&amp;[entity name]</code>	The name of the current entity being printed.
<code>&amp;[entity type]</code>	The type of entity being printed.
<code>&amp;[entity view]</code>	The entity view: i.e. Topology, gPROMS Language etc.
<code>&amp;[entity created by]</code>	The user id of the person who created the entity.
<code>&amp;[entity created date]</code>	The date the entity was created (in short-date format).
<code>&amp;[entity modified by]</code>	The user id of the person who last modified the entity.
<code>&amp;[entity modified date]</code>	The date the entity was last modified (in short-date format).

The default header text is `&[project name] - created by &[project created by] on &[project created date]&[break]Page &[page] of &[pages]`. The footer is empty by default.

The Show borders check box enables borders around each page, at the margin boundaries.

The Show line numbers check box enables printing of gPROMS language to be accompanied by line numbers.

The Begin each entity on a new page check box forces each entity to begin on a new page. By default, each entity is printed directly after the last one, which is more economical on paper usage, particularly for small entities.

The Topology Page Setup is applied only to Topology entities, i.e. flowsheets. For large flowsheets, many smaller details may not be clear if the whole flowsheet is printed on a single page, which is the default. The controls in this section of the dialog therefore allow one to specify that Topology entities should be printed over several pages, and how this should be done.

As the aspect ratio of the flowsheet is always maintained, one need only specify how many pages should be used either horizontally or vertically. This is done by selecting one of the two radio buttons labelled Rows or Columns. Once selected, you may then enter the number of pages in the box to the right. For example, if you were to select the Rows button and enter 2 in the box to the right, gPROMS will enlarge the Topology so that it will fit on two pages horizontally. The number of pages used vertically then depends only on the shape of the Topology.

The final control in this section is a checkbox to enable crop marks. These are guides printed in the corners of each page to aid alignment in a guillotine, should one wish to tape each page together to form a complete diagram. As with the page borders, the page margins are used for the placement of the crop marks, so if the pages are to be cut with scissors then page borders may be a more suitable alternative. This, however, would affect all printed pages; not just the Topologies.

The figure here illustrates a Topology printed over multiple pages (in this case, 2 rows) with crop marks enabled.

The command buttons at the bottom of the Page Setup dialog are:

- Reset

Reverts all settings to the defaults, including any changes previously saved by pressing OK.

- OK

Closes the dialog and accepts all changes made to the values in the controls.

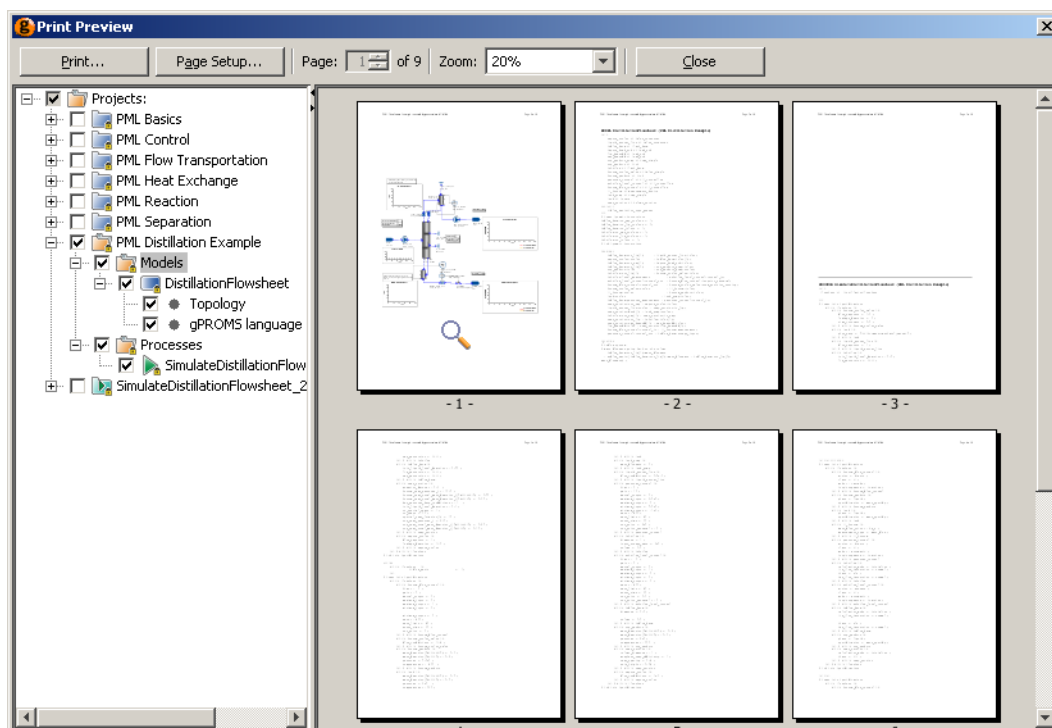
- Cancel

Closes the dialog, abandoning any changes made.

## Print Preview

The Print Preview dialog lets you select which parts of gPROMS Projects or Cases to print. Once Print... is selected from the File menu, a dialog similar to the one below will appear.

**Figure 8.3. The Print Preview Dialog**



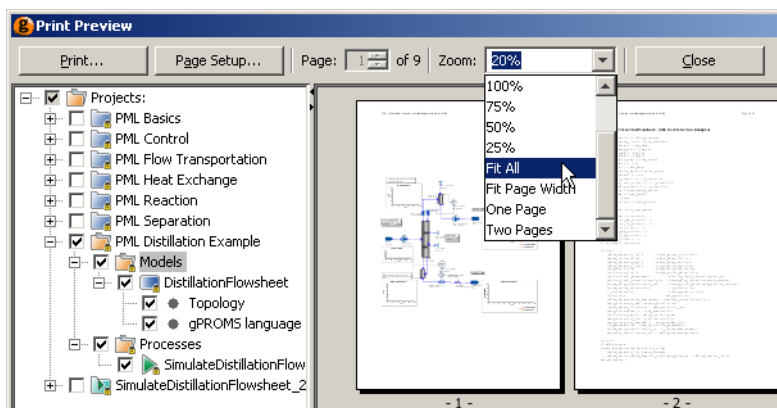
The Print Preview window is split into three sections:

- A set of controls at the top for:
  - Accessing the Page Setup and Print dialogs
  - Selecting the page shown in the Preview pane
  - Zooming the Preview pane
  - Closing the Print Preview dialog
- A tree-view of the available Projects and Cases that can be printed
- A Preview pane

## Print Preview Controls

Along the top of the Print Preview dialog, there are five controls.

**Figure 8.4. Print Preview Controls**



- The first two command buttons respectively activate the Page Setup and Print... dialogs.

When the Print... button is pressed, any changes made to the tree view are remembered the next time the Print Preview is activated.

- Next is a combobox control that specifies the page displayed in the Preview pane.
  - Left-click either of the arrow buttons to the right to increment or decrement the displayed page, or select the number shown and type the desired page number.
  - Note this control may be disabled if more than one page is displayed in the Preview pane. In this case, the view can be changed by moving the scroll bar on the right of the Preview pane.
- Next is the zoom control
  - Left click on the tab to the right to bring down the list of options and select the desired zoom level
  - The print preview can be shown at a specified magnification
    - 25, 50, 75, 100, 200 or 400% by default
    - Other values are possible simply by selecting one of the above and then editing the text to the left
  - Alternatively, the view can be scaled to fit a certain number of pages to the size of the Preview pane

- The "Fit All" option zooms out so that all pages can be shown in the Preview pane.
- The zoom controls affect only what is shown in the Preview pane and have no effect on the final printout
- Finally, the Close button closes the Print Preview dialog.

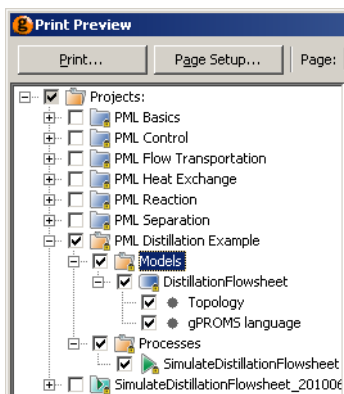
## The Tree-View Pane

This is the area of the Print Preview dialog where you can select which entities should be printed. If you have activated the Print Preview using the File menu, then the Tree View will contain an initial selection depending on which entities were selected in the Model Builder Project Tree at the time.

The Tree View is a typical folder browser just like the tree view in Windows Explorer. Branches can be opened and closed as you would expect. However, next to each entity or folder of entities (e.g. the list of Models within the Project) is a check box. Items that contain a tick are selected for printing; those without a tick will not be printed. If a check box contains a tick on a white background, this means that all sub entities will also be printed; a grey background indicates that some sub entities have been deselected.

Once you have selected all of the entities that you want to print and removed those that you don't, press the Print... button to activate the Print dialog, where you can select a printer, choose how many copies to print and finally send the print job to the printer.

**Figure 8.5. The Tree View (with default selection for Projects)**



## The Preview Pane

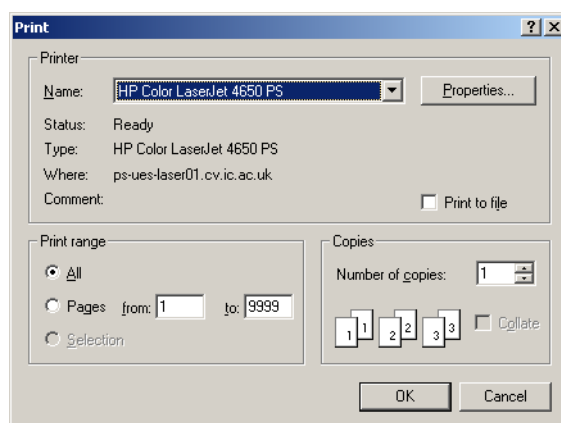
The Preview pane shows how the gPROMS Project(s) or Case(s) will be printed out. The view shown in this pane can be customised using the controls described here and also by moving the mouse pointer over the Preview pane so that it turns into an image of a magnifying glass (as shown in the figure here). When the magnifying glass is visible, pressing the left mouse button zooms the image in one step; whereas the right mouse button zooms the view out.

