Aspen Custom Modeler 2004.1

Examples Guide

aspentech
driving process profitability

Who Should Read this Guide

This Examples Guide contains a general overview of ACM functionality and also more complex and extensive examples of using Aspen Custom Modeler.

Contents

Introducing Aspen Custom Modeler

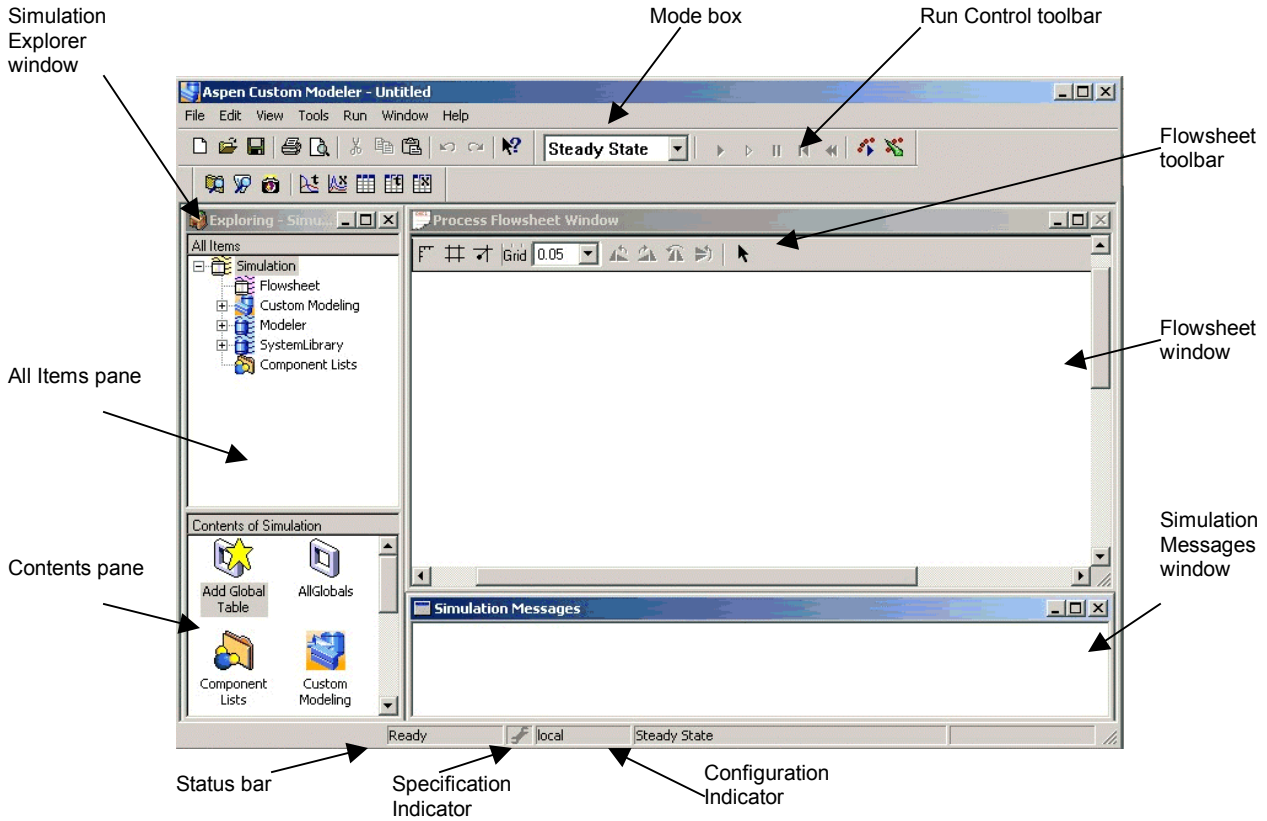
Aspen Custom Modeler 2004.1 (ACM) is an easy-to-use tool for creating, editing and re-using models of process units. You build simulation applications by combining these models on a graphical flowsheet. Models can use inheritance and hierarchy and can be re-used directly or built into libraries for distribution and use. Dynamic, steady-state, parameter estimation and optimization simulations are solved in an equation-based manner which provides flexibility and power.

ACM uses an object-oriented modeling language, editors for icons and tasks, and Microsoft Visual Basic for scripts. ACM is customizable and has extensive automation features, making it simple to combine with other products such as Microsoft Excel and Visual Basic. This allows you to build complete applications for non-experts to use.

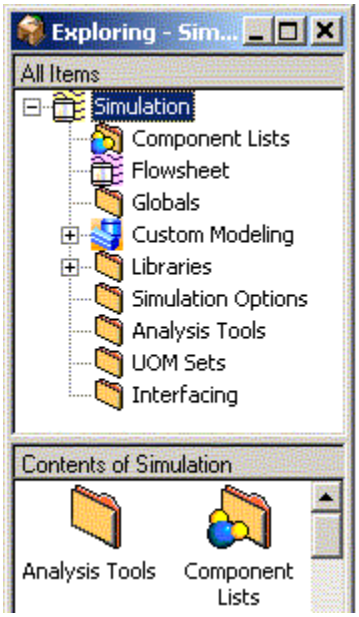
1 Application Overview

Understand the Default Windows

The names and layout of the items in the default windows are:



Simulation Explorer



The Simulation Explorer:

- Displays the main elements of a simulation, such as models, parameters, ports, procedures, streams, tasks, and variables.

Note: These components are known as types. Types can be either defined in the modeling language.

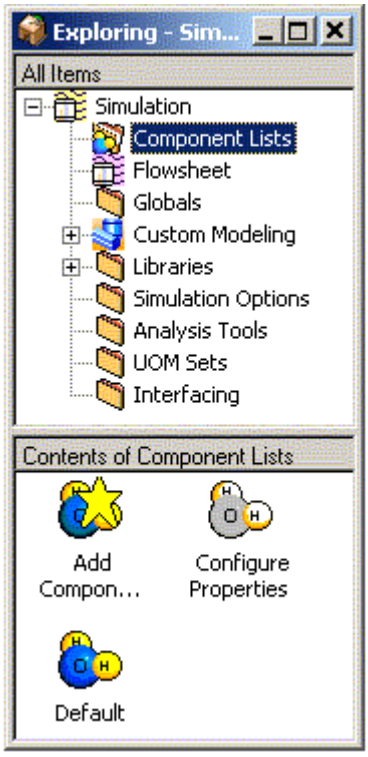
- Enables you to edit and create types

At the top level in the Explorer is the Simulation folder. This is a container for the contents of the Simulation Explorer which are:

- Component Lists folder
- Flowsheet folder
- Globals folder
- Custom Modeling library
- Libraries folder
- Simulation Options folder
- Analysis Tools folder
- UOM Sets folder
- Interfacing folder
- Diagnostics folder (if available)

To expand or collapse folders in the Simulation Explorer, click the plus sign (+) or the minus sign (–) before the folder name.

Component Lists Folder



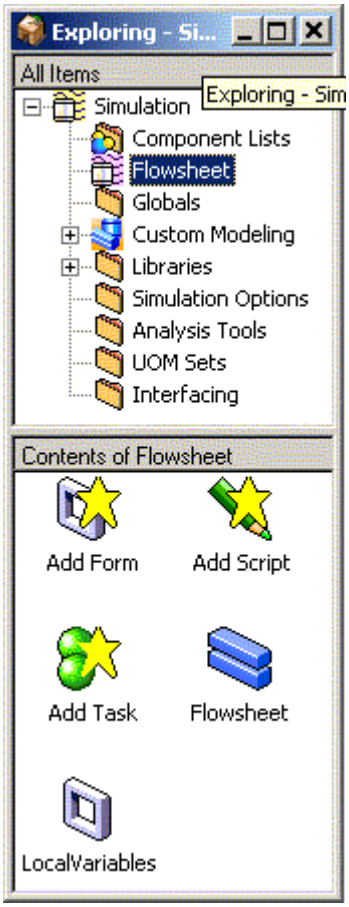
The Component Lists folder enables you to:

- Configure properties for your simulation
- Define a list of component names for your simulation
- Access existing component lists or sets in the current simulation

Flowsheet Folder

The Flowsheet folder contains information for the blocks, streams and structure instances in the current flowsheet. Blocks and streams can contain scripts, tasks, and forms, structure instances can contain scripts and forms.

Tip: To view scripts, tasks, and forms for a particular block or stream, click the block or stream in the All Items pane of the Simulation Explorer and look in the Contents pane.



From the Flowsheet folder, you can:

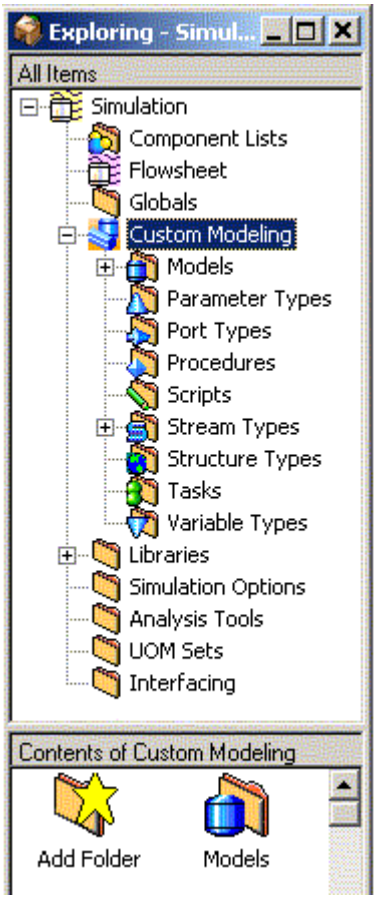
- View all the blocks and streams used in the current flowsheet
- View all instances of structure types in the simulation
- Define equations that contain variables from different blocks on your flowsheet by double-clicking the Flowsheet equals icon in the Contents pane.
- Display all the local variables for the flowsheet by clicking the LocalVariables table icon.
- Display the scripts, tasks, and forms for the whole flowsheet
- Create your own flowsheet tasks, scripts, and forms, by clicking the Add Task, Add Script, and Add Form icons.

Globals Folder

The Globals Folder allows you to access forms displaying global variables as well as adding forms.

Custom Modeling Library

The Custom Modeling library consists of folders that contain Add icons to enable you to develop the components for your simulation. The Stream Types folder also contains built-in stream types.



From the Custom Modeling Library you can:

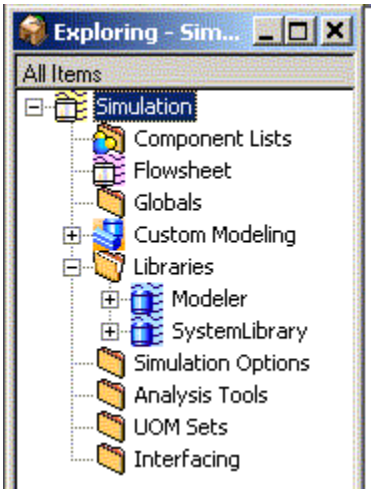
- Add blocks and streams to your simulation by dragging and dropping icons from the Simulation Explorer onto the flowsheet. Blocks and streams can also be defined in the modeling language.
- Add your own folders to the Custom Modeling library by double-clicking the Add Model Folder icon in the Contents pane:

- The Custom Modeling library folders are:

Folder	Contents
Models	All the models in the current simulation, including their associated icons, scripts, forms, and tasks, plus an Add Model icon.
Parameter Types	All the parameter types in the current simulation, plus an Add Parameter icon.
Port Types	All the port types in the current simulation, plus an Add Port Type icon.
Procedures	All the procedures in the current simulation, plus an Add Procedure icon.
Stream Types	All the stream types in the current simulation; the in-built stream types, Connection and ControlSignal; and an Add Stream Type icon.
Structure Types	All the structure types in the current simulation including their associated scripts and forms.
Tasks	All the callable tasks in the current simulation, plus an Add Task icon.
Variable Types	All the variable types in the current simulation, plus an Add Model icon.

Libraries Folder

The Libraries Folder allows you to access any ACM libraries that are attached to your simulation. In ACM 2 libraries are attached by default.