If the Preview pane is too small, the whole Print Preview dialog can be resized by left clicking on the border of the window (when the mouse pointer turns into a double-headed arrow), or the border between the tree-view and Preview panes can be moved in an identical manner.

## The Print Dialog

Pressing the Print... button on the Print Preview dialog activates the Print dialog, shown below.



**Figure 8.6. Print Dialog**

This is a standard print dialog that can be found in most Windows applications, so needs no explanation here.

## Initial Print Selection

When the Print... dialog is activated from the File menu, gPROMS applies an initial selection of entities to be printed, which can then be modified in the Print Preview dialog. The rules that define the initial print selection depend on which entities are selected in the Project Tree. Multiple Projects and/or entities can be selected by using the **SHIFT** and/or **CTRL** keys in combination with the left mouse button (**CTRL** toggles individual entities, **SHIFT** selects ranges).

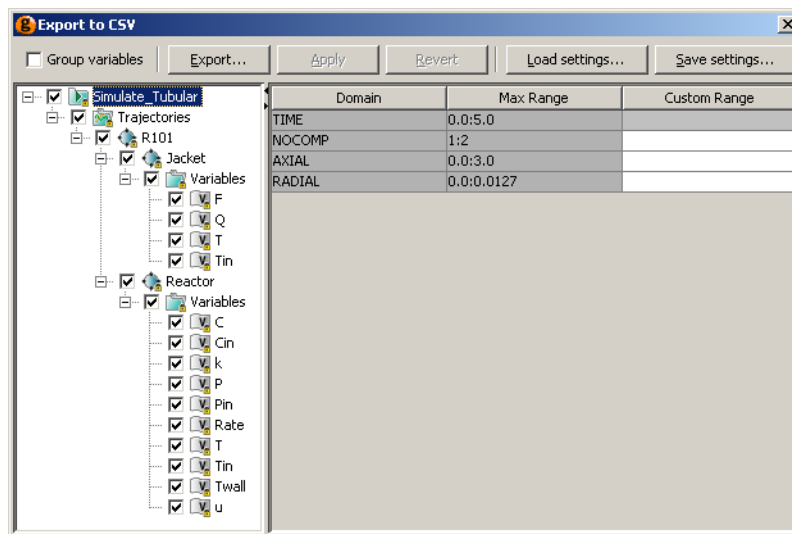
- If no Projects or Cases are selected in the Project Tree, then this is the same as selecting *all* Projects and Cases
- For each Project/Case that is selected in the Project Tree, the initial print selection will depend on whether the Project/Case has been printed before:
  - *All* printable entities defined in the Project/Case will be selected in the Preview Tree if the Project/Case has not been printed before
  - If the Project/Case has been printed before, then only the entities that were printed last time will be selected in the Preview Tree
- If individual entities are selected within a Project (but not the Project itself) then only those entities will be selected in the Preview Tree

## Exporting Data to CSV Files

gPROMS can export the results of Simulation activities, stored in Cases, to one or more comma-separated-value (CSV) files. These files contain the results of the simulation in a simple ASCII format and therefore can easily be imported by a wide variety of applications, such as spreadsheets, mathematical software, data visualisation software etc. A dialog allows the user to specify which variables are exported, including defining subsets of arrays, so the output files can be customised quite comprehensively.

Note that if one simply wants to copy a small amount of data to another application, then it may be more efficient to copy the data directly as described in Viewing Results.

To activate the Export to CSV dialog, right click on a Case and select the Export to CSV... menu item. This will enable the dialog shown below (for the Tubular Reactor example).

**Figure 8.7. Export to CSV Dialog**

There are various controls on this dialog, the main two of which are the tree view in the left-hand, bottom pane, which allows the selection of variables for export; and the domain specification for the variables, which allows the user to choose which elements of a distributed variable are exported. The remaining controls, at the top of the dialog are as follows:

- **Group Variables**

This checkbox specifies that variables that have the same domain dependency should be exported together in the same file, so that a number of separate CSV files are created: one for each set of variables that share the same domain dependency. In this example, four files are created: one for variables that depend only on TIME; one for those that depend only on TIME and NOCOMP; one for TIME, AXIAL and RADIAL; and finally one for variables that depend on TIME, NOCOMP, AXIAL and RADIAL.

If the checkbox is unchecked, then all variables will be exported to a single CSV file.

- **Export...**

The Export... button launches a file browser dialog so that a target directory for the CSV file(s) can be specified, along with a filename. If the Group Variables option is checked, then the filename specified will be appended with an underscore and a number to differentiate all of the files. So if the filename "Tubular" is specified, then for the above example the following four files will be created: Tubular\_1.CSV, Tubular\_2.CSV, Tubular\_3.CSV and Tubular\_4.CSV.

Once the data have been successfully exported, the current settings will be saved temporarily in Model Builder, just as if the Apply button (see below) had been pressed.

- **Apply and Revert**

When the Export to CSV dialog is first launched, as described above by right-clicking on the Case, all Variables will be selected, the Group Variables checkbox will be unchecked and the Domain ranges will be at their default values (the maximum range). As soon as any change is made to any of these settings, the Apply and Revert buttons will be enabled. Pressing the Apply button will save the settings while this Case is open in Model Builder, so that you can close the dialog and retain the settings. Pressing the Revert button will undo all of the changes made since the last time the Apply button was pressed.

Settings saved in this way will be lost once the Case or Model Builder is closed. The same settings can be saved to a file by using the Save settings... button (see below).

If the Export to CSV dialog is closed after some changes have been made, then Model Builder will ask if these changes are to be applied. Pressing the Yes button will apply the changes (just like pressing Apply) and close

the dialog; No will discard the changes and close the dialog; while Cancel will return to the dialog without applying or reverting the changes.

- Load settings...

This button enables a file browser dialog to select a settings file to load. Once the settings file has been loaded, the current settings will be lost and replaced by the ones specified by the file. Should any of the custom ranges be different, then this will be indicated in the Custom Ranges column.



- Save settings...

The Save settings... button saves the current settings in a file so that they can be restored at a later date to any Case of the same type. This saves repeating all of the specifications each time a new Case needs to be exported. (The Apply button only saves settings in memory for each Case individually, so they are lost whenever a Case is closed and cannot be applied to new Cases.)

The settings that are saved include which Variables are selected, the ranges of each domain over which they will be exported (specified independently for each Variable, even if they share the same domain dependency) and the status of the Group Variables checkbox.

When the Save settings... button is pressed, a file browser dialog will appear so that a target directory for the settings file can be specified. The filename may also be modified.

Once the settings have been saved to a file, they will also be saved in Model Builder and therefore the Apply and Revert buttons will be disabled.

The tree view allows one to select which Variables will be included in the CSV file(s). Simply browse through the tree expanding and collapsing the branches using the  and  icons respectively, and select which Variables are to be included by checking (or unchecking) the box next to their name.

Note that if you only want to export a few variables, then it may be more economical to invoke the Export to CSV dialog in a different manner: instead of right-clicking on the Case, browse through the Trajectories branch of the Case and select the Variable or Model instance whose data you want to export, then right-click on it and select Export to CSV... from the menu. This will launch the Export to CSV dialog with only the selected Variable checked; if a Model instance was selected, then all of the Variables within that Model instance and its sub Models will be checked.

These different ways of opening the Export to CSV dialog may seem a little confusing. To summarise:

Entity Right-Clicked	First Time	Subsequent Times
Case	All Variables selected	If the Apply button has been used to save any settings, then all of these will be restored, including Variable selections; otherwise the behaviour will be as the first time.
	Default (max) ranges selected	
	Group Variables unchecked	
Trajectories	All Variables selected	All Variables selected
	Default (max) ranges selected	Saved ranges restored
	Group Variables unchecked	Group Variables restored
Model instance	All Variables within that Model and all of its sub Models will be selected	All Variables within that Model and all of its sub Models will be selected
	Default (max) ranges selected	Saved ranges restored
	Group Variables unchecked	Group Variables restored
Variable	Only the selected Variable is checked	Only the selected Variable is checked
	Default (max) ranges selected	Saved ranges restored

Entity Right-Clicked	First Time	Subsequent Times
	Group Variables unchecked	Group Variables restored

If you open the Export to CSV dialog using any of the last three methods and you want to restore a previously-saved Variable selection, then simply press the Revert button. Also, if you close the dialog, you may be asked to save changes even if you have made none, because the Variable selection may differ from that in the saved settings.

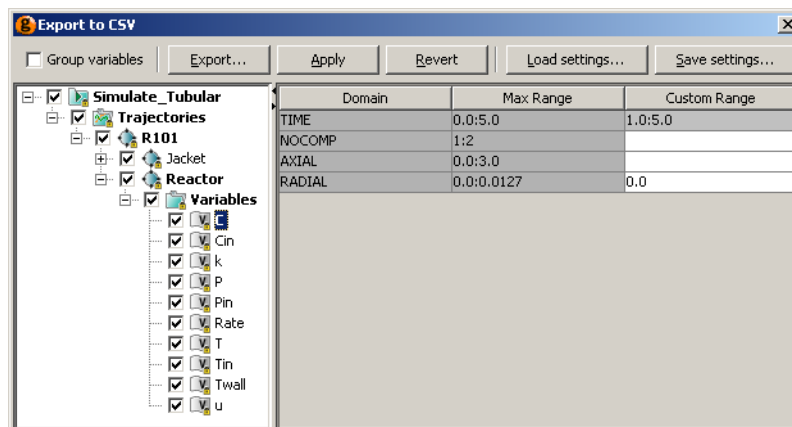
By default, all elements of a distributed Variable will be exported to the CSV file. To modify this, select a Variable by left clicking on its name. This will then populate the table in the right-hand pane. This shows a list of domains (both discrete and continuous) over which the Variable is distributed, their maximum ranges and a custom range. If this custom range is left blank, then all elements of that range will be included in the exported data. To modify this, simply left click on the cell that defines the custom range for the desired domain and enter a value using the same format in which the max range is given. For example, suppose in the Tubular Reactor example, that we only want to export values of the concentration for the last 4 seconds of simulation. This can be achieved by typing

1.0:5.0

in the custom-range cell.