System Library

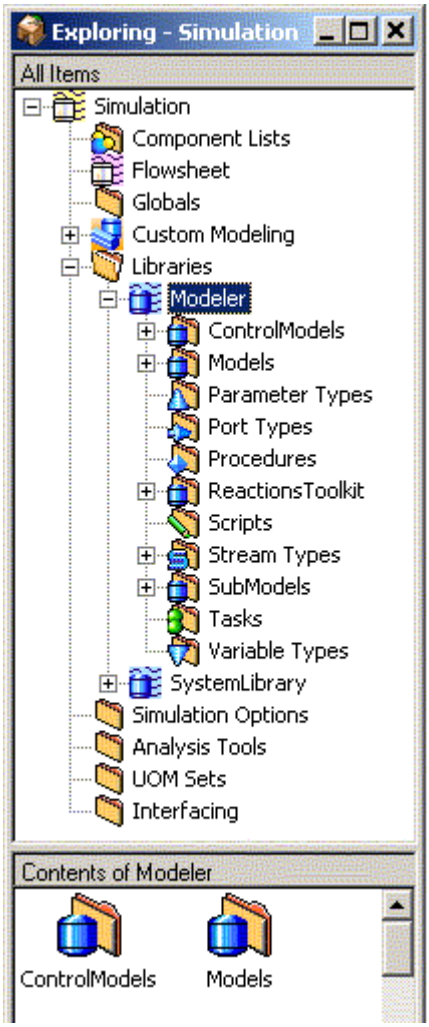
The SystemLibrary library contains built-in types for parameters, ports, procedures, streams, tasks, and variables.

You do not need to access this library directly. It is used by item in other libraries and when you compile your own models.

Modeler Library

The Modeler library contains predefined control models, variable types, port types, string parameters and physical properties procedure calls. It also contains models and structures used in the Aspen Reactions Toolkit.

You can use the items in this library in your own simulations.

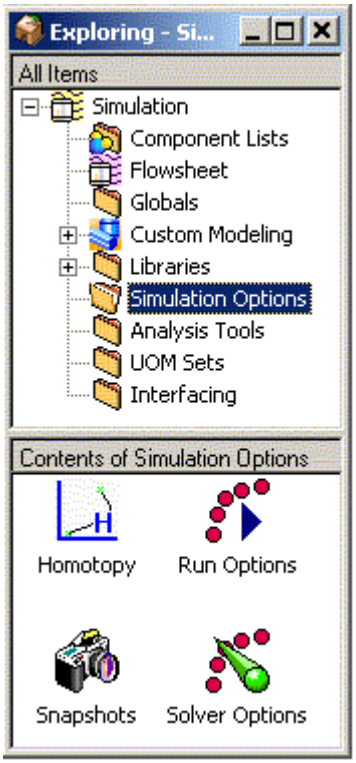


Simulation Options Folder

The Simulation Options folder allows you to access facilities that enable you to control your simulations.

These are:

- Homotopy which can be used to aid convergence
- Run Option where you can change run mode and dynamics simulation settings
- Snapshots where you can load results from previous simulations
- Solver Options where you can alter solver settings

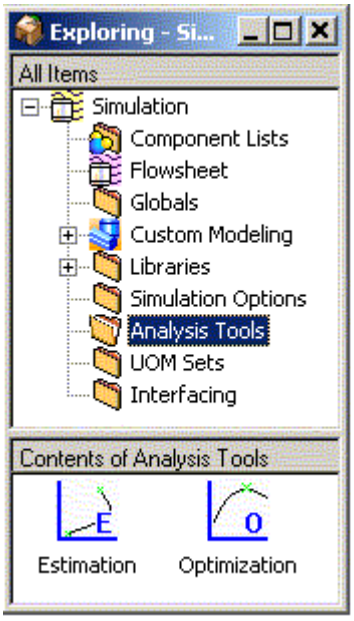


Analysis Tools Folder

The Analysis Tools folder allows you to access facilities that enable you to control some special run modes.

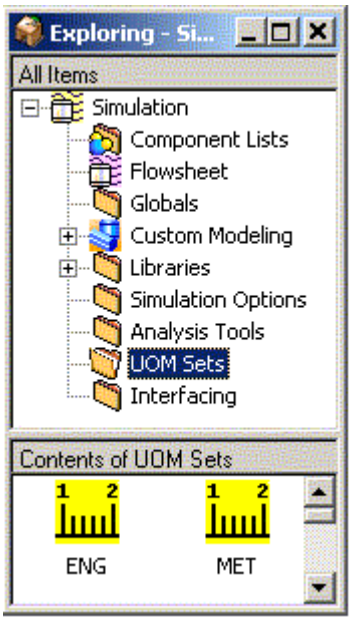
These are:

- Estimation
- Optimization



UOM Sets Folder

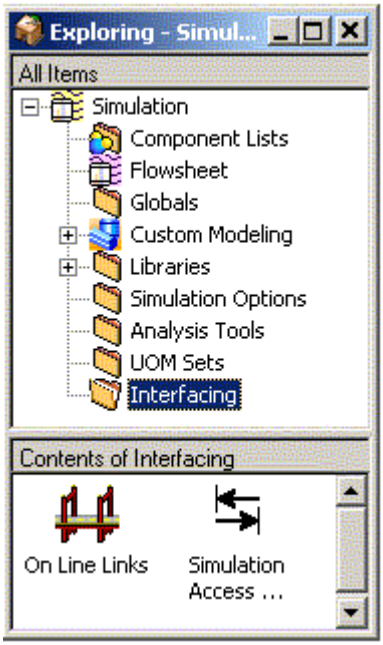
The UOM Sets folder allows you to view the UOM sets defined for the simulation and to change which is the set currently being used.



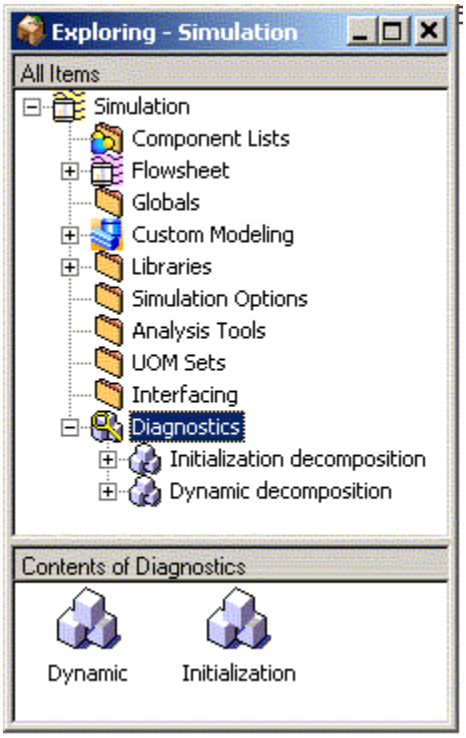
Interfacing Folder

The Interfacing folder allows you to access facilities to interface external systems to ACM. These are:-

- On Line Links allows you to interface with an OPC server
- Simulation Access Extensions allows a programming interface



Diagnostics Folder



The Diagnostics folder contains the available decomposition information, that is, groups of equations with their associated variables, for the current simulation.

It only appears after you have run a simulation. This folder can contain 1 or more decomposition nodes depending on the run mode. For steady-state, a steady-state decomposition will be present; for initialization an initialization decomposition will be present and for dynamics an initialization and a dynamic decomposition will be present. The decomposition nodes contain group nodes, which contain equations that are solved when the group is solved. The groups are solved sequentially.

You can use this information to diagnose your simulation if it fails to converge. You can expand the icons by double-clicking them, to view the groups of equations, represented by 3-D box icons, and their associated equations and variables:

If an equation group is torn, the icon has a blue tear in it, and if the group is not converged, the icon has a red cross through it:

You can choose to show lists of variables in groups eliminating the display of equations using the "Show variables in groups" item on the Diagnostics node context menu.

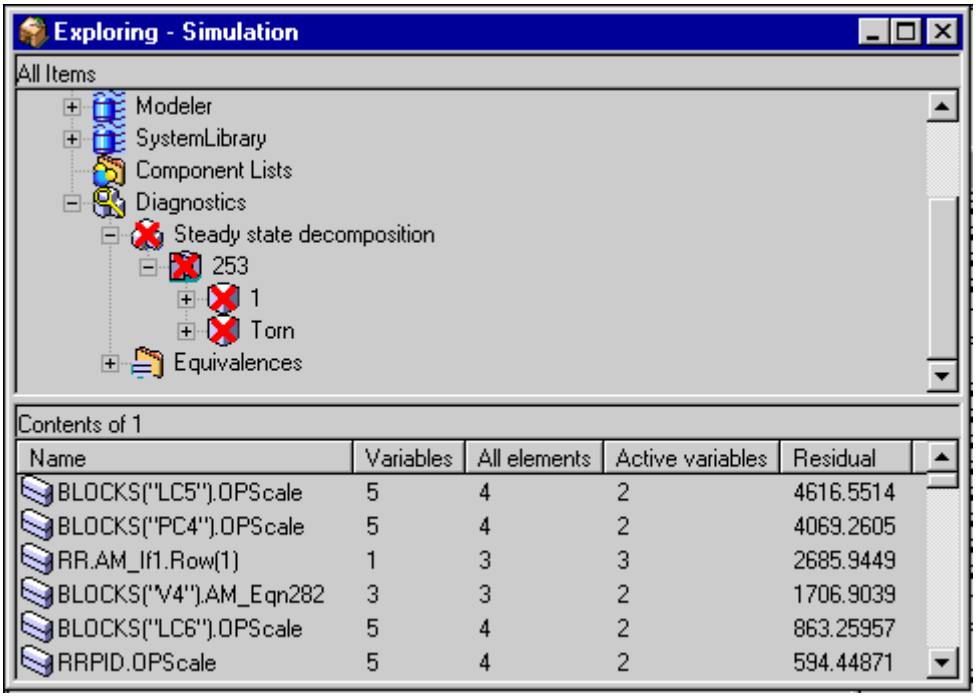
Variables which have an indeterminate value are labeled with a "?" overlaying the variable icon.

Tips:

- The Diagnostics folder appears only after you have run a simulation.
- To access properties and options for the Diagnostics folder, in the All Items pane of the Simulation Explorer, click Diagnostics with the right mouse button.

Diagnostic Folder Showing Failed Initialization Run

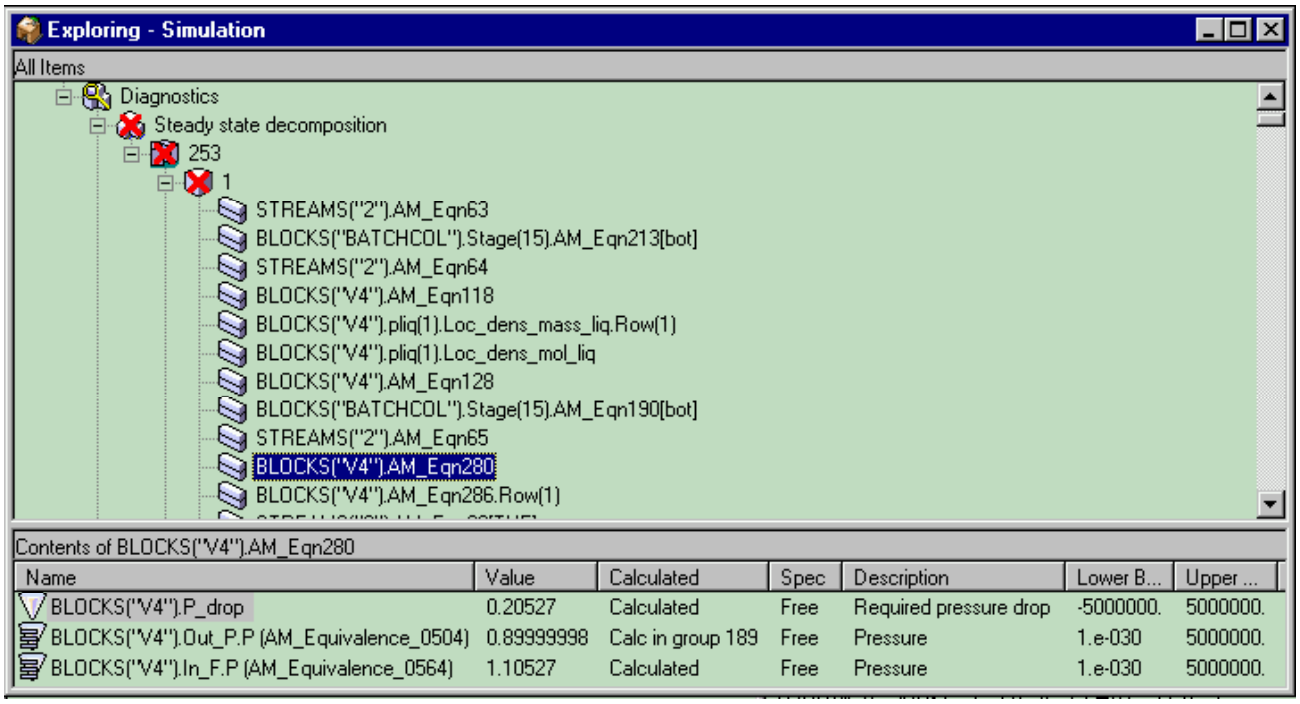
In this example, an initialization run has failed to converge in torn group 253 sub-group 1 (Torn groups are shown with a blue tear in it). The diagnostics node has been set to show only the first unconverged group using its context menu item "Show first unconverged only". This is useful for large simulations with large numbers of groups which may take some time to display. The list of equations in the group gives information about the equations including the residual. Residuals not close to 0 show an equation which has not converged:



Name	Variables	All elements	Active variables	Residual
BLOCKS("LC5").OPScale	5	4	2	4616.5514
BLOCKS("PC4").OPScale	5	4	2	4069.2605
RR.AM_If1.Row(1)	1	3	3	2685.9449
BLOCKS("V4").AM_Eqn282	3	3	2	1706.9039
BLOCKS("LC6").OPScale	5	4	2	863.25957
RRPID.OPScale	5	4	2	594.44871

Variables in an Equation

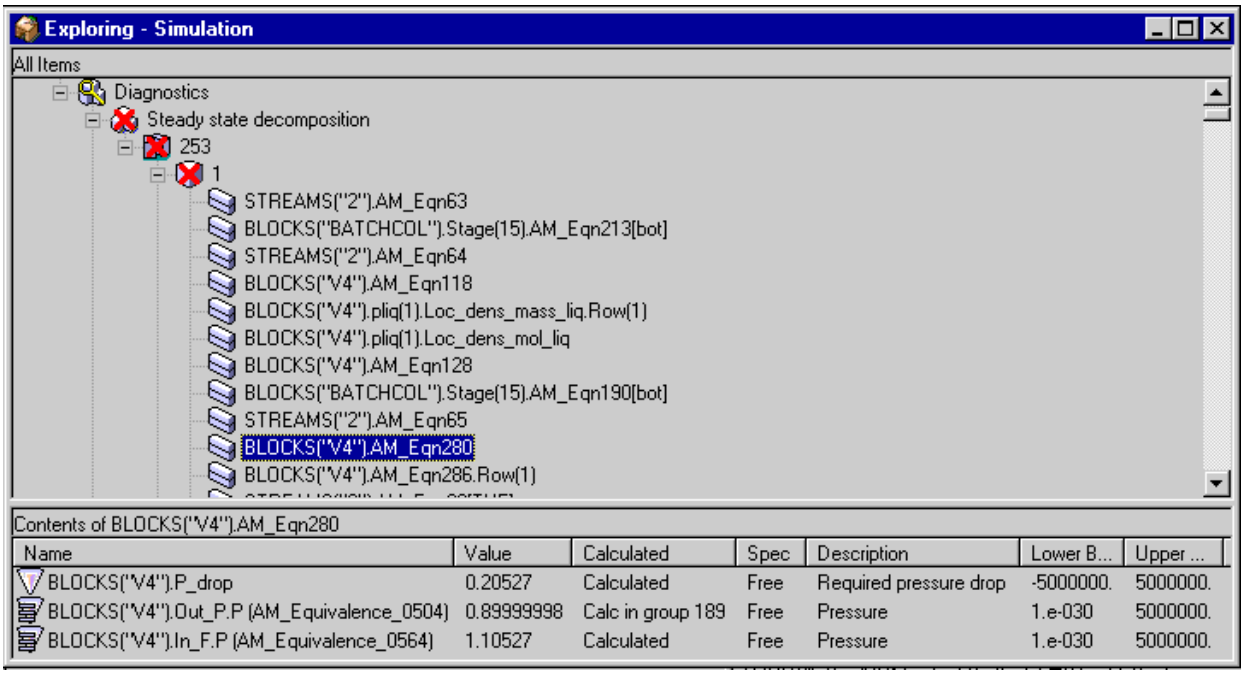
In this example, the variables in an equation are displayed. This gives detailed information about the variables, including where the value was calculated. For example, the last variable was calculated in group 253 sub group 1:



Double-clicking on this variable, or using the Find Group context menu item, displays another Explorer window for that group so you can look at the other variables and equations in that group.

Variables as an Equivalence

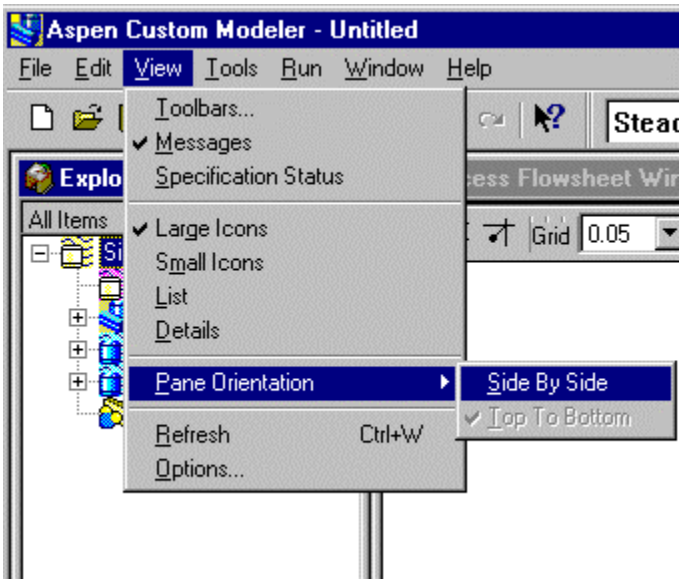
An equivalenced variable is illustrated by the 3 lines on the icon and the equivalence name in brackets. An equivalence is where the system has decided that several variables are equivalent and can be eliminated from the simulation:



Using the Find Equivalence option on the context menu for the variable will open another Explorer window showing a list of all variables in that equivalence. You can access all equivalences from the Equivalences node under Diagnostics.

Change the Simulation Explorer Pane Orientation

The default pane orientation displays the Simulation Explorer with the All Items pane above the Contents pane:



To change pane orientation so that the All Items pane is next to the Contents pane:

1 Make sure that the Simulation Explorer window is active by clicking in it.

2 From the View menu, point to Pane Orientation and then click Side by Side.

Sort Items in the Simulation Explorer

To sort items of a node in the Simulation Explorer:

- In the Contents pane of the Simulation Explorer:

To	Do this
Sort by name	Click the Name heading.
Sort by type	Click the Type heading.

Text Editor

The text editor appears when you create or edit a task or script, or edit Flowsheet or Model text. To edit one of these types, you double-click the equals sign for the type in the Contents pane of the Simulation Explorer:

The Aspen Custom Modeler Text Editor provides color highlighting to make the text easier to understand:

- Commented out text is green.
- Keywords are blue.
- String text is magenta.

Note: In the model editor, all commented out text appears highlighted in green. However, note that the text editor does not highlight commented information after a nested comment.

Keyboard Shortcuts

You can use keyboard shortcuts in the Text Editor to speed up your work.

 Shortcut keys for moving around a document .

 Shortcut keys for editing a document .

Use Text Editor Shortcut Keys to Move Around in a Document

To move about a document in the Text Editor, use these shortcut keys:

To move	Press
The mouse cursor	Left mouse button
One word to the left	CTRL+LEFT ARROW
One word to the right	CTRL+RIGHT ARROW
To the beginning of paragraph	CTRL+UP ARROW
To the end of paragraph	CTRL+DOWN ARROW
To the beginning of the line	HOME
To the end of the line	END
One word to the right	CTRL+R
One word to the left	CTRL+L
To the beginning of the visible text	CTRL+U
To the end of the visible text	CTRL+D
Down a page	PAGE DOWN
Up a page	PAGE UP
To the beginning of the text	CTRL+HOME
To the end of the text	CTRL+END
To the beginning of the visible text	CTRL+PAGE UP
To the end of the visible text	CTRL+PAGE DOWN

Note: To invoke a pop-up menu, click the right mouse button.

Use Text Editor Shortcut Keys to Edit a Document

To change a document in the Text Editor, use these shortcut keys:

To	Press
Change write mode	INSERT
Delete one symbol to the right of the cursor	DELETE
Delete one symbol to the left of the cursor	BACKSPACE
Delete current line (doesn't clear selection)	CTRL+Y
Undo last action	CTRL+Z
Cut the selection	CTRL+X
Copy the selection	CTRL+C

Paste the selection	CTRL+V
Select All	CTRL+A
Find	CTRL+F
Replace	CTRL+H
Goto Line Number	CTRL+G
Copy the selection	CTRL+INSERT
Paste the selection	SHIFT+INSERT
Duplicate	SHIFT+CTRL+INSERT
Clear	CTRL+DELETE
Cut the selection	SHIFT+DELETE
Undo	CTRL+BACKSPACE
Search Next	F3

If you know the shortcut keys to move the insertion point, you can select text using the same key combination while holding down SHIFT. For example, the END key moves the insertion point to the end of the line, and SHIFT+END selects the text from the insertion point to the end of the line.

Simulation Messages Window

The Simulation Messages window displays messages about loading an input file, compiling types, checking specification and all the normal diagnostic output from the solvers during the solution process.

In the Simulation Messages window, you can click the right mouse button to:

- Pause or scroll back simulation messages.
- Change the print level and output destination for a simulation.
- Clear the contents of the Simulation Messages window.

If you cannot see the Simulation Messages window, from the View menu click Messages.

Flowsheet Window

You build flowsheets by dragging icons from the Simulation Explorer and dropping them onto the Flowsheet window.

You can use either the default icon, called System, which is provided with ports, or your own customized model icons.

Blue arrows appear on the blocks where ports are available and you can drag ports around the icon to suit your particular flowsheet.

You connect blocks using the following stream types:

- In-built Connection stream type.
- Your own created stream types.
- In-built ControlSignal stream (for connecting control variables).

Note: You can make multiple connections to one connection point on a block.

Icon Editor

The Icon Editor is used to create your own icons for a model and appears when you click Add Icon in the Contents pane of the Simulation Explorer:

You can also choose whether icons can appear as two- or three-dimensional shapes.

To add ports, you drag and drop model ports to the desired position on the icon perimeter.

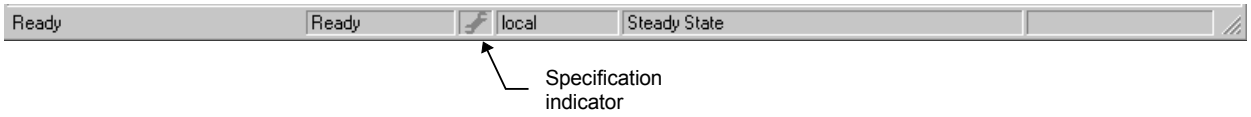
If you cannot see Add Icon, look at the All Items pane of the Simulation Explorer, make sure the Custom Modeling folder or your required library is expanded, expand Models, and then click the required model. Add Icon will then appear in the Contents pane.

Status Bar

The Status bar provides a quick means of checking the status of your simulation. Several types of indicator enable you to check different aspects of a simulation.

The Specification Indicator

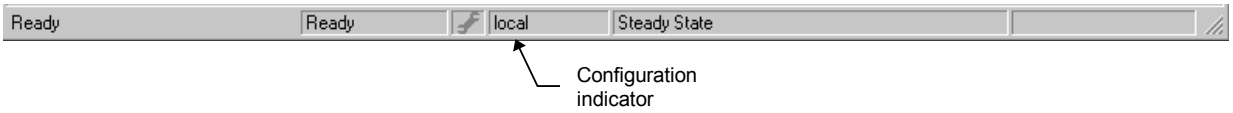
The specification indicator on the Status bar enables you to check the specification status of your simulation as you work:



To access further information contained in the Status window, double-click the specification indicator.

The Configuration Indicator

The configuration indicator enables you to see whether or not your simulation is using a remote server:



If you are running simulations on the local machine, the indicator shows Local as the current configuration. If you are running simulations on a remote server, the indicator shows the name of the configuration that it is currently using.

Status Window

To display the Status window, double-click the specification indicator on the Status bar. From the Status window, you can:

- Launch the Variable Find with predefined options.
- Check for structural singularities.
- Create a flowsheet script of all the specification changes from the base case.
- Initialize all the state variables or all the derivatives to zero (steady state conditions).
- Launch Specification Analysis to obtain a detailed analysis of the specification status of variables in your simulation, and recommendations on which variables to change to solve the simulation.

Specification Analysis

The Specification Analysis dialog box gives you a detailed analysis of the variables in your simulation, and recommendations on which variables to change to solve the simulation. For example, if your simulation is underspecified by 10 variables (which means that you need to change the specification status: Fixed, Free, or Initial, of 10 variables), Specification Analysis recommends 10 suitable variables and enables you to fix them.

Under some circumstances, the simulation cannot be solved because the specification status of the wrong variables has been set to Fixed. In this case, Specification Analysis recommends which variables should be changed to Free.

Also, if some model equations are causing solution difficulties, these will be presented in the Details dialog box.

To display the Specification Analysis dialog box:

- From the View menu, select Specification Analysis.

  – or –

- From the Status window, click Analyze.

To perform a specification analysis on the current run, in the Specification Analysis dialog box click Analyze Now.

You can obtain further information, as well as details of any equations which may be causing problems, by clicking the Details button.

Variable Find

Variable Find is used to:

- Find and list the variables within a simulation.
- Provide an easy way of generating scripts and customized tables.

You can access Variable Find in several ways, including the following:

- Double-click the specification indicator on the Status bar, and then click one of the buttons, All, Fixed, Initial, or Free.
- From the Tools menu, click Variable Find.

Set the Working Folder

The working folder is used to store temporary files created when you run a simulation. If these temporary files are deleted from this folder, any saved snapshots and plotting results will be lost.

Note: When you change the working folder, any open simulation closes. To avoid this, set the working folder before you open a simulation.

To set the working folder:

1. From the File menu, click Set Working Folder.
2. In the Browse for Folder dialog box, click the folder that you want to be your default working folder, and then click OK to close the dialog box.

Capture Screen Layout

You can save the layout of the GUI at any time by using the Capture Screen Layout option available under the Tools menu. This prompts you for a script name. This script is created under the Flowsheet node in the Explorer.

This script contains automation commands to do the following:

- Close all form and explorer windows which are currently open.
- Restore the main flowsheet window to the view current when the script was saved.
- Restore any hierarchy flowsheet window to their saved size, position and view.
- Restore all saved form and explorer windows to their saved size and position.

To restore the GUI to the saved layout, double-click on the script in the Explorer window.

Opening and Saving Simulations

Open a Library

To open a library:

1. From the File menu, click Open Library.
2. In the Open dialog box, type or select the name of the library file that you want to use for your simulation.
3. Click Open to load the library.
   The file appears as a library node in the Simulation Explorer.

Notes:

- If a library is referenced in a language text file, the library is opened automatically when loading the file.
- The library is loaded from: the directory specified in the Libraries statement in the language text file (if any); or from the same directory as the language text file; or from the Aspen Custom Modeler libraries directory.
- After you have opened a library, you cannot close or remove it from the user interface. However, you can remove it from a language text file.

Caution: If the language text file contains references to a type in the library that you have removed, compilation errors may result.

Open the Polymers Plus Library

The Polymers Plus library contains all the polymer models and streams. You must open the library before you can build a polymer flowsheet. To open the library:

1. From the File menu, click Open Library.
2. From the list provided, select Polymers Plus Library.

Open a Simulation

To open an existing simulation:

1 From the File menu, click Open.
The Open dialog box appears, showing the current working folder.

2 Move to the folder that contains the required .acmf or .acmd file.

3 To open the file, double-click the file name.

Import a Flowsheet

You can use the Import Flowsheet command to import a flowsheet that is stored in an external language file (for example, to load a control scheme developed for a previous problem) into the current simulation.

Notes:

- The following objects are imported from the external file:
  - Blocks
  - Streams
  - Scripts
  - Tasks
  - Form definitions
  - Forms

  However, component lists and constraints in the external file are not imported. All the types referenced by the imported flowsheet must already be accessible from the current simulation.
- If ControlSignal streams are imported and connected to variables in the existing flowsheet, the degrees of freedom in the simulation are automatically adjusted by freeing the manipulated variables.

To import a flowsheet into the current simulation:

1 From the File menu, click Import Flowsheet.

2 In the Import dialog box, locate the file to be imported. On this dialog you can also choose to import the flowsheet into a hierarchy flowsheet. The contents of the file chosen are then added to the hierarchy rather than the main flowsheet. By default it is added to the main flowsheet.

3 In the Resolve Name Conflicts dialog box, click the name of the object that you want to change and then do any of the following:

To	Click
Remove an object from the file that is being imported	Delete
Rename an object that is being imported, to avoid conflict with existing objects	Click Rename, or Add Prefix, or Add Suffix
Replace existing objects in the	Replace

current simulation with those of the same name in the file that is being imported	
Undo the last operation carried out in this dialog box	Undo

4 When you have resolved all the name conflicts, click OK to accept your changes.

Printing Flowsheets

The **File | Page Setup** menu option available on any flowsheet window allows you to ask for multi-page printing of flowsheets.

Use the Page Setup dialog to specify:

- How many sheets to use for the height and width of the flowsheet.
- Paper size.
- Orientation.
- Margin size.

Import Types into a Simulation

To insert existing types into the current simulation:

1 From the File menu, click Import Types.
The Open dialog box appears, showing the current working folder.

2 Move to the folder that contains the required .acmf file.

3 To insert the file into the current simulation, double-click the file name.

Note: When you insert the .acmf file, make sure the file contains only information relevant to the Custom Modeling library. This information includes definitions for models, stream types, parameter types, variable types, port types, procedures, and tasks. Flowsheet, Properties, and Options sections in the imported file are ignored.

Save a Simulation

To save a simulation:

- From the File menu, do one of the following:

To	Click
Save the simulation in the format in which it was opened	Save
Supply a new name for the simulation, copy the simulation, or save it in a different format. **Note:** The contents of the working directory are saved with the new file.	Save As

Note: Linked and embedded objects are not saved when you save a simulation in modeling language format (.acmf file.)

Export Parts of a Simulation

You can export as an .acmf file either of the following:

- A folder that contains a group of types, for example, models or variable types.
- An individual type, for example, a particular model.

To export a simulation, or parts of a simulation:

1 In the Simulation Explorer, click the part that you want to export, then click the right mouse button and click Export.

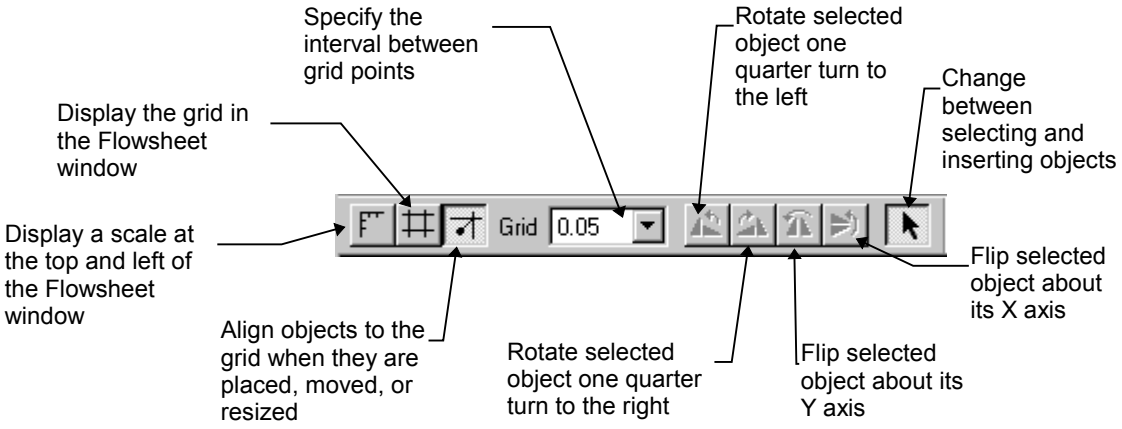
2 In the Export dialog box, type a name for the file in the File Name box and then click Save.

Note: Linked and embedded objects are not saved when you export a simulation in modeling language format (.acmf file).

Building Flowsheets

Use the Flowsheet Toolbar

Use the Flowsheet toolbar to manipulate objects in the flowsheet window.



If you can't see the Flowsheet toolbar, click the View menu, and then click Toolbars, and check the box next to Process Flowsheet.

Use a Flowsheet as Wallpaper

To use your flowsheet as wallpaper or background for your simulation:

1 Make the flowsheet window active by clicking in it.

2 From the Window menu, click Flowsheet as Wallpaper. The flowsheet is now the background simulation.

Note: This behavior persists between sessions. It is only available for the main flowsheet and not for hierarchy flowsheets.

Create a Flowsheet

To create a flowsheet:

1 In the All Items pane of the Simulation Explorer, make sure the Custom Modeling folder or your required library is expanded, and then click Models.
A list of existing models and the Add Model icon appear in the Contents pane.

2 Drag and drop the required model icon onto the flowsheet, using the mouse buttons as follows:

To	Use
Insert one block only	Left mouse button
Choose whether to insert one or many blocks If you select Insert Many, the cursor changes to a cross-hair and you can insert many blocks on the flowsheet. To stop inserting blocks, click the Select Mode button on the Flowsheet toolbar.	Right mouse button

To use a different icon, in the Contents pane, double-click the required model. Drag and drop the required icon onto the flowsheet.

3 Repeat steps 1 and 2 for all the models that you want to connect to define the current simulation.

4 In the All Items pane, click Stream Types.

Note: You can use in-built streams by double-clicking Stream Types and using Connection for blocks and ControlSignal for control variables.

5 In the Contents pane, drag and drop the required stream onto the flowsheet window.
Blue arrows appear on the blocks where ports are available.

6 To use the Universal ports provided, click a blue arrow and drag the selected port to the desired position on the icon perimeter. If the port has

more than one model port attached, choose the port name from the dialog box that appears.

7 Click the ports on the blocks to connect them.

8 To stop making connections, click the Select Mode button on the Flowsheet toolbar.

Move Blocks and Streams

You can move the following objects on the flowsheet:

- A block.
- The name of a block.
- A stream name.
- Multiple blocks and streams.

If you want to move a stream, you have to change the connectivity of the stream.

Move a Block

To move a block:

1 In the flowsheet window, press and hold down the mouse button on the block (but not on the block ID) that you want to move.
The outline of the block is highlighted, the mouse pointer changes to the move shape, and a text box appears, showing name of the block.

2 Drag the block to the location you want and release the mouse button.

To make minor adjustments to the position of a block, you can select the block and then use the arrow keys.

Move a Block Name

To move a block name:

1 In the flowsheet window, press and hold down the mouse button on the block name that you want to move.
The mouse pointer changes to the move shape.

2 Drag the name to the location you want and release the mouse button.

To make minor adjustments to the position of a block name, you can select the name and then use the arrow keys.

Move a Stream Name

You cannot move a stream name away from a stream but you can move the stream name along a stream:

To do this:

1 In the flowsheet window, press and hold down the mouse button on the stream name, until the mouse pointer changes to the move shape.

2 Drag the name to the location you want and release the mouse button.

Move Multiple Blocks and Streams

To move a group of blocks and streams:

1. In the flowsheet window, select the group of blocks and streams you want to move.
2. Hold down the mouse button on any block or stream within the region you have selected.
   The mouse pointer changes to the move shape.
3. Drag the group to the location you want, and release the mouse button.

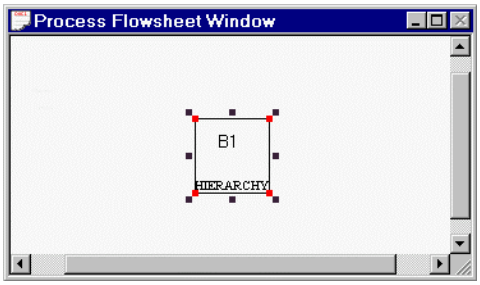
You can also select the group then use the arrow keys to move it to the new location.

Hierarchies

Hierarchies

A hierarchy is a block which can contain a flowsheet. A hierarchy can be created on the main flowsheet or on another hierarchy by dragging the Hierarchy model node from the Explorer onto the destination flowsheet.

The Hierarchy model node is located in the Models folder of all libraries and under the Custom Modeling node. Hierarchy blocks have a single icon, e.g.,

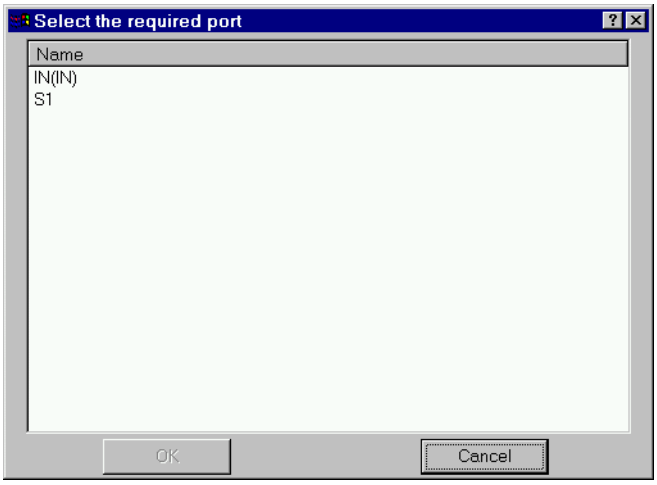


To access the contents of the hierarchy:

1. Double-click on the hierarchy block. A new Process Flowsheet window appears.
2. Edit this in the same way you edit the main flowsheet.