One might go further and only desire the values of a variable at a particular point in the domain: this can be achieved simply by specifying a single value in the custom domain. So to see only the values of the concentration for RADIAL = 0, simply enter 0 in the custom-range cell for the RADIAL domain. Both of these specifications are shown in the figure below.

**Figure 8.8. Example of Custom Range specification**



Multiple ranges and values can also be specified, by separating them with commas. Some examples are shown below.

0.0:1.0,2.0,3.0:4.0  
 1,2,4:10  
 "H2", "N2"  
 "H2": "CH4"

The first line illustrates the specification of ranges and points together for a continuous domain. The second line shows a similar specification for an integer domain. The third and fourth lines illustrate how to specify ranges for Variables defined over Ordered Sets. The first of these lines indicates that the Variable should be exported for only two of the elements in the Ordered Set; the last line specifies a range of elements: all elements between and including "H2" and "CH4".

The dialog checks to make sure that the range specifications are correctly formatted and within the allowed maximum range. Should a mistake be made when entering a custom range, the cell border will turn red and the

new range will not be accepted until the mistake is corrected. (Spaces between commas are ignored and there is some flexibility when entering numbers (e.g. if one types 5 referring to a continuous domain, then this will be converted to 5.0), but it is necessary to adhere to the general format.)

Note that once a custom range has been specified for a Variable, its name will appear in bold face to indicate that it is not at the default value (the whole range), as do all of the section headings that contain it. Clicking on one of these section headings allows one to specify the domain ranges of all of its entities at the same time. Starting with the above example, suppose we wanted now to set the ranges of all of the Variables in the Reactor Model instance to the same value. First, we would left-click on either the Variables heading (within the Reactor Model — since there may be other Model instances, all of which will contain a Variables heading) or on the Reactor heading. Once the heading is selected, the common ranges will be shown in the cells on the right. In this case, the top and bottom cells will be showing <different> to indicate that the ranges of the Variables are not all the same. To set them all to the same value, simply enter the ranges in the cells as before.

## Multiple selection

Many *standard* editing features, e.g. cut, copy, paste, delete etc. all found under the Edit menu are available within the gPROMS ModelBuilder environment. Most of these (and other) actions can be applied to multiple Entities and Projects. Multiple selections are made using **Ctrl+Left-click** and **Shift+Left-click** mechanisms applied to the Project and Entity names in the ModelBuilder's navigation tree.

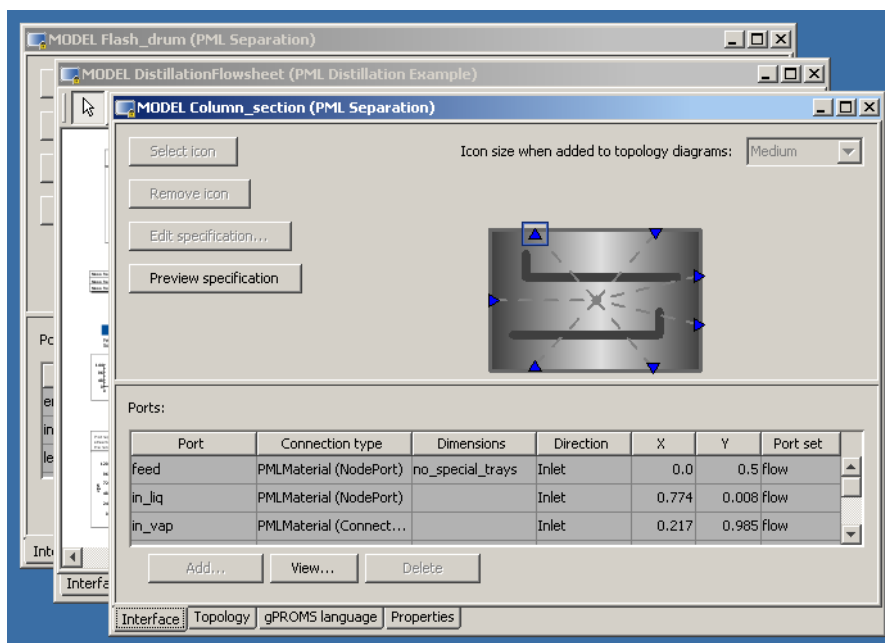
Similarly, the Open action in the Project menu can be applied simultaneously to multiple Projects in the file selection dialog.

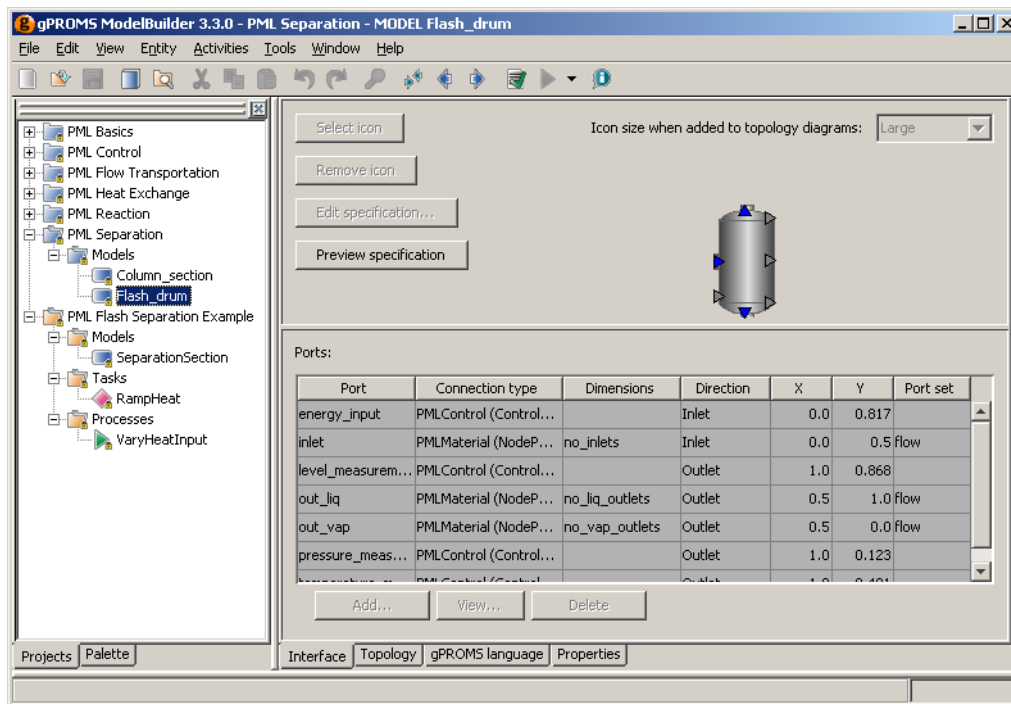
## Desktop view

The Window menu gives the user the option of having multiple editors open simultaneously (see first figure) or only having a single editor open at a time (see second figure). In multiple editor mode, the Window menu contains standard Cascade, Tile Horizontally, Tile Vertically and Close All options.

The default setting for this option can be adjusted in the ModelBuilder Preferences dialog for details.

**Figure 8.9. Multiple editor mode**



**Figure 8.10. Single editor mode**

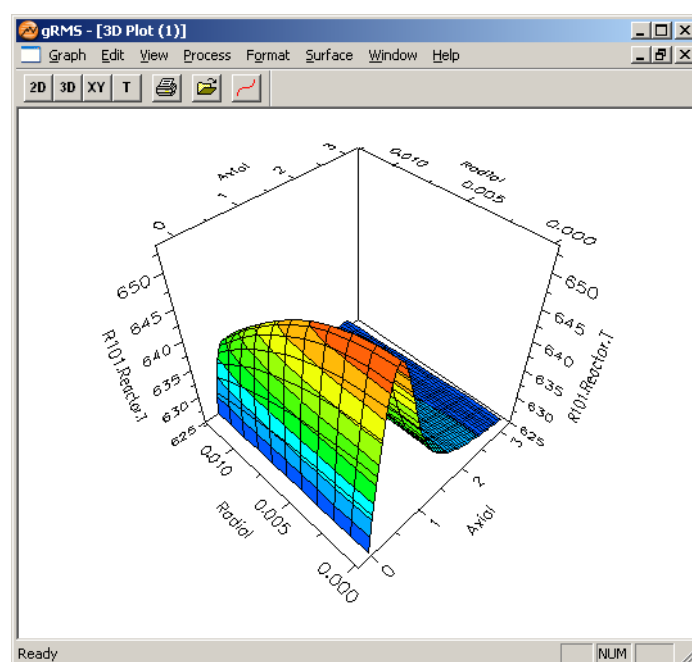
## Collapse project tree action

The Collapse project tree action (**Ctrl+T**) is found on ModelBuilder's View menu, allowing the Project tree to be collapsed to a view showing only the names of the open (Library) Projects and Cases.

# Chapter 9. gRMS Output Channel

The gRMS (gPROMS Results Management Service) application provides facilities for plotting and printing gPROMS results as 2D and 3D graphs. It is started when an Activity is executed if specified in the execution control dialog. This chapter goes into some more detail that you may find useful in making the most out of this powerful results presentation tool.

**Figure 9.1. Temperature profile in tubular reactor**



The UNIX version of gRMS starts up displaying the gRMS Toolbar which contains menu-items for loading/saving data and for creating new plots. Each plot is displayed in a separate window with its own menu-bar containing items specific to that plot.

The Windows version of gRMS starts up displaying an empty frame. Each plot is displayed in a sub-window within that frame. All menu-items appear on the main frame's menu and plot specific menu-items always operate on the currently active plot. If there is no active plot, then plot specific menu-items cannot be used and will be grayed out.

## gRMS processes

gRMS organises all its results in Processes. A gRMS Process:

- is created when a gPROMS Activity starts being executed and it is selected in the execution dialog that the results are to be sent to gRMS<sup>1</sup>;
- receives data from gPROMS including information on the variables in the problem as well as their values during the simulation;
- remains in existence even after the gPROMS simulation has terminated or, indeed, gPROMS has been exited;
- can be saved as a permanent file on disk; such files normally have a .gRMS file extension;
- may be reloaded by gRMS from the above file at a later time in order to display results etc. This can be done using the **Open...** menu-item.

<sup>1</sup>For Estimation Activities, a separate gRMS Process is created for each experiment.

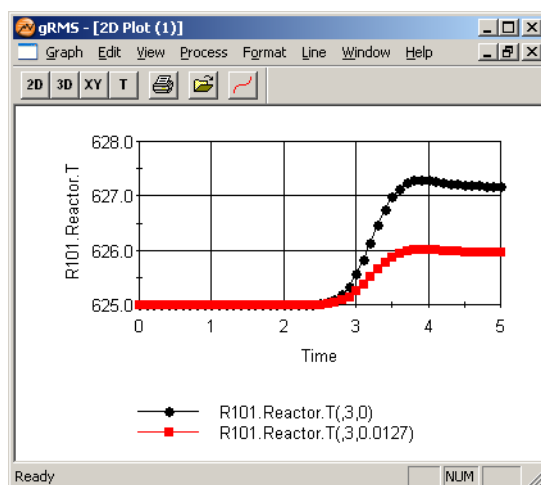
gRMS manages its processes using the **Process** menu. Initially this menu contains the **Open...**, **Save All** and **Close All** items. Each new Process that is created (either by an executing gPROMS Process or by loading in a .gRMS file) appears as an additional item on this menu. The menu-item for each Process in the **Process** menu is a pull-right menu containing the following items:

**Table 9.1. Process menu items**

<b>Close</b>	Close the Process. If the Process has not been saved, then the data that it contains will be lost.
<b>Save</b>	Save the process using its current name. This menu-item is disabled if the process has already been saved.
<b>Save As...</b>	Display a standard file dialog allowing you to choose a destination directory and filename to which the Process will be saved.
<b>Properties...</b>	Displays statistics about the Process including the number of variables, domains and time-intervals.

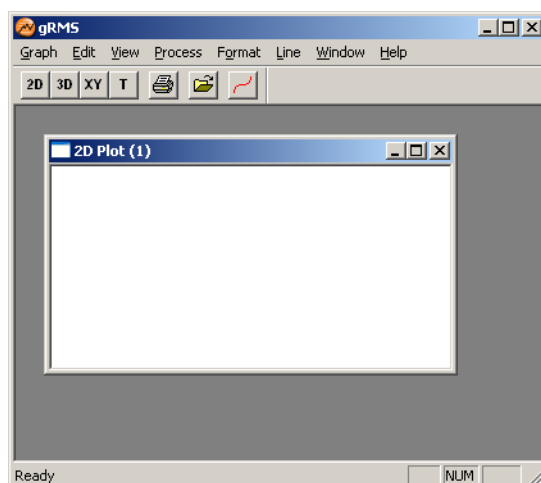
## Plotting 2D graphs

**Figure 9.2. 2D graph**



To plot a new 2D graph, select the **Graph -> New 2D Plot** menu-item. A new **2D Plot Window** is displayed containing an empty 2D plot.

**Figure 9.3. New 2D Plot Window**





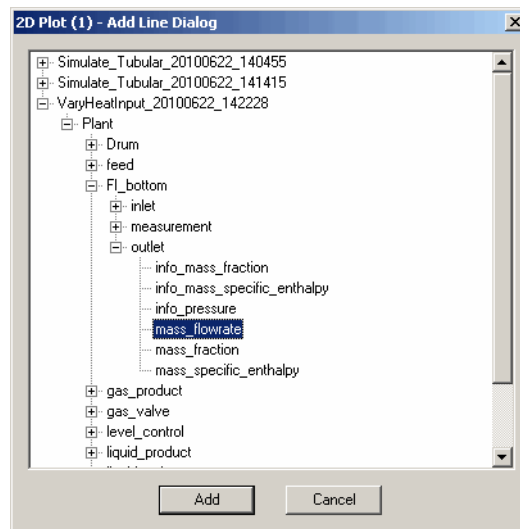
## Adding lines to a plot

To add a new line to a plot select the **Line -> Add...** menu-item. The **Add Line Dialog** is displayed which allows you to navigate the model hierarchy and choose a variable to be plotted. When a line is added to a plot, a **Line Properties Dialog** will be displayed so that the line can be formatted. The line will be drawn on the plot only if it has a single free domain i.e. the corresponding variable is a function of a single independent variable (usually Time).

On Windows, navigation of the model hierarchy is achieved using a tree-style mechanism. To add a line corresponding to a certain variable:

- *either* double-click on the variable,
- *or* click on the variable to select it and then click **OK**.

**Figure 9.4. Add line**



On UNIX, navigation of the model hierarchy is achieved using a list. Selecting (single-clicking) an item in the list causes it to be listed in the **Selection** field. The action of the **OK** button depends on what type of item is in the **Selection** field:

**Table 9.2. Selection field items**

<UP>	Moves up the hierarchy by one level.
xxx*	Moves down the hierarchy into model instance xxx.
xxx	Adds a line to the plot using the variable xxx.

The behaviour of this dialog can be altered by the three toggle buttons which, by default, are all set off.

**Show variables as flat list** - setting this toggle flattens the model hierarchy and activates the **Filter** field which can then be used to filter the contents of the list using standard UNIX wild-cards ("?" to represent one character, "\*" to represent any number of characters.)

**Show all descendants** - by default, the list only displays the models/variables at the current level of the hierarchy. With this toggle set, the list also includes the names of all models/variables deeper in the hierarchy.

**Show full variable names** - by default, the list only displays the next element in the name of each model/variable. Setting this toggle causes the full names to be listed.

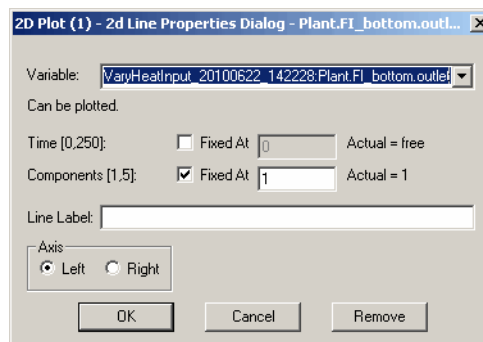
## Formatting lines

For each line on the plot, the **Line** menu contains a pull-right menu-item (if the line cannot be plotted, then on Windows its name is preceded with "\*" and on UNIX it is highlighted in red.) The pull-right menu contains the following items:

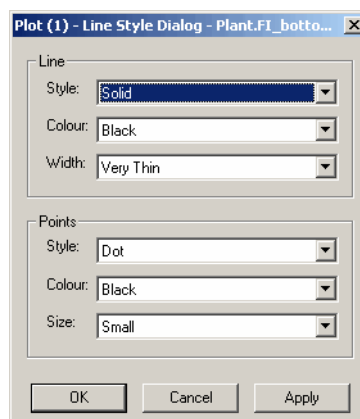
**Table 9.3. Line menu items**

<b>Properties...</b>	Displays a <b>Line Properties Dialog</b> for the line. This dialog presents you with a list of all the domains of the line (3 in the example shown, <b>Time</b> , <b>Axial</b> and <b>Radial</b> .) In order to be plottable a line must have only one free-domain which is achieved by fixing the other domains to a point. In addition, this dialog allows you to specify a label for the line that will appear in the plot legend, and to specify which of the y-axis the line should be plotted against. N.B. If a plot contains only one line it will be plotted against the Left axis even if you specify otherwise. A drop-down list on this dialog allows you to change the data-source to any similar variable or to convert the line into a Template.
<b>Style...</b>	Pops up a <b>Line Style Dialog</b> for the line to allow you to change its appearance on the plot.
<b>Copy</b>	Adds an identical copy of the line to the plot.
<b>Remove</b>	Removes the line from the plot.

**Figure 9.5. Line properties**



**Figure 9.6. Line styles**



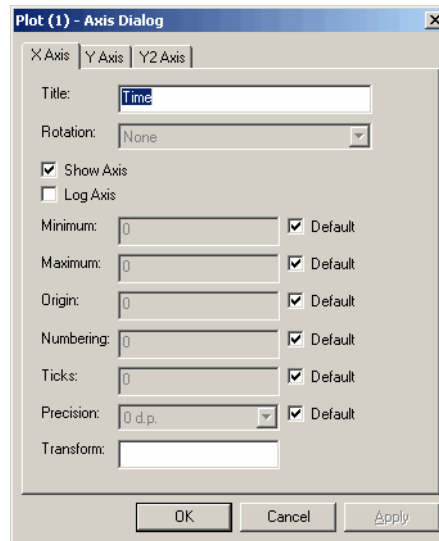
## Formatting 2D plots

The format of the plot is controlled via the items in the plot's **Format** menu.

### Axes

In Windows, the format of the axes can be changed using the **Axis Format Dialog** which is displayed when the **Format -> Axis...** menu-item is selected.