Hierarchy Connections

When you try to connect a stream to the inlet or outlet ports on the hierarchy block a dialog appears:



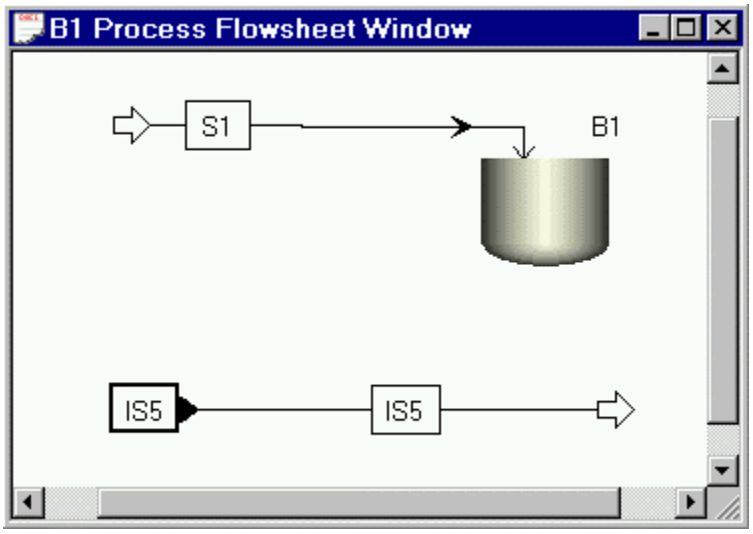
This dialog allows you to either:

- Create a new stream in the hierarchy (by selecting IN(IN) for inlets, OUT(OUT) for outlets);
- or connect to an existing stream, in this case stream S1.

The dialog lists all streams in the hierarchy which currently start from source blocks for a connection to an inlet port and from sink blocks for a connection to an outlet port.

If no streams terminate at sinks (or sources), a new stream is created in the hierarchy flowsheet without the dialog being presented.

This example illustrates the hierarchy flowsheet if the IN(IN) entry had been selected:



A new stream IS5 has been created which terminates at a sink block. The new stream in the hierarchy has been given the same name as the stream in the parent flowsheet but they are different streams. The full name of the new stream is B1.IS5. This stream starts at a connector symbol which contains the name of the connected stream in the main flowsheet.

Note: You cannot connect control streams in this way.

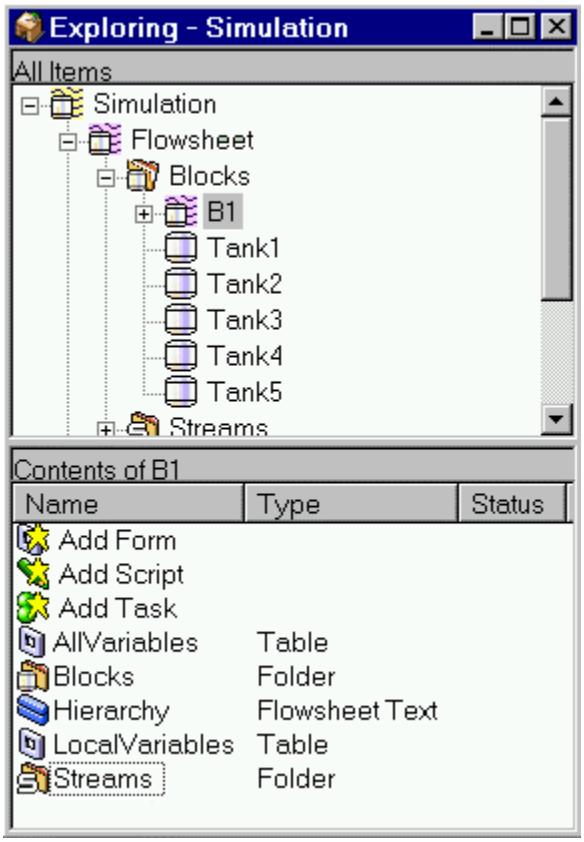
Accessing Hierarchies

To access the flowsheet for a hierarchy:

1. Double-click on the hierarchy block in its parent flowsheet or use the Expand menu item, located on the context menu for a hierarchy block.
2. A new Process Flowsheet window appears. You can edit this in the same way you can edit the main flowsheet, although hierarchy blocks cannot be made to be wallpaper.

You can also access a hierarchy from the Explorer. This allows you to edit the non-graphical contents of the hierarchy.

The hierarchy, in this case B1, appears in a Blocks folder, but has the same icon and contents as the Flowsheet:



You can access constraints, forms, scripts and tasks. Note that as these are contained within the hierarchy you can only access variables, etc., which are in blocks or streams in this hierarchy or beneath. You cannot, for example, add variables contained in a block in the parent flowsheet to a plot owned by a hierarchy.

Note: The hierarchy also contains Blocks and Streams folders containing the blocks and streams which appear on the hierarchy

flowsheet.

Importing to and Exporting from Hierarchies

You can read the contents of a simulation file (.acmf file) into a hierarchy. The main flowsheet defined in the input file then appears as the flowsheet in the hierarchy and any forms, scripts, tasks or constraints are added to the hierarchy.

Any name clashes with the existing contents of the hierarchy are flagged and resolution is allowed in the same way as for importing to the main flowsheet.

You can export the contents of a hierarchy to a simulation file (.acmf file). This takes the contents of the hierarchy and "promotes it" so that it then becomes the main flowsheet in the generated simulation file. All simulation contents outside the flowsheet context are written as normal except that the contents of kept results are edited to remove references to variables which are not contained in the exported hierarchy.

These facilities are available from the File menu Import Flowsheet and Export menu items. These bring up file dialogs which have a combo box allowing you to select which hierarchy to use. By default the action is on the main flowsheet. You can also access these facilities from the context menu of a hierarchy block node in the Explorer.

Change Stream Connectivity

You can disconnect the end of a stream from a block and then connect it to another port on the same or a different block. To change the port to which a stream is connected:

1 In the flowsheet window, click the stream and then click the right mouse button.

2 From the menu that appears, do one of the following:

To	Click
Disconnect the source end of the stream	Reconnect Source
Disconnect the output end of the stream	Reconnect Destination

Blue arrows appear on the blocks where ports are available. For example, for a feed stream, the outlet ports are highlighted.

3 Continue as you would for a new stream. Click the port to which you want to connect the stream end, or click a blank part of the flowsheet to place a feed or product.

Reroute a Stream

To reroute a stream automatically:

1 In the flowsheet window, click the stream you want to reroute.

2 Click the right mouse button on the stream.

3 Click Reroute Stream.

Tip: You can also select one or more streams and then press CTRL+J to reroute them.

Arrange Blocks on a Flowsheet

You can arrange blocks on a flowsheet in the following ways:

- Aligning two blocks.
- Positioning all the blocks on a flowsheet automatically.
- Positioning the next block automatically.
- Position the blocks one at a time to create your own layout.

Align Two Blocks

To align two blocks in a flowsheet:

1. In the flowsheet window, click the stream between the two blocks.
2. Click the stream with the right mouse button.
3. Click Align Blocks.

Tip: You can also select one or more blocks and streams and press CTRL+B.

Position All the Blocks on a Flowsheet

If the number of unplaced blocks is small or you are not concerned about the layout of the flowsheet, you can use Place All to automatically position all the blocks on the flowsheet at the same time.

To position all the blocks on the flowsheet:

1. Select a block or group of blocks and then click the right mouse button on one of the blocks.

Tip: To select a group of blocks, hold down the SHIFT key and click each block.

2. From the popup menu, click Unplace Blocks.
3. In the Unplaced Blocks dialog box, click Place All.
4. If necessary, move individual blocks and reroute streams. You can move stream segments or corners to achieve the desired routing.

Position the Next Block Automatically

If you have selected a group of blocks, the blocks can be automatically positioned on the flowsheet in a logical order. To do this:

1. Select a group of blocks and then click the right mouse button on one of the blocks.

Tip: To select a group of blocks, hold down the SHIFT key and click each block.

2. From the popup menu, click Unplace Blocks.
3. In the Unplaced Blocks dialog box, click Place Next.

   The block that logically should appear next in the flowsheet is placed in an appropriate position.
4. Click Place Next again to position each block in the group.
5. If you do not like the position of a block, move it to a different location.

As you place blocks, the streams that connect them also appear. You can move stream segments or corners to achieve the desired routing.

Position a Single Block on the Flowsheet

Place is used to position an individual block on the flowsheet. Use Place when you want to achieve a specific layout.

To place a single unplaced block on the flowsheet:

1. Click a block, and then click the right mouse button on one of the blocks.
2. From the popup menu, click Unplace Blocks.
3. In the Unplaced Blocks dialog box, click and hold down the mouse button on the name of the block.
4. Drag the block to the flowsheet and drop it where you want the block positioned.

As you place blocks, the streams that connect them also appear. You can move stream segments or corners to achieve the desired routing.

View a Flowsheet

You can change the way the flowsheet is displayed by:

- Using Pan to select a view of a flowsheet.
- Changing the Zoom level of a flowsheet.

Use Pan to Select a View of a Flowsheet

To use Pan to select a view of the flowsheet at the current zoom level:

1. In the flowsheet window, click the right mouse button.
   A full view of the flowsheet appears, with a dotted rectangle.

2 Move the rectangle to the area that you wish to zoom in on and click the left mouse button.

3 To cancel Pan, click the right mouse button.

Change the Zoom Level of the Flowsheet

To change your view of a flowsheet by using the Zoom commands:

- From the View menu point to Zoom, then click the option you require.

  – or –

1 In the flowsheet window click the right mouse button.

2 From the menu, click:

This zoom option	To
Zoom In	Zoom in
Zoom Out	Zoom out
Zoom Full	Show the full flowsheet

Work with Icons

You can perform the following actions on icons:

- Replacing an icon with a different icon
- Resizing an icon
- Rotating an icon

Replace an Icon

To replace an icon with a different icon:

1 In the flowsheet window, click the block whose icon you wish to change, and then click the block with the right mouse button.

2 Click Exchange Icon.
The icons for the block changes to the next icon in the list for the model.

You can also change the icon by clicking the block, then pressing the letter **n** to change to the next icon available for the block, or **p** to change to the previous available icon.

Resize an Icon

To resize an icon:

1 In the flowsheet window, click the block whose icon you wish to resize, and then click the block with the right mouse button.

2 Point to Resize Icon, then Shrink or Enlarge to shrink or enlarge the icon by a built-in factor.

  – or –

  Position the mouse pointer over one of the corners of the block icon until the Resize mode pointer appears:

Drag the mouse pointer until the icon is the desired size.

Rotate an Icon

To rotate an icon in a flowsheet:

1. In the flowsheet window, click the block whose icon you want to rotate and then click the block with the right mouse button.
2. Click Rotate Icon.
   A submenu appears, enabling you to rotate the icon to the right (clockwise) or left, or flip the icon around either axis (for example, to reverse flow direction).

   You can also use the buttons on the Flowsheet toolbar to rotate and flip an icon.

Name Blocks and Streams

By default, names are automatically assigned to blocks and streams as you add them to your flowsheet. You have several options for changing the way blocks and streams are named:

- Supplying your own prefixes for automatic naming.
- Switching off automatic naming and supplying your own names.
- Renaming existing blocks and streams.

Change Options for Naming Blocks and Streams

To change the naming of blocks and streams, you can do one of the following:

- Supply prefixes for automatic naming.
- Turn off automatic naming and be prompted for a name for each block and stream as you place it.

To specify the naming options:

1. From the Tools menu, click Settings.
2. Click the AutoName tab and do one of the following:

To	Do this
Supply prefixes for automatic naming	Type a prefix in the text box for Auto Block and/or Auto Stream name prefix. A sequential number is added to the prefix.
Use your own names for each block and/or stream	Clear the appropriate check boxes for Automatic Block and Automatic Stream name generation. You will be prompted for a name for each block or stream as you place it on the flowsheet.

Rename a Block or Stream

To rename a block or stream:

1. In the flowsheet window, click the block or stream you want to rename and then click with the right mouse button.
2. From the menu, click Rename Block or Rename Stream.
3. In the Input dialog box, type the new name and click OK.

Note: The name can contain up to 27 characters, but it cannot contain spaces, or non-alphanumeric characters (for example, -\ /:*?"<>|).

Delete a Block or Stream

To delete a block or stream:

1. In the flowsheet window, click the block or stream to select it and then click with the right mouse button.
2. From the popup menu, click Delete Block or Delete Stream.
3. When prompted by the Confirm dialog box, click OK.

   You can also select the block or stream, then press the DELETE key.

Update References to a Link

To update references to a link that has been renamed or moved:

1. If necessary, click in the Flowsheet window to make it the active window.
2. From the Edit menu, click Links.
3. In the Links dialog box, click the name of link for which you want to update the reference, and then click Change Source.
4. In the Change Source dialog box, click the application to which you want to refer and then click Open.
5. Click Close to update the reference and close the Links dialog box.