**Figure 9.7. Axis Format Dialog (Windows)**

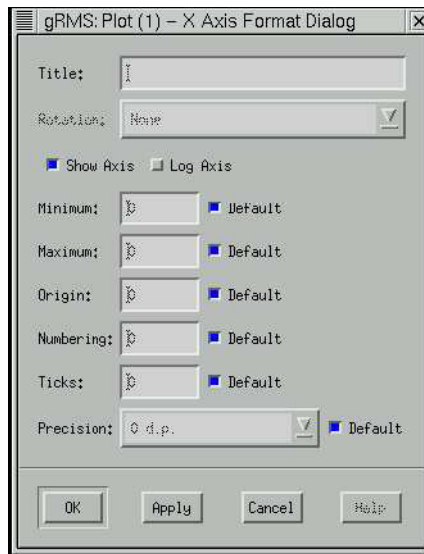


**Table 9.4. Axis Format Dialog (Windows) Entries**

Title	Title to display on the axis. By default, this is the name of the first line plotted against that axis.
Rotation	Orientation of the axis title. This cannot be changed for the $x$ -axis.
Show Axis	Select to display the axis.
Log Axis	Select to display the axis with a logarithmic scale.
Minimum	Minimum value of the data on this axis.
Maximum	Maximum value of the data on this axis.
Origin	Specifies where the axis should be drawn. For example, setting the $x$ origin to 4.0 will cause the left $y$ -axis to cross the $x$ -axis at 4.0.
Numbering	Increment between axis numbers.
Ticks	Increment between axis ticks.
Precision	Number of decimal places the axis numbering should use.

A bounding box can be displayed around the axes by selecting the **Format -> Bounding Box** menu-item.

On UNIX, the **Axis Format Dialog** can be used only to alter the format of one axis at a time and **Format -> Axis** menu-item is a pull-right menu containing an item for each axis.

**Figure 9.8. Axis Format Dialog (UNIX)**

The bounding box is displayed by selecting the **Format -> Axis -> Bounding Box** menu-item.

## Default line styles

When lines are first created on 2D plots, gRMS chooses an unused line style from the following list of default styles:

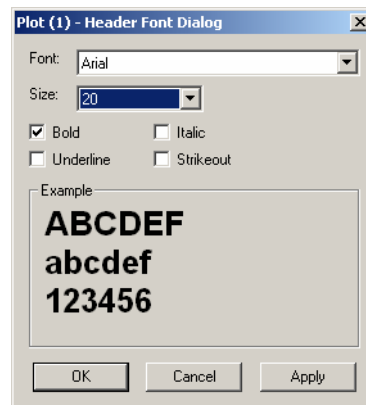
**Table 9.5. Default Line Styles**

Name	Line Style	Colour	Data Points Style
Black	Solid	Black	Dot
Red	Solid	Red	Square
Blue	Solid	Blue	Triangle
Green	Solid	Green	Diamond
Magenta	Solid	Magenta	Star
Dashed Black	Dashed	Black	Dot
Dashed Red	Dashed	Red	Square
Dashed Blue	Dashed	Blue	Triangle
Dashed Green	Dashed	Green	Diamond
Dashed Magenta	Dashed	Magenta	Star

The list of default styles can be edited using the **Format -> Default Line Style** menu-item. As changes to the default style list are lost when gRMS is shut down, this facility is only really useful when used in conjunction with Plot Templates.

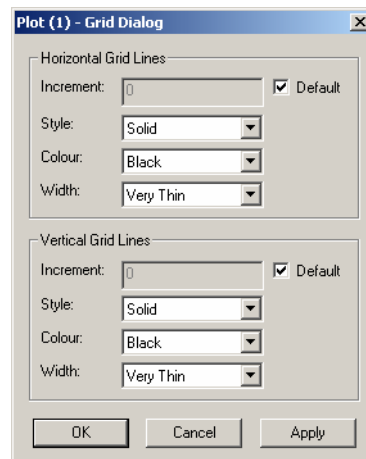
## Fonts

The fonts used to display text on the plot can be changed using the items in the **Format -> Fonts** menu. These can be used to display a **Font Selection Dialog** which allows a font to be picked from a system dependent list.

**Figure 9.9. Font selection**

## Grid

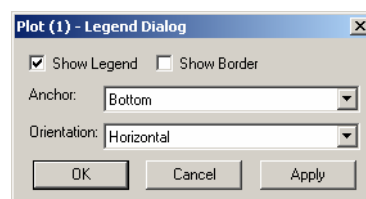
The format of the plot's grid can be changed using the **Grid Dialog** which is displayed when the Format -> Grid... menu-item is selected.

**Figure 9.10. Grid format**

By default, the **Increment** of a grid-line is the same as the axis-numbering. To remove the grid-lines set the **Increment** to 0.

## Legend

The format of the plot's legend can be changed using the **Legend Dialog** which is displayed when the **Format -> Legend...** menu-item is selected.

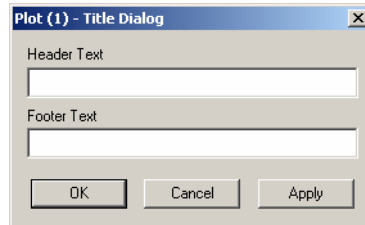
**Figure 9.11. Legend**

N.B. The values of the **Anchor** and **Orientation** are only hints, if the window is too small then the legend may not appear as specified.

## Title

The plot can be supplied with a header and footer using the **Title Dialog** which is displayed when the **Format -> Title...** menu-item is selected.

**Figure 9.12. Title and footer**



## Scaling, zooming and translation

**Table 9.6. Transformation desired**

<b>Scaling</b>	With <b>Ctrl</b> pressed and the middle mouse button depressed, moving the mouse up and down zooms the plot in and out.
<b>Translation</b>	With <b>Shift</b> pressed and the middle mouse button depressed, moving the mouse translates the plot.
<b>Zooming</b>	With <b>Shift</b> pressed and the left mouse button depressed the mouse can be used to select an area to zoom into.

Pressing **r** resets the scaling, translation and zooming.

N.B. For mice with only two buttons pressing the middle button is simulated by pressing both buttons simultaneously.

## Plotting 3D graphs

To plot a new 3D graph select the **Graph -> New 3D Plot** menu-item. A new **3D Plot Window** is displayed containing an empty 3D Plot. The appearance of this window is the same as for a 2D Plot except the **Line** menu is replaced by the **Surface** menu.

## Adding a surface to a plot

Adding a surface to a 3D Plot is achieved in the same way as adding a line to a 2D Plot. Only one 3D surface can be plotted at a time. If the plot already contains a surface, then the **Surface -> Add...** menu-item is renamed **Surface -> Change...**; otherwise, the behaviour is the same.

## Formatting surfaces

The **Surface** menu contains the following items for formatting the surface displayed on a 3D Plot.

**Table 9.7. Surface menu items**

<b>Properties...</b>	Displays a <b>Surface Properties Dialog</b> for the surface. This dialog is functionally identical to the <b>Line Properties Dialog</b> .
<b>Remove</b>	Removes the surface from the plot.

<b>Draw Mesh</b>	When set, the surface is plotted in 3D with the X-Y grid projected onto the surface.
<b>Draw Shade</b>	When set, the surface is plotted in 3D with flat shading.
<b>Draw Contour</b>	When set, contour lines are automatically drawn between distribution levels in the data.
<b>Draw Zones</b>	When set, each distribution level in the data is displayed in a different solid colour.

## Formatting 3D plots

The format of the plot is controlled via the items in the plot's **Format** menu.

### Axes

Windows: Same as for 2D plots though there are fewer controllable parameters.

Unix: The format of the axes can be changed using the **3D Axis Format Dialog** which is displayed when the **Format -> Axis...** menu-item is selected.

Functionally, this is the same as for 2D plots except there are fewer controllable parameters.

### Fonts

Windows: Same as for 2D plots except that only the style (Bold, Italic, ...) and not the face can be selected for the Axis font.

UNIX: Same as for 2D plots except the Axis font is picked from a pull-right menu rather than the **Font Selection Dialog**.

### Legend

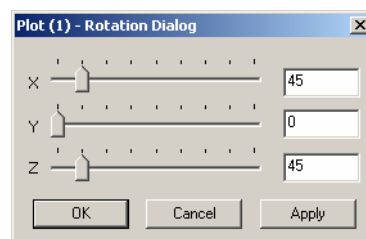
Same as for 2D plots except that there is an option to display the legend as either **Stepped** or **Continuous**.

### Rotation

The default rotation of the plot about the (X,Y,Z) axes is (45,0,45). With the middle mouse button (or both buttons for 2-button mouse) depressed, moving the mouse rotates the plot. If you hold down **x**, **y**, or **z**, then the rotation is restricted to being around that axis. If you hold down **e**, then the rotation is restricted to being perpendicular to the screen. Alternatively you can change the rotation from the **Format** menu.

Windows: The rotation of the plot can be changed using the **Rotation Dialog** which is displayed when the **Format -> Rotation...** menu-item is selected.

**Figure 9.13. Plot rotation**



Unix: The rotation of the plot can be changed to a limited set of preset values using the sub-items in the **Format -> Rotation** pull-right menu.

## Title

Same as for 2D plots.

## Scaling, zooming and translation

Same as for 2D plots except you need to hold down **Ctrl** and not **Shift** when zooming. N.B. You can actually hold down **Ctrl** when zooming 2D plots, but in this case the axes will only be displayed if they lie within the selected zoom area.

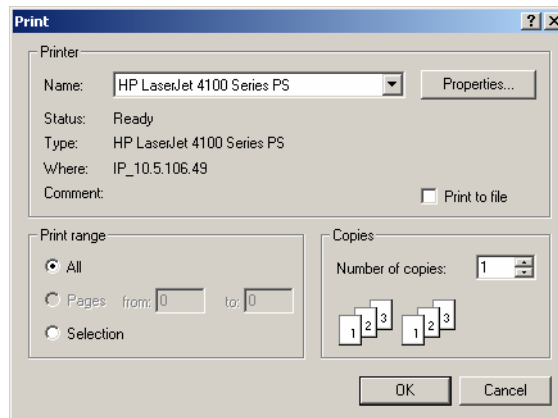
## Printing gRMS plots

2D and 3D Plots can be printed by selecting the **File -> Print...** menu-item.

Windows gRMS uses the standard Windows print dialog.

The printed plot will be scaled to fit the page whilst maintaining the same aspect ratio as displayed on the screen.

**Figure 9.14. Print dialog**



Unix The **Print Dialog** will be displayed allowing you to choose the format of the printed plot, and the name of the printer or file you wish to print to.

### PostScript

The default format which outputs an encapsulated PostScript (EPSF-2.0) image of the graph using device-independent PostScript operators. Clicking the **Props...** button pops-up the **Printer Properties Dialog** which contains additional formatting options for **PostScript** output. The **Fonts** option requires a little explaining, by default gRMS tries to **Use X Fonts** which means that the plot is printed using the closest available font to that displayed on the screen. If this does not print correctly then you can disable this option and select fonts from the four menus. N.B. To obtain the best WYSIWYG (what you see is what you get) output filling the whole of the printed page, stretch the Plot Window so that it has the aspect ratio of your paper in the orientation you are using and then set the **Maintain Aspect Ratio** toggle button to be off. Three additional output formats are also available, however the options in the **Printer Properties Dialog** are not available for them

### PostScript Bitmap (Monochrome)

An encapsulated PostScript (EPSF-2.0) image of the graph created by taking the pixels on the screen and outputting them using the PostScript image operator. The resolution is not as good as with the standard PostScript format, and the file size is much larger. The



only reason to use this format is if the plot uses especially unusual fonts which are not reproduced correctly by the standard PostScript format.

PostScript Bitmap (Colour)

Same as the above, but in colour.

X Window Dump

A standard X Windows Dump representation of the graph.

## Viewing and exporting data

Data from 2D and 3D plots can be viewed in a window or exported as a *tab/space/comma delimited* ASCII text file suitable for importing into a spreadsheet. N.B. *comma delimited* format is only available in the Windows version of gRMS.

### 2D plots

The data can be viewed by selecting the **Graph -> View Data...** menu-item. The data is displayed in a table with the values of the free-domain in the first column and the values of lines plotted against that domain in subsequent columns. If the plot contains lines from variables in different Processes, or lines plotted against different free-domains then multiple tables are displayed.

Windows    The data can be exported to a file by selecting **Graph -> Export Data....** This displays a Windows file dialog for you to specify the file name and type.

Unix        The data can be exported to a file by selecting either the **Graph -> Export Data -> Tab Delimited Table...** or **Graph -> Export Data -> Space Delimited Table...** menu-item. This displays a standard Motif file dialog for you to specify a file name.

### 3D plots

The data can be viewed by selecting either the **Graph -> View Data -> Table...** or **Graph -> View Data -> Matrix...** menu-item. In **Matrix** format the data is exported in a table with the x-values labelling the columns, the y-values labelling the rows and the z-values in the table. The **Table** format exports the data in a three column table (x,y,z).

Windows    The data can be exported to a file by selecting **Graph -> Export Data....** This displays a Windows file dialog for you to specify the file name and type.

Unix        The data can be exported to a file by selecting one of the **Graph -> Export Data -> Tab Delimited Table...**, **Graph -> Export Data -> Space Delimited Table...**, **Graph -> Export Data -> Tab Delimited Matrix...** or **Graph -> Export Data -> Space Delimited Matrix...** menu-items. This displays a standard Motif file dialog for you to specify a file name.

## Exporting images

Graphical images of plots can be exported from gRMS for inclusion in documents and presentations.

Windows    Select the **Graph -> Export Image...** menu-item. This displays a standard Windows file dialog for you to specify a file name and image type from the following:

- Enhanced Metafile (emf)
- Aldus Placeable Windows Metafile (wmf)
- Windows Bitmap (bmp)
- Standard PNG (png)
- Interlaced PNG (png)

- JPEG (jpg)

Unix      Select the **Graph -> Print...** menu-item and use the **Printer Dialog** to select output to a file.

## Templates

To simplify the use of gRMS when creating many similar plots, e.g. for multiple runs of the same process, gRMS allows plot and line *Templates* to be defined. A template is a *description* of a plot (or line) that contains everything needed to display the plot except for the data itself. Using a template requires sources of data for the plot (or line) to be specified; this is known as instantiation. There are two types of Templates, *Line/Surface Templates* and the much more useful *Plot Templates*. All references to lines on 2D plots in the following discussion also apply to a surface on a 3D plot.

### Line templates

Line templates are *descriptions* of lines; they contain everything needed to display a line except for the data itself. Line templates contain information about the line colour, line style, line width, point style (what symbol is used), point colour, point size and the line label. Like lines, line templates appear on the Lines menu of the Plot Window. They can be manipulated exactly like lines. They can be instantiated from their Line Properties Dialog, by selecting a variable from the variable drop-down list (this contains a list of all variables with the same name, distributed over domains of the same name and in the same order). Lines can be converted into templates from their Line Properties Dialog by selecting the Template: \*.xxx item from the Variable drop-down list.

### Plot templates

Plot Templates are *descriptions* of plots; they contain everything needed to display a plot except for the data itself. Plot templates contain information about each axis (scale, origin, minimum value, maximum value, tick parameters, title, etc.), all fonts used, grid lines, the legend, the title, the properties of each line (as in a line template) and the set of variables to be plotted. One can imagine that setting all of this information each time a process is executed will become extremely tedious. Also note that a plot template can be instantiated with processes that are not identical to the one used to create the template: gRMS will search through the data to find as many matches to the variables that it has in the template. Plot templates are created from normal 2D and 3D plots. N.B. It is possible to create a template from a plot that contains no lines or surfaces; just line templates and axes, font, grid, legend, title and default line style formatting. Such a template requires no instantiation.

### Saving a plot as a template

To save a plot as a template select the **Graph -> Save As Template...** menu-item. This displays a standard file dialog allowing you to choose a name for the template file. We suggest that plot templates are saved with the file extension .gpt.

### Creating a new plot from a saved template

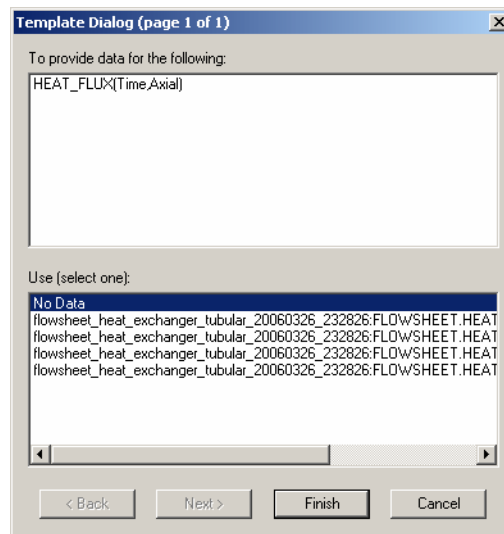
To create a new plot from a previously saved template select the **Graph -> Open Template...** menu-item. This displays a standard file dialog allowing you to choose the template to open.

### Using an existing plot as a template

If you want to quickly use a plot as a template without going through the Save/Open procedure then select the **Graph -> Use As Template...** menu-item.

### Instantiating a plot template

The Plot Template Dialog is displayed when you use any plot template that requires instantiation (i.e. was created from a plot containing lines or surfaces). This dialog is used to instantiate the template.

**Figure 9.15. Instantiating a plot template**

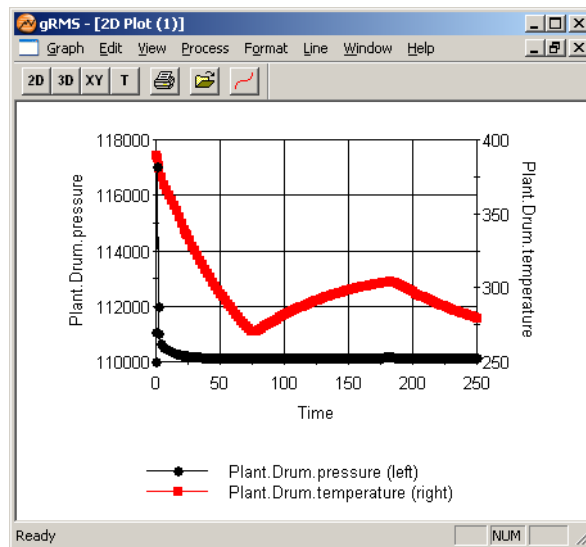
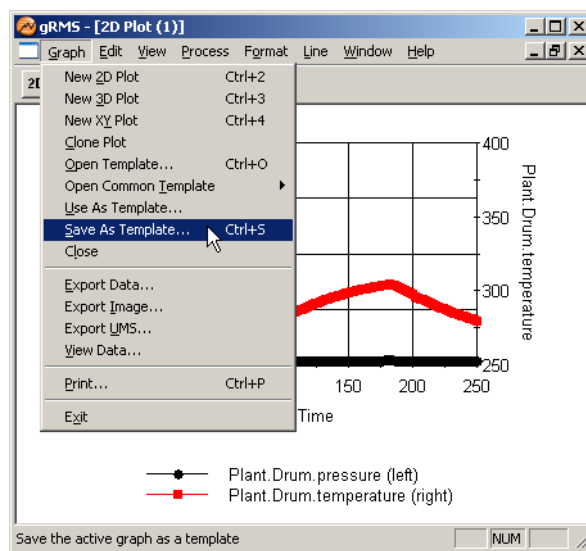
The dialog displays the details for one required data-source at a time. If more than one data-source needs to be instantiated then the Back and Next buttons can be used to move through the required data-sources. The dialog contains two lists. The top list contains all the variables that are required from the data-source whilst the bottom list contains all the possible data-sources meeting these requirements (from those processes loaded into gRMS). A particular data-source is chosen by selecting it from the bottom list. When you have specified all the data-sources click Finish. By default all the data-sources are instantiated as *No Data* which means that lines depending on those data-sources will become *Line Templates* on the finished plot.