Work with Flowsheet Tasks

A task is a set of instructions that you can create within your AspenTech product. A flowsheet task defines a sequence of actions that take place during a dynamic simulation, for example, changing feed flow rate.

Caution: Do not change estimated or reconciled variables with a task while you are performing a dynamic estimation run; and do not change design or control variables with a task when you are performing a dynamic optimization run.

From the Simulation Explorer, you can:

- Create a new task.
- Edit an existing task.

Create a Flowsheet Task

To define a disturbance or other sequence of actions to occur during a dynamic run, you need to create a Flowsheet task.

Caution: Do not change estimated or reconciled variables with a task while you are performing a dynamic estimation run; and do not change design or control variables with a task when you are performing a dynamic optimization run.

To create a Flowsheet task:

1. In the All Items pane of the Simulation Explorer, click Flowsheet.
2. In the Contents pane, double-click Add Task.
3. In the Create Task dialog box, enter a name for the new task.
4. In the Text Editor window, enter the information for the new task.
5. From the Build menu click Compile.
   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
6. To quit the editor, click the Close button.

Note: Before the task you have created can be used in a simulation, you will need to activate the task.

Edit a Flowsheet Task

To edit a flowsheet task:

1. In the All Items pane of the Simulation Explorer, click Flowsheet.
2. In the Contents pane, double-click the task you want to edit.
3. In the Text Editor window, make the necessary changes.
4. From the Build menu click Compile.
   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
5. To quit the editor, click the Close button.

Work with Flowsheet Scripts

A script is a set of instructions, written in Microsoft Visual Basic, to automate the setup of a simulation.

Using your AspenTech product you can:

- Create a script
- Edit a script

For more information on using scripts, see the Automation Reference.

Create a Flowsheet Script

To automate actions in flowsheets, you need to create scripts. You can create scripts for a flowsheet in three ways:

- From the Simulation Explorer
- From Variable Find
- From the Status Window

For detailed information on using with scripts, see the Automation Reference.

Create a Flowsheet Script from the Simulation Explorer

To create a flowsheet script:

1 In the All Items pane of the Simulation Explorer, click Flowsheet.

2 In the Contents pane, double-click Add Script.

3 In the Create Script dialog box, enter a name for the new script.

4 In the Text Editor window, type the information for the new script. When you have finished typing your script, you can do one of the following:

To	Do this
Run the script	Click the right mouse button in the Edit window, point to Build, and then click Invoke Script. The script is saved then run. After running, the Scripting dialog box appears. Click OK to dismiss this box.
Save the script	From the File menu, click Save Text. When you close the Text Editor, you will be prompted to save any further changes.

Further information Scripts are a simple method for automating activities, such as changing parameter and variable properties. The scripting language is Microsoft Visual Basic Scripting Edition (VBScript). This is supplied automatically with your installation.

If you want to use the more powerful features of VBScript, you can obtain full documentation on the VBScript language from the Microsoft web site.

Create a Flowsheet Script from Variable Find

To create a flowsheet script from Variable Find:

1 From the Tools menu, click Variable Find, and then click Find.

2 Hold down the CTRL key, and then click each variable you want to select from the list at the bottom of the Variable Find dialog box, and then click Script.

To select all the variables in the list, press CTRL+A.

3 In the Specification Script Creation dialog box, type a name for the script and then click OK.
A new script icon appears in the Contents pane for Flowsheet.

The script shows the value and spec attributes for each selected variable. You can run the script to restore these values to what they were when the script was created.

Further information: Scripts are a simple method for automating activities, such as changing parameter and variable properties. The scripting language is Microsoft Visual Basic Scripting Edition (VBScript). This is supplied automatically with your installation.

If you want to use the more powerful features of VBScript, you can obtain full documentation on the VBScript language from the Microsoft web site.

Create a Flowsheet Script from the Status Window

To create a flowsheet script from the Status window:

1 From the View menu, click Status Window.

2 On the General Tab, click Script.
A new icon appears in the Contents pane for Flowsheet, showing the new script. This script is a record of all the changes you made from the default specification (fixed, initial, or free) as defined in the models.

Further information: Scripts are a simple method for automating activities, such as changing parameter and variable properties. The scripting language is Microsoft Visual Basic Scripting Edition (VBScript). This is supplied automatically with your installation.

If you want to use the more powerful features of VBScript, you can obtain full documentation on the VBScript language from the Microsoft web site.

Edit a Flowsheet Script

To edit a flowsheet script:

1 In the All Items pane of the Simulation Explorer, click Flowsheet.

2 In the Contents pane, click the required script, then click the right mouse button and click Edit.

3 In the Text Editor window, make the necessary changes. When you have finished changing your script, you can do one of the following:

To	Do this
Run the script	Click the right mouse button in the Edit window, point to Build, and then click Invoke Script. The script is saved and then run. After running, the Scripting dialog box appears. Click OK to dismiss this box.
Save the script	From the File menu, click Save Text. When you close the Text Editor, you will be prompted to save any further changes.

Further information: Scripts are a simple method for defining parameter and variable properties, that is, spec, value, upper and lower limits. The script

language uses Microsoft Visual Basic Scripting Edition (VBScript). This is supplied automatically with your installation.

If you want to use the more powerful features of VBScript, you can obtain help from the Microsoft web site as a free download.

Work with Flowsheet Constraints Sections

A Constraints section is used to define constraints on a simulation, for example, costs, efficiency, purity, and so on. You can also use a Constraints section to define an objective function for an optimization run.

Using your AspenTech product, you can:

- Create a flowsheet constraints section.
- Edit a flowsheet constraints section.

Create a Flowsheet Constraints Section

To create a flowsheet constraints section:

1. In the All Items pane of the Simulation Explorer, click Flowsheet.
2. In the Contents pane, double-click Flowsheet.
3. In the Text Editor window, enter the information for the flowsheet Constraints section.

Tip: You can use the Constraints section to write equations that contain variables from different blocks on your flowsheet.

4. From the Build menu, click Compile.
   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
5. To quit the Text Editor, click the Close button.

Edit a Flowsheet Constraints Section

To edit a flowsheet Constraints section:

1. In the All Items pane of the Simulation Explorer, click Flowsheet.
2. In the Contents pane, double-click the equals icon for the flowsheet:

3. In the Text Editor, make the necessary changes.

Tip: You can use the Constraints section to write equations that contain variables from different blocks on your flowsheet.

4 From the Build menu, click Compile.
The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.

5 To quit the Text Editor, click the Close button.

Specify Settings for a Simulation

To specify settings for your simulation:

- From the Tools menu, click Settings.
- From the Settings dialog box you can specify:
  - Options for naming blocks and streams.
    Font type for flowsheets, plots and tables.
    Libraries to be attached to new simulations.
  - Applications for which your AspenTech product will be available.
  - Rendition of icons on a flowsheet.
    Properties of a plot.
    Preferences for preparing, saving, and running a simulation.
    Preferences for handling snapshots.
    Debug option for external routines.
  - Applications for which only fixed variables will be available when connecting a control variable.
  - Default behavior when you double-click script items in the explorer (it can be run or edit).

Work with Inserted Objects

You can insert objects created with a Microsoft® Windows® application, such as a Microsoft Excel spreadsheet or a Microsoft Word document, into a flowsheet.

You can also edit objects that you have inserted into your flowsheet.

Click the following links to display information about inserting and editing objects:

- Inserting an object into a flowsheet.
- Editing an inserted object.

Insert an Object into a Flowsheet

To insert an object created with a Microsoft Windows application, such as a Microsoft Excel spreadsheet or a Microsoft Word document, into your simulation:

1 From the Edit menu, click Insert Objects.

2 In the Insert Object dialog box, click the required options and then OK to close this dialog box.
The Microsoft Windows application is now part of the flowsheet.

Edit an Inserted Object

To edit a Microsoft Windows object that has been added to the flowsheet:

1 In the Flowsheet window, click the object you want to edit.

2 At the bottom of the Edit menu, point to the object type and then click Edit.
You can now make the necessary change to the object and save them.

Cut, Copy and Paste of Flowsheet Items

You can manipulate flowsheet items using cut, copy and paste facilities. These are accessed using the standard keys ctrl X (Cut), ctrl C (Copy) and ctrl V (Paste) and standard menu items. They allow you to copy or cut the currently selected item (or items) in a flowsheet window and paste them into any other flowsheet window on your desktop. This could be a hierarchical flowsheet view or the main flowsheet view in any running ACM application.

The Cut operation prompts you to ask if the selected items should be deleted. Click Cancel to abort the operation.

If the destination flowsheet contains items with the same name as the pasted items, a dialog is presented allowing conflicts to be resolved. This is the Flowsheet Import dialog available using the Flowsheet Import facility. It allows you to rename or delete the copied items.

Only flowsheet items are copied. Models on which the items are based and referenced objects such as component lists are not copied. The copy fails if the destination simulation does not have a model of the same name as the source simulation. (You can load just the models from an input file using the Import Types facility available on the File menu.) Also, if the model in the destination simulation has a different structure e.g. ports or variables with different names, warnings are generated and some of the copied data will be lost e.g. connections not made.

Once you paste the items you can move them as a group to the required location on the flowsheet.

Updating the Model used by a Block on a Flowsheet

You can change the model used by a block by dragging and dropping a model onto the block on your flowsheet. You are then asked to confirm the change of model. Once the model has been changed the connections and assignments are reapplied. If port or variable names in the new model do not

match those in the existing model, the connection or assignment is not reapplied and a warning is given.

Switching to Local Model

When you have flowsheet items that use model or stream types from libraries but you also have models or streams types with the same name locally under the Custom Modeling folder, you can switch the flowsheet items to use the local models. This can be done from a menu item labeled Switch to local model, available on the context menu for a model or stream type node in the Explorer. After confirmation the relevant flowsheet items are updated. If the new model has different ports or variables from the original, some connections or assignments will not be reapplied. Warnings are given when that happens.

If you copy a model from a library into the Custom Modeling folder, or use the Import Types facility, the system checks that flowsheet items using models or stream types with the same name exist. If they do, the system asks if you want to update the flowsheet items to use the newly created local types. Confirming this request causes the newly created model to be used. In the case of the Import Types facility it repeats this for each model or stream type imported.

Running Simulations

Running Simulations

Your Aspen Modeler product can run five different types of simulation:

- Initialization Specifying the initial conditions for a subsequent dynamic run. You may initialize state variables or variables related to state variables, time derivatives, or a combination of both.
- Steady-state Running a simulation where the time derivatives of a dynamic simulation are equal to zero.
- Dynamic Running a simulation where the variables change over time.
- Estimation May be one of:
- *Parameter Estimation* Fitting model parameters to experimental data.
- *Data Reconciliation* Comparing the steady-state plant performance with that predicted by a model.
- *Optimization* Optimizing steady-state or dynamic solutions using an objective function and supplied constraints.

Run an Initialization Simulation

To run an initialization simulation:

1 Define the initial conditions for your simulation.

2 In the mode box on the Run Control toolbar, click Initialization.

3 On the Run Control toolbar, click the Run button to run the current simulation.

You can perform the following actions on an initialization run:

- Rewind the simulation to time = 0.
- Reset the simulation, which removes the current solution point and returns all variables to their default values.

Reset a Simulation

To reset a simulation:

1 From the Run menu, click Reset.

The Reset dialog box appears, warning you that this will reset all user-assigned variable values to their default values. The spec and bounds properties of the variables as well as all parameter values are unaffected.

The values and derivatives which are affected are dependent on the run mode, as shown in this table:

Run Mode	Specs for which Value is reset to default	Specs for which Derivative of state variables is reset to default (of 0)
Steady State	Free, Initial, RateInitial	Free, Initial, Fixed, RateInitial
Dynamic	Free, RateInitial	Free, Initial, Fixed, RateInitial
Optimization	Free, RateInitial	Free, Initial, Fixed, RateInitial
Estimation	Free, RateInitial	Free, Initial, Fixed, RateInitial
Initialization	Free, RateInitial	Free, Initial, Fixed, RateInitial

Note: Unlike the Restart command, any change to the status of tasks will also be unaffected.

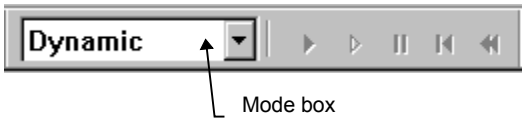
2 Click OK to reset the simulation.

3 On the Run Control toolbar, click the Run button to start the solution of the next dynamic run from its initial conditions at time zero.

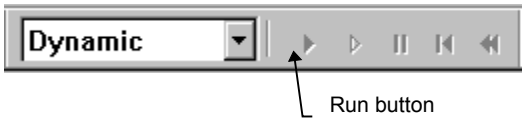
Run a Steady-State Simulation

To run a steady-state simulation:

1 In the mode box on the Run Control toolbar, click Steady State:



2 On the Run Control toolbar, click the Run button to run the current simulation:



You can perform the following actions on a steady-state run:

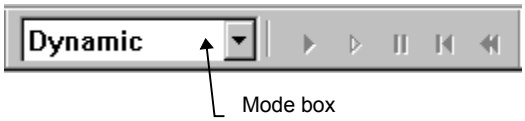
- Rewind the simulation to time = 0.
- Reset the simulation, which removes the current solution point and returns all variables to their default values.

Run a Dynamic Simulation

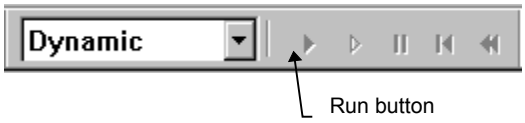
To run a dynamic simulation:

1 Define the initial conditions for your simulation.

2 In the mode box on the Run Control toolbar, click Dynamic:



3 On the Run Control toolbar, click the Run button to run the current simulation:



You can perform any of the following actions on a dynamic run:

- Step the simulation one time interval at a time.
- Pause the simulation.
- Restart the simulation, so that the next run restarts from its initial conditions at time = 0.
- Rewind the simulation back to a previous time point if snapshots are available. If there are no snapshots, Rewind will take the simulation back to time= 0.
- Reset the simulation, which removes the current solution point and returns all variables to their default values.
- Change the time settings for the simulation.

Step a Simulation

To step through a dynamic simulation one time interval at a time:

- On the Run Control toolbar, click the Step button:

Pause a Dynamic Simulation

You can pause a dynamic simulation in two ways:

- By clicking the Pause button.
- At a previously specified time.

Pause a Simulation

To pause a running simulation:

- On the Run Control toolbar, click the Pause button:

Pause a Dynamic Simulation at a Specific Time

To pause a dynamic simulation at a specific time or after a specified number of time intervals:

1 From the Run menu, click Run Options.

2 In the Run Options dialog box, you can do one of the following:

To	Do this
Pause at a specific time	Check the Pause At box and enter the time for suspending your dynamic run.
Pause after a specified number of time intervals	Check the Pause After box and enter the number of intervals.

3 To apply your changes, do one of the following:

To	Click
Keep the dialog box open	Apply
Close the dialog box	OK

Restart a Simulation

To restart a dynamic simulation:

1 From the Run menu, click Restart.

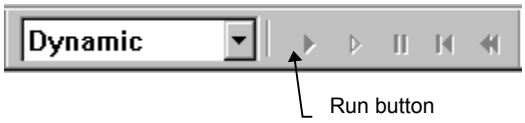
– or –

On the Run Control toolbar, click the Restart button:

The first snapshot (taking the most recent run first) labeled with a simulation time of 0 is loaded.

Note: If you want to load a specific snapshot, use Rewind and choose the required snapshot.

2 On the Run Control toolbar, click the Run button to start the solution of the next dynamic run from its initial conditions at time zero:



Change the Time Settings

For a dynamic run, you can change:

- The time interval at which data is available for plots and snapshots during a simulation, known as the communication interval.
- The units for the communication interval.
- The relationship between real time and simulation time.

Change the Communication Intervals for a Dynamic Run

To change the communications intervals for a dynamic run:

1 From the Run menu, click Run Options.

2 In the Run Options dialog box, enter the value of the communication intervals and the time units of measurement that you want to use for your simulation.

3 To apply your changes, click one of the following:

To	Click
Keep the dialog box open	Apply
Close the dialog box	OK

Specify the Real Time Synchronization for a Dynamic Run

To specify the real time synchronization for a dynamic run:

1 From the Run menu, click Run Options.

2 In the Run Options dialog box, type the value of the real time synchronization factor.

3 To apply your changes, click one of the following:

To	Click
Keep the dialog box open	Apply

Close the dialog box	OK

Note: If the synchronization factor is 0, synchronization is off (default setting). If you specify a synchronization factor, the time taken per communication interval is the communication time in the units indicated, multiplied by the synchronization factor (rounded to the nearest second). The larger the factor, the longer the run will take in real time.

Rewind a Simulation

To rewind a dynamic simulation:

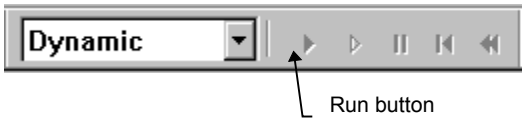
1. From the Run menu, click Rewind.

   – or –

   On the Run Control toolbar, click the Rewind button:

   The Rewind dialog box appears.

2. Click the required snapshot and then click Rewind to rewind to this snapshot.
   The dialog box closes automatically.
3. On the Run Control toolbar, click the Run button to run the simulation:



Run an Estimation Simulation

To run an estimation simulation:

1. Define your estimated variables, steady-state, and dynamic experiments using the Estimation dialog box (which is the recommended method), or Microsoft Visual Basic scripts.

Caution: Do not change the estimated or reconciled variables with a task or with the Simulation Access eXtensions (SAX) while you are performing an estimation run.

2. Change the integrator to either Gear or Variable Step Implicit Euler.
3. In the mode box on the Run Control toolbar, change the mode to Estimation.
4. On the Run Control toolbar, click the Run button to run the simulation.

You can perform the following action on an estimation run:

- Reset the simulation, which removes the current solution point and returns all variables to their default values.

For more general information on Estimation, see *Improving Your Simulations*.

Define an Estimation Simulation using the Estimation Dialog Box

To define an Estimation simulation using the Estimation dialog box:

1 From the Tools menu, click Estimation.

2 In the Estimation dialog box, you can:

Define your estimated variables.

Create and modify steady-state and dynamic experiments.

Reset Estimation experiments.

Obtain results of estimation simulations.

Note: For a steady state data reconciliation, some or all of the estimated variables will also be defined as measured variables in a steady state experiment. Only one active experiment can contain estimated variables.

Define Estimated Variables

To define estimated variables:

1 In the Estimation dialog box, click the Estimated Variables tab.

2 Type the name of the new variable in the box and then click Add.

– or –

Drag and drop variables from a table or from Variable Find.

Note: You can also remove Estimated variables and change their current value and lower and upper limits from this tab.

After a successful Estimation run, the standard error for each estimated variable is displayed here.

Define Steady-State Estimation Experiments

To define a new steady-state estimation simulation:

1 In the Estimation dialog box, click the Steady-State Experiments tab and click New.

2 To display details of the experiment, click the Edit button or double-click the experiment name.

3 You can now:

Add or remove Measured or Fixed variables

Change the weight, value and units for each experiment variable by selecting them on a table. Measured variables are shown as reconciled.

Change the experiment weight and activity.

After a successful Estimation run, the Predicted Value, Absolute Residual, % Residual, and Standard Residual for each steady-state Measured Variable is displayed on the table.

Note:

You can also:

- Copy existing experiments by selecting the experiments to copy and clicking the Copy button.
- Change the weight and activity for each experiment by double-clicking the weight and changing it or clicking the Active box to toggle the activity.
- Remove unwanted experiments by selecting them and clicking the Remove button, or clicking Remove All.
- Edit an experiment by double-clicking the name.
- Rename an experiment by clicking it when it is selected.

Define Dynamic Estimation Experiments

To define a new dynamic estimation simulation:

1 In the Estimation dialog box, click the Dynamic Experiments tab and click New.

2 To display details of the experiment, click the Edit button or double-click the experiment name.

3 You can now:

Add or remove Measured, Fixed, or Initial variables.

Change the weight, value and units for each experiment variable by selecting them on a table. Measured variables are shown as reconciled.

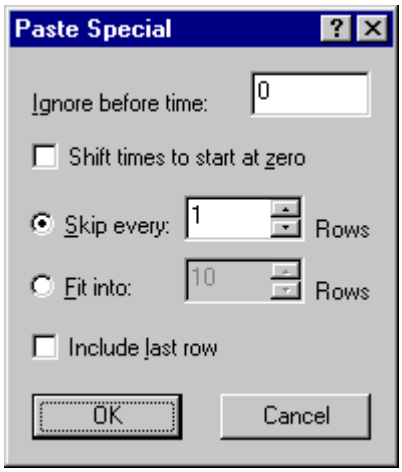
Change the experiment weight and activity.

Display the data on a scatter graph, including results, if available.

4 To edit Measured and Fixed variables, you can also double-click the variable name or select the name and click Edit.
A dialog box is displayed where you can enter values for the variable at different time points.

5 To add new time points, click Add, or double-click a blank column to create a new point.

Note: You can paste columns of values from other applications such as MS Excel, using standard cut-and-paste or drag-and-drop methods, or alternatively you may use Paste Special:



If you paste	Then
A single column	The values in the currently selected column will be overwritten, unless you paste to the Time column, in which case new time points are created after deleting the selected time points.
Multiple columns (weights not included)	New time points are created, assuming the first column is time, and the last column is the Measure. Intervening columns are ignored.
Multiple columns (weights included)	New time points are created, assuming the first column is time, the second from last the Measure value, and the last Weight. Intervening columns are ignored.

Tip: If you are pasting from Excel, you can use the CTRL key to select multiple non-adjacent columns, but ensure the Time column is on the left and the values and weights (optional) are on the right of the selection. You can also select Paste Special from the context menu available over the dynamic data grid. This brings up a dialog allowing you to select which rows of data are to be used and to adjust the experiment data times.

6 After a successful Estimation run, the Predicted Value, Absolute Residual, % Residual, and Standard Residual for each dynamic Measured Variable is displayed on the table.

Note:

You can also:

- Copy existing experiments by selecting the experiments to copy and clicking the Copy button.
- Change the weight and activity for each experiment by double-clicking the weight and changing it or clicking the Active box to toggle the activity.
- Remove unwanted experiments by selecting them and clicking the Remove button, or clicking Remove All.
- Edit an experiment by double-clicking the name.
- Rename an experiment by clicking it when it is selected.

Reset Estimation Experiments

To reset Estimation experiments:

1. From the Tools menu, click Estimation.
2. In the Estimation dialog box, click Reset.

Access Results of Estimation Simulations

To obtain results from Estimation simulations:

1. From the Tools menu, click Estimation.
2. In the Estimation dialog box, click the relevant tab:

Type of Results	Tab
Estimated variables, standard error	Estimated Variables tab
Steady-state Measured variables, predicted values and residuals	Measured Variables tab for each steady-state experiment
Dynamic Measured variables	Measured Variables tab for each dynamic experiment
Correlation and covariance matrices	Status tab, click Matrices button
Additional MLL results	Status tab, click MLL button

Statistical Results from Estimation Simulations

At the end of an estimation simulation, Aspen Custom Modeler provides statistical information for you to assess the quality of fit of your estimated variables to the measured data.

This topic gives definitions of the statistical measures used in Aspen Custom Modeler. It is in two parts:

- Statistical results available through the Maximum Log Likelihood dialog box.
- Other statistical measures.

You should look at both parts for a complete overview of the statistical measures in Aspen Custom Modeler.

Additional Maximum Log Likelihood Results

The Maximum Log Likelihood dialog box displays the following results:

- Maximized Log Likelihood Function.
- Weighted Residual Sum of Squares.
- Weighted Residual Sum.
- Standard Error of Estimate.
- Percentage Variation Explained.
- Heteroscedasticity Parameter.

For more information on Maximum Log Likelihood, see Improving Your Simulations.

Definitions of Other Statistical Measures

Aspen Custom Modeler applies the following definitions to these statistical measures:

- Covariance
- Correlation
- Standard error
- 95% confidence interval
- RMS error
- F-value
- Standardized Residuals

Covariance

Covariance is a measure of how sensitive the values of the converged estimated variables are to small perturbations in the measurements. It answers the question: How good is our solution to the estimated variables?

For a strict mathematical definition of covariance and how to approximate it, see *Nonlinear Parameter Estimation* by Yonathon Bard (1974, Academic Press). To approximate covariance, Aspen Custom Modeler uses the formula:

$$Covariance = \frac{2 \times Sum\ of\ Squares}{max(1, m - p)} \times (J^T J)^{-1}$$

Where:

m	=	Total number of measurements.
p	=	Number of estimated variables.
J	=	m by p Jacobian matrix of partial derivatives of

the measurements with respect to each estimated variable.

The sum of squares is the weighted absolute squared error between the observed and predicted values.

This covariance approximation is valid provided you have a good fit; if the sum of squares is large, the covariances may be inaccurate. The approximation can also be suspect if the solution of the estimated variables is on the bounds.