## Common templates

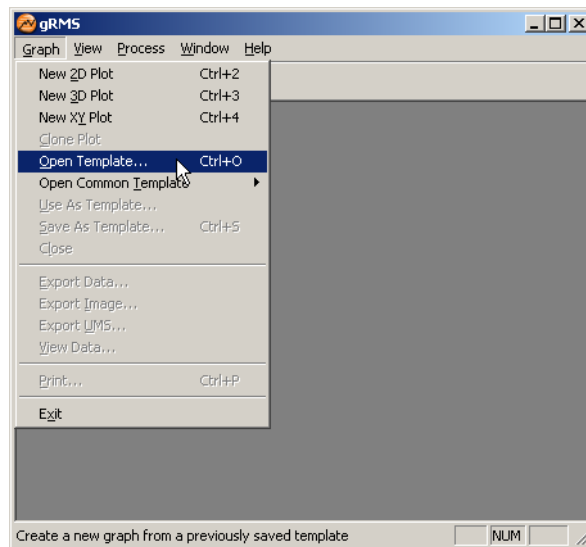
Plot templates saved in the 'oc' directory of the gPROMS installation directory (as identified by the value of the GPROMSHOME environment variable) are known as common templates. These appear in the **Graph -> Open Common Template** menu for easy access. If you have common templates called plot2d.gpt and plot3d.gpt then gRMS uses these when you select the **Graph -> New 2D Plot** and **Graph -> New 3D Plot** menu-items. Usually you would create these templates from plots with no lines or surfaces.

## Plot template example

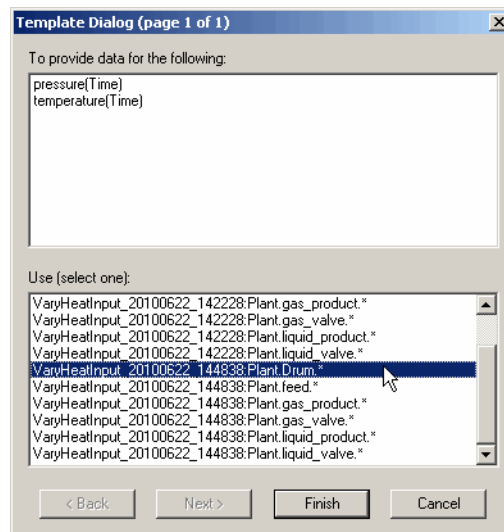
Plot templates can be used to simplify viewing of the results for similar sub-models within the same gPROMS process or for viewing the results of the same model for different simulation runs (Processes). The example shown creates and uses a template for models with pressure and temperature variables.

**Figure 9.16. Create a plot of P and T from one model.****Figure 9.17. Save the plot as a template.**

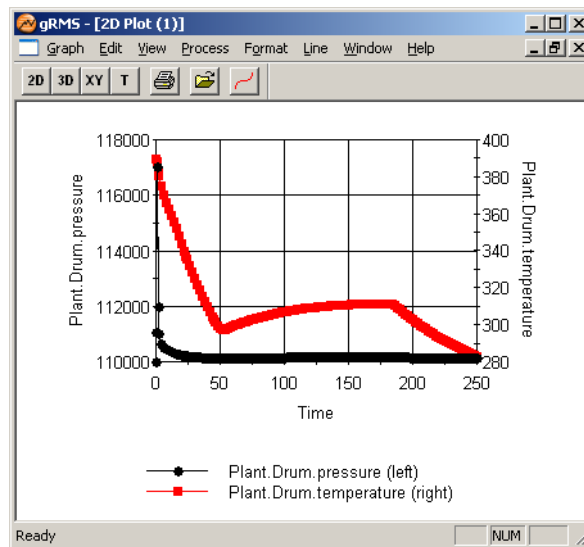
Use the Save File dialog to select a location and file name for the template.

**Figure 9.18. Open the template.**

Use the Open File dialog to select the template file to open.

**Figure 9.19. Instantiate the template from a different model with P and T.**

(Notice the date stamp of the Process name: it is a more recent Simulation of the same Model. The dialog shows all Models, within any Process, that can be used with the template, so one can easily apply the template to different Models within any Process.)

**Figure 9.20. View the resulting plot.**

This method is useful if you want to save the template and reuse it on another occasion (i.e., after the plot window has been closed, when all of the plot formatting will be lost). If you already have a plot window open and want to use it as a template for a more recent simulation (or a different model with the same set of Variables), then a quicker approach is to use the Use As Template... option in the File menu. This bypasses the saving and loading steps in the procedure described above. Of course, this doesn't save the template permanently: to do so, use the Save As Template... option.

## Advanced use of gRMS

In all likelihood you will never need to use any of the features described in this section, but for the adventurous herein may lie some items of interest.

## Preventing gRMS from starting automatically with gPROMS

To prevent gPROMS automatically starting gRMS set the NOGRMS environment variable. N.B. gRMS will also not be started automatically if the GRMSPORT or GRMSHOST environment variables are set, or if the UNIX version of gPROMS is started with the -port or -host flags.

## Starting gRMS independently from gPROMS

If you wish to start gRMS independently of gPROMS then you can start it with the command line:

```
gRMS.exe [-port number] [-dir directory] [-print printer] [-lpr]
```

where the contents of [ square brackets ] are optional command line switches for the following:

**Table 9.8. Command line switches**

-port	gRMS is a server application that receives data over a network from gPROMS. This command line switch tells gRMS to listen for connections from gPROMS on a given port number. When gRMS is started by gPROMS this is set automatically. If not specified gRMS checks to see if GRMSPORT is set and if not uses port 9876.
-------	---

-dir	This is the directory which gRMS will try to open and save files to by default. When gRMS is started by gPROMS this is set automatically. If not specified gRMS check to see if GRMSDIR is set. If GRMSDIR is not set then it uses GPROMSDIR/output and if that is not set it uses the directory it is started in.
-print	This switch is only for the UNIX version and can be used to specify the name of the default printer that gRMS should print plots to. If not specified gRMS checks to see if GRMSPRINTER is set.
-lpr	This switch is only for the UNIX version. By default gRMS uses the UNIX system program <b>lp</b> to print plots, some earlier versions of UNIX do not have this program and use one called <b>lpr</b> instead, this switch tells gRMS to use the <b>lpr</b> program. If not specified gRMS checks to see if GRMSLPR is set..

## Running gPROMS and gRMS on different machines

Because gPROMS and gRMS communicate using the TCP/IP protocol they can be run on separate machines and communicate over a local area network, or the internet. This is best demonstrated by example: if we want to run gRMS on a machine called marzipan.psenterprise.com (the name of the machine gPROMS is running on is not important) and communicate using port 9999 then gRMS is started like this:

```
gRMS.exe -port 9999 -dir ~/gPROMS/output
```

and gPROMS is started like this:

```
gPROMS -port 9999 -host marzipan.psenterprise.com
```

N.B. The Windows version of gPROMS does not currently accept command line arguments so you would have to set GRMSPORT and GRMSHOST instead.

## Multiple gPROMS runs communicating with a single gRMS

More than one gPROMS run can communicate their results to a single instance of gRMS at a time. Again we demonstrate by example, and use port 9999 for communication. gRMS is started like this:

```
gRMS.exe -port 9999 -dir ~/gPROMS/output
```

the first gPROMS is started like this:

```
gPROMS -port 9999
```

and the second gPROMS is started in the same way:

```
gPROMS -port 9999
```

Both gPROMS runs will now communicate their results to gRMS.

N.B. The Windows version of gPROMS does not currently accept command line arguments so you would have to set GRMSPORT instead.

## gRMS resources under UNIX

X-Windows provides a mechanism to customise applications using Resource Files. It is beyond the scope of this manual to discuss the eccentricities of this mechanism other than to refer the user to a book ( X-Window System

User's Guide, OSF/Motif 1.2 Edition, O'Reilly & Associates, Inc., ISBN 1-56592-015-5 ) and to list the resources that can be used to customise gRMS. If you would like help in creating a gRMS resource file on your system then please contact

support.gPROMS@psenterprise.com

The resources provided for customisation can be split into three sets:

- Those mimicking the command line switches. If gRMS is started from the command line using switches then these override the associated resource settings.
- Those that control the overall appearance of gRMS.
- Those that control the appearance of individual gRMS windows and dialogs.

**Table 9.9. Resources mimicking the command line switches.**

Resource	Function
Grms.directory	Duplicates function of <b>-dir</b> command line switch.
Grms.lpr	Duplicates function of <b>-lpr</b> command line switch.
Grms.port	Duplicates function of <b>-port</b> command line switch.
Grms.printer	Duplicates function of <b>-print</b> command line switch.

**Table 9.10. Resources controlling individual windows and dialogs.**

Resource	Function	Default
height	The height of the dialog or window.	Not set, except for Grms.PlotWindow.height that is set to 600.
width	The width of the dialog or window.	Not set, except for Grms.PlotWindow.width that is set to 500.
foreground	The foreground colour of the dialog or window.	Not set, so defaults to value of Grms*.foreground.
background	The background colour of the dialog or window.	Not set, so defaults to value of Grms*.background.
x	The initial x position of the top left corner of the dialog or window.	Not set, so position depends on Window Manager.
y	The initial y position of the top left corner of the dialog or window.	Not set, so position depends on Window Manager.

**Table 9.11. Names of individual windows and dialogs.**

Grms	Grms*.PlotWindow
Grms*.AddDialog	Grms*.PrintDialog
Grms*.AxisDialog	Grms*.PrintPropsDialog
Grms*.ErrorDialog	Grms*.PropsDialog
Grms*.FontDialog	Grms*.QuestionDialog
Grms*.GridDialog	Grms*.StyleDialog
Grms*.InformationDialog	Grms*.TitleDialog
Grms*.LegendDialog	



**Table 9.12. Resources controlling general appearance.**

Resource	Function	Default
Grms*.background	Default background colour for gRMS windows.	grey
Grms*.foreground	Default foreground colour for gRMS windows.	black
Grms*.fontList	Comma separated list of three fonts, the standard font, the bold font and the italic font. N.B. Due to bugs in the DEC/OSF1 version of Motif, the bold and italic fonts do not have any effect.	fixed, fixed=BOLD_TAG, fixed=ITALIC_TAG
Grms*.selectColor	Colour for selected toggle buttons.	blue
Grms*.DirList.background	Background colour for list in File Dialog, should be set the same as Grms*.Text.background.	light grey
Grms*.DirList.foreground	Foreground colour for list in File Dialog, should be set the same as Grms*.Text.foreground.	black
Grms*.FilterText.background	Background colour for "Filter" text fields, should be set the same as Grms*.Text.background.	light grey
Grms*.FilterText.foreground	Foreground colour for "Filter" text fields, should be set the same as Grms*.Text.foreground.	black
Grms*.ItemsList.background	Background colour for list in Font Dialog and Add Dialog, should be set the same as Grms*.Text.background.	light grey
Grms*.ItemsList.foreground	Foreground colour for list in Font Dialog and Add Dialog, should be set the same as Grms*.Text.foreground.	black
Grms*.Text.background	Background colour for text fields.	light grey
Grms*.Text.foreground	Foreground colour for text fields.	black

---

# Chapter 10. Microsoft Excel Output Channel

The Microsoft Excel™<sup>1</sup> output channel is a method for storing the results of a simulation in spreadsheet form. Each time a gPROMS simulation is executed, the output channel creates a new Microsoft Excel workbook containing the values of the variables arranged in worksheets. These data can then be plotted or manipulated using Excel's existing facilities or exported to other common data management systems.

## Enabling the Microsoft Excel Output Channel

The Microsoft Excel output channel is enabled via a specification in the SOLUTIONPARAMETERS section of the PROCESS entity. The default specification is written

```
SOLUTIONPARAMETERS
  gExcelOutput := ON ;
```

With the above specification, gPROMS will generate a temporary file called `ProcessName.xls`<sup>2</sup>. However, it is recommended that the default filename is overridden using the following specification:

```
gExcelOutput := "FullFileName" ;
```

In this case, the results will be stored in `FullFileName.xls`. Where `FileName.xls` corresponds to the full pathname of the new Excel file (e.g. `C:\My Documents\MyResults.xls`). In the latter case, if the file already exists (e.g. the process has already been executed at least once), the file will be called `FileName[2].xls`. If this file also exists, gPROMS will increment the number in brackets until a new file can be generated without overwriting an existing one.

## Format of the Microsoft Excel output

The output of the simulation is written into several worksheets within the Microsoft Excel workbook. The first worksheet, called *Details*, contains a list of units and variables. The data for each variable are stored in individual worksheets. When gPROMS executes the process, it automatically opens Excel and the output file along with a macro file (`Output.xls` in the OC directory). You can use the macros to select the worksheet that contains the data for a specific variable. Simply select the cell in *Details* that contains the name of the variable that you want and then press **CTRL+SHIFT+g**.

## Additional options

In addition to specifying the name of the result workbook when the Excel Output Channel is enabled, there are two additional options that can be specified. These limit the amount of data sent to the output channel, thus reducing the size of the workbooks generated.

Because the maximum amount of data that can be stored in a workbook is limited, it is possible for large gPROMS dynamic simulations to exceed the available space in the workbook<sup>3</sup>. In addition, since one worksheet is created for each Unit in the simulation, it is possible to exceed the maximum number of worksheets that can be stored in a workbook. These two issues can be addressed using the following optional specifications:

```
gExcelOutput := "<steady-state><UnitDepth>FullFileName"
```

where *UnitDepth* is an integer and *FullFileName* is the path to the Excel workbook as before.

---

<sup>1</sup>The facilities described in this Appendix are supported by Microsoft Excel 97 and later versions.

<sup>2</sup>This will be created in the output directory of the gPROMS execution directory used during the execution of the activity

<sup>3</sup>There are a maximum of 65536 row in a worksheet, which must accomodate the values of all (monitored) variables in the Unit at each time that gPROMS reports a value (i.e. every reporting interval, plus each time a discontinuity is detected): one row per variable per time. If there are 100 variables in a complex model and gPROMS reports 1000 values for each of these, then the capacity of a single worksheet is far exceeded.

The alternative specification:

```
gUserOutput := "gExcelOutput::<steady-state><UnitDepth>FullFileName"
```

may also be made, which makes use of the more generic output capabilities of gPROMS (see the System Programmer Guide).

The <steady-state> option instructs gPROMS to output variable values for only the final reporting time. Clearly, this will have no effect on a steady-state model, but will dramatically reduce the data sent to the Excel Output Channel (depending, of course, on the reporting interval and the length of the simulation). If this is sufficient to prevent overflows in Excel, then the <UnitDepth> option may be omitted:

```
gExcelOutput := "<steady-state>FullFileName"
```

If gPROMS is also creating too many workbooks, then this can be remedied by also specifying the <UnitDepth> option. This can only be used in conjunction with the <steady-state> option and specifies how many sub Units are combined in a single worksheet. By default, a separate worksheet is created for each gPROMS Unit irrespective of the depth at which the unit is specified in the model hierarchy. In order to limit the Unit depth for which a new Excel worksheet is created, the <UnitDepth> option can be specified along with <steady-state>, where *UnitDepth* is any integer value. For example:

```
gExcelOutput := "<steady-state><2>C:\gPROMS Projects\MyResults.xls"
```

Stores the data from the end of the simulation in C:\gPROMS Projects\MyResults.xls, with one worksheet for each Unit in the top or second level in the hierarchy.

Suppose we have the following Model structure:

```
MODEL JacketedReactor
  UNIT
    Reactor AS CSTR
    Jacket  AS CoolingJacket
    ...
  END

MODEL CSTR
  UNIT
    Kinetics AS KineticExpressions
    ...
  END
```

And the Process has one instance of JacketedReactor:

```
PROCESS
  UNIT R101 AS JacketedReactor
END
```

Then all of the variables in R101 will be stored in a single worksheet. A second worksheet will be used to store all of the variables in R101.Reactor, along with all of the variables in R101.Reactor.Kinetics (and any further sub Models). The variables in R101.Jacket (and any sub Models) will all be stored in a third worksheet.

Of course, if *UnitDepth* is too small and there are many sub Models with many variables, then the capacity of a single worksheet may be exceeded, so the value chosen will depend on the size of the model.

In general, the following rules apply to the value of *UnitDepth*:

- $UnitDepth \leq 0$ : a value of 1 is assumed;
- $UnitDepth = 1$ : only top-level Units generate new worksheets—in the above example only one worksheet would be generated, containing all variables from the R101 Unit and all of its sub Units;

- *UnitDepth* = n: all units to a depth of n generate new worksheets—each worksheet contains all of the variables from the Unit and all of its sub Units;
- if *UnitDepth* is greater than the actual depth of a particular Unit, then all Units generate new worksheets—in the above example, specifying *UnitDepth* = 10 is the same as not supplying the option at all: all Units to any depth will generate new worksheets.

Finally, if sending only the results at the final simulation time provides too little data and a convenient value for the ReportingInterval cannot be found, the results can be sent at specific times only by using the RESETRESULTS task. In the following example, the results will be sent to the Excel Output Channel only at simulation times 1 and 5.

```
SCHEDULE
  SEQUENCE
    CONTINUE FOR 1
    RESETRESULTS gExcelOutput
    CONTINUE FOR 4
  END
```

## Using the graph generation macro

A second macro is available that can generate simple 2-D plots. From anywhere in the workbook, press **CTRL+SHIFT+n** to start the macro. A window will then appear with a list of available units and variables. Selecting a variable and pressing the Add button either generates a new worksheet containing a copy of the data and the 2-D plot (for scalar variables) or brings up an option window (for arrays and distributions). If a variable depends on a number of different domains, one must be selected to be plotted on the abscissa by pressing the corresponding radio button. For the remaining domains, appropriate values must be entered into the corresponding boxes. Once all domains have been specified, pressing the Add button will generate the graph. You can plot multiple instances of the same variable in the same graph (e.g. the mole fractions of all components) simply by entering a new value for the component index and pressing Add. However, at present, variables with different names cannot be plotted on the same graph. It is, of course, a simple procedure to copy the required data into a new worksheet and to create the graph manually.

---

# Chapter 11. gPLOT Output Channel

Apart from the gRMS and Microsoft Excel output channels, gPROMS provides a general purpose ASCII output channel, called *gPLOT*, that can be used to interface with any other software. If

```
gPLOT := ON ;
```

is specified in the *Solutionparameters* section of a Process entity, then gPROMS will create an ASCII text file called *ProcessName.gPLOT* where *ProcessName* is the name of the Process being executed. gPROMS ModelBuilder will automatically load the gPLOT file into the *Results* Entity group within the Case folder. The file can easily be exported for use in applications outside ModelBuilder using the Export tool. Alternately, the name of the gPLOT file can be specified explicitly by using the alternative declaration in the *Solutionparameters* section

```
gPLOT := <Filename> ;
```

In this file, gPROMS will generally record results:

- at the initial time;
- just *before* each discontinuity;
- just *after* each discontinuity;
- at the regular recording interval specified by the user.

Each such result set will contain the corresponding value of the simulation time, followed by the values of all recorded variables. Each entry is on a separate line.

**Table 11.1. Format of gPLOT files**

Line 1	The number of variables being monitored (n).
Next n lines	The names of variables in order.
Next n+1 lines	'0' on the first line (the initial simulation time), followed by the initial values of the variables on the subsequent n lines.
Next n+1 lines	A value of the simulation time on the first line, followed by the values of the variables at this time on the subsequent n lines.  The above repeated at every reporting time, and before and after every discontinuity.
Last line	A terminator (-1.00000E+09).