Correlation

Correlation is directly computed from the covariance matrix. It gives a standardized measure of the relationship between the estimated variables. The correlation value lies between -1 and +1. The larger the absolute value, the stronger the relationship or dependence between the estimated variables.

Standard Error

The standard error (also known as standard deviation) is the square root of the variance. It is a measure of the quality of the converged or fitted value of the estimated variable. The smaller the standard error, the more confident you can be in the accuracy of the estimated value.

95% Confidence Interval

The 95% confidence interval gives an upper and lower bound for the converged value of each estimated variable: there is a 95% probability that the true value of the estimated variable lies within these bounds.

RMS Error

RMS (root mean square) error is the square root of the weighted absolute squared error between the observed and predicted values of the measurements, divided by the number of measurements. Use the RMS error to compare how good the fit of the measured variables is for different sets of measured data.

F-Value

You can use the F-value to assess the relative quality of different models, that is, how well each model fits the measured data you have provided for an estimation simulation. This is done by comparing the computed F-values from the estimation simulation, using the same set of data but using different models.

The F-value is:

$$\frac{\left(sum\ of\ \ squares\ of\ \ weighted\ predicted\ values\right)/\left(No.\ of\ \ estimated\ \ \mathrm{var}iables\right)}{\left(sum\ of\ \ squares\ of\ \ weighted\ errors\right)/\left(No.\ of\ \ measurements - No.\ of\ \ estimated\ \ \mathrm{var}iables\right)}$$

The larger the F-value, the better the model fits the data.

Standardized Residuals

The Standardized Residual of a measured variable is the residual (difference between the Observed and Predicted values) divided by the standard error of the measured variable. It is a scaling of the Absolute Residual which enables comparison of residuals for different variables in a unit-less fashion.

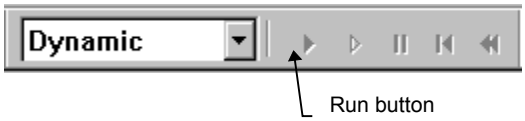
Run an Optimization Simulation

You can optimize either a steady-state or a dynamic simulation. The procedure for running an optimization simulation varies depending on the type of simulation you want to perform.

Run a Steady-State Optimization Simulation

To run a steady-state optimization simulation:

1. From the Tools menu, click Optimization and then on the Setup tab of the Optimization dialog box. Ensure Perform Steady-State Optimization is selected.
2. Click the Options button and on the Optimizer tab make any necessary changes to the options for your simulation, then click OK.
3. On the Decision Variables tab, add the decision variables for your optimization simulation.
4. On the Objective Variables tab, specify the variables that you want to optimize and select minimize or maximize from the drop down menu, and click Close.
5. Define any constraints on the Steady State Constraints tab and click Close.
6. In the mode box on the Run Control toolbar, change the mode to Optimization.
7. On the Run Control toolbar, click the Run button to run the simulation:



Run a Dynamic Optimization Simulation

A dynamic optimization simulation maximizes or minimizes an objective by finding optimal values of fixed (decision or control) or initial variables that satisfy your dynamic model over a given time range and optionally any additional constraints. This is achieved by performing a series of "inner" dynamic simulations or runs and using an "outer" nonlinear programming Successive Quadratic Program (SQP) solver.

The partial derivatives (sensitivities) of the objective with respect to the decision, control, and initial variables are computed during solution of the dynamic simulations. These derivatives are used by the optimizer to compute the correct search direction towards the optimum.

Dynamic Optimization

To perform a dynamic optimization, you must specify:

- At least one objective variable.
- Whether the final time is Fixed or Free.
- At least one variable that is either a decision, initial, or control variable.

You can also optionally specify:

- Additional equality and inequality final time constraints as language in the flowsheet constraints section.
- Final time constraints on variables directly in the Optimization Tool dialog.
- Path constraints on variables.
- Upper and lower bounds of the control variables on each element.
- Upper and lower bounds on the element sizes.
- A maximum move value for control variables.

For an example of dynamic optimization, see the Dynamic Optimization Example.

Tips:

- You are not forced to specify any additional constraints.
- If you do not specify an objective variable, a normal dynamic simulation is performed.
- You can use only the Gear or Variable Step Implicit Euler (VSIE) integrators for dynamic optimization. Note that the VSIE integrator is in general less accurate than the Gear integrator. Hence, you may need to tighten the VSIE tolerances and/or increase the optimization tolerance when using it for dynamic optimization.
- You must ensure that the accuracy used for the dynamic runs is smaller than the optimization tolerance.
- Dynamic optimization is suitable for use with small to medium-sized flowsheets. This is because a dynamic optimization simulation may need to perform many dynamic runs to find the optimum solution. Each dynamic run within the optimization will be slower than a single normal dynamic run due to the additional cost of computing the sensitivities.
- Before starting a dynamic optimization, perform manual dynamic runs to assess the behavior and to help pose feasible constraints.
- The cost of the inner dynamic runs increases with the number of decision, initial, and control variables. It also increases with the number of elements used to discretize the control variables. You are therefore advised to keep the number of elements small initially, and increase them in a series of optimization simulations, using the final converged values of the control variables as initial values for the subsequent simulation.
- To ensure faster performance, when you specify the communication time intervals for the run, enter a large value (for example, you could specify the final time), unless you are interested in the intermediate values of the variables for the run.
- Only a specific types of tasks and model conditional (IF) equations work with dynamic optimization. Tasks and IF equations inside models will work if:

- They purely depend on a fixed time value.
- The element sizes are not moving.
- The final time is fixed.

or if:

- The element sizes are moving but the IF or task always activates or switches at a fixed scaled location within the same element.
- In all other cases, IF equations and tasks introduce a discontinuity to the optimization, which is likely to cause a convergence failure. In some cases, for example, if the optimum is far enough away from the discontinuity, you may obtain convergence but this could be to a sub-optimal solution.
- If you are using an optimizer which requires gradients, sensitivities (partial derivatives of the objective and constraints with respect to the decision, initial, control variables and element sizes) are computed during the inner dynamic simulations. They are computed directly from the derivatives of your model equations. Therefore, it is essential that derivatives returned by your procedures are accurate, especially as normal dynamic simulations may not show errors. If you have procedures which return derivatives, is it recommended to use the ACM check procedure derivatives option to confirm that all your derivatives are correct before starting a dynamic optimization simulation.

Perform a Dynamic Optimization Run

To run a dynamic optimization:

1. In the mode box on the Run Control toolbar, change the run mode to Optimization.
2. From the Tools menu, click Optimization and then on the Setup tab of the Optimization dialog box, click Perform Dynamic Optimization.
3. On the other tabs, specify your Decision, Control, Initial, and Objective variables.

Caution: Do not change decision or control variables with a task or with the Simulation Access eXtensions (SAX) while you are performing a dynamic optimization run, otherwise you will get unpredictable behavior.

4. Specify the discretization method for your control variables.
5. Specify the final time.
6. Specify any additional final time or path constraints on free variables on the Dynamic Constraints tab.

Note: It is also possible to specify final time constraints as inequality and equality equations in the Constraints section of Flowsheet.

7. Click Close to accept your Dynamic Optimization specification.

8 On the Run Control toolbar, click the Run button to run the simulation.

Important note: When the dynamic optimization simulation has completed, the values of the variables in your flowsheet are the values at the *final time*. Before performing a second dynamic optimization run or after using Pause, you must rewind the simulation or use a result to restore the initial point. If you do not do this, the second dynamic optimization will use the final values of the first optimization as initial conditions.

Use the Optimal Control Discretization

After you have completed an optimization simulation, you may wish to use the optimal piecewise constant or linear discretizations of your control variables in a separate dynamic simulation. To do this:

- Open the Optimization dialog box and click the Setup tab.

  A task which emulates the existing discretizations of all your control variables is written. The task uses a series of ramps (for piecewise linear discretizations) or changes to values (for piecewise constant discretizations).

You can also write a script of the current optimization specification by pressing the Script button.

Initialize a Dynamic Simulation Globally

To initialize a dynamic simulation globally:

1 From the View menu, click Status Window, and then click the Initial Changes Tab.

2 You can now do one of the following:

To	Click
Initialize all time derivatives to zero	Steady State
Initialize all the state variables	States

3 To Close the Status window, click the Close button .

Use Properties Plus

When you want to use Properties Plus for Physical Properties you can either invoke the Properties Plus user interface directly (if you have it installed) or reference an intermediate file (extension .aprpdf or .appdf) In both cases double click on the Configure Properties node in the Explorer in the Component Lists folder. This will display the Physical Properties Configuration dialog. Click on the Use Aspen Properties radio button.

If you want to use the Properties Plus user interface either use the Edit Using Aspen Properties button to launch the user interface or the Import Aspen

Properties File to load an existing Aspen Properties input file (extension .aprbkp or .aprinp) You can then use Edit Using Aspen Properties button to edit the properties once you have done the import if you require. When you next save the simulation file the Properties Plus input will be included in the file. Then whenever you load the simulation input file ACM will check if an intermediate file needs to be generated.

If you want to use the Aspen Properties intermediate file directly click on the Use Properties definition file button and either type in or browse for the required file.

To generate the intermediate file you can use either Aspen Plus or Properties Plus. In either product you can specify, property methods (for example PENG-ROB and UNIFAC), components and property data to be used in your simulations. Refer to the Aspen Plus or Properties plus help for more information.

To use an existing input file using Aspen Plus:

1 Start Aspen Plus.

2 Select Open an Existing Simulation / More Files...

3 Locate the folder where the required input file (.bkp or .inp) is located.

4 Open the file.

5 Run the simulation and then save it in the Aspen Plus document format (*.apw). A file runid.appdf is created, with the *.apw file.

6 Exit Aspen Plus.

To use an existing input file using Properties Plus:

1 Start Properties Plus.

2 Select Open an Existing Simulation / More Files...

3 Locate the folder where the required input file (.aprbkp or .aprinp) is located.

4 Open the file.

5 Run the simulation and then save it in the Properties Plus document format (*.aprop). A file runid.aprpdf is created, with the *.aprop file.

6 Exit Properties Plus.

Select New instead of Open when you want to create an input file from scratch.

Change Data in a Table for a Block or a Stream

To change data in a table for a block or a stream:

1 In the Flowsheet window, click a block or stream, and then click the right mouse button and point to Forms.

2 Click the table whose data you want to change.

Tip: The AllVariables table shows all the variables in the block.

3 Use the table as shown:

To change	Do this
A variable's value	Click the appropriate cell in the Value column and type the new value.
A variable's spec	Click the appropriate cell in the Spec column and type the new spec value.

4 To close the table, click the Close button .

Units of Measurement

Click one of the following to display information on:

- Changing units of measurement for displaying data.
- Defining physical quantities, units of measurement, and conversions.
- Units of measurement for your models.
- Time units of measurements in simulation models and the graphical user interface.

Change Units of Measurement for Displaying Data

The following built-in units of measurement (UOM) sets are provided:

Set	Description
ENG	English engineering units
MET	Metric engineering units
SI	SI units
METCBAR	A variation on the MET unit set that uses Celsius for temperature and bar for pressure
SI-CBAR	A variation on the SI unit set that uses Celsius for temperature and bar for pressure
Metric	Metric units
US	US units

Note: ENG, MET, SI, METCBAR and SI-CBAR are all identical to the Aspen Plus units of measurement sets of the same name.

You can change the units of measurement either globally or for individual variables:

To change the UOM	Do this

Globally	From the Tools menu, point to Units of Measurement and then click the required set.
For a single variable	On a table: 1. Display the Units attribute for the table. 2. Click the Units cell for a variable to select different units of measurement.

Units of Measurement for Your Models

The built-in units of measurement conversions require models to be written in the Metric units of measurement set.

Also, the built-in physical property procedures and interface require models to be written in Metric units. You are therefore strongly advised to write your models in this units of measurement set. You can write your models in different units of measurement sets, but you will not be able to use Properties Plus or the built-in units of measurement conversions.

Metric Units of Measurement Set

PhysicalQuantity	base units
AbsoluteTemperature	"K"
Angle	"rad"
AngularVelocity	"rad/s"
AngularVelocityRPM	"rpm"
Area	"m2"
Diffusivity	"cm2/s"
ElectricPower	"kW"
Energy	"GJ"
Head	"kJ/kg"
HeatFlow	"GJ/hr"
HeatFlux	"kW/m2"
HeatFluxLinear	"W/m"
HeatOfReaction	"GJ/m3/hr"
HeatTransferCoeff	"kW/m2/K"
HookesConstant	"N/m"
HTCoeffArea	"kW/K"
Length	"m"
LengthShort	"mm"
Mass	"kg"
MassDensity	"kg/m3"
MassEnthalpy	"MJ/kg"
MassFlow	"kg/hr"

MassFlux	"kg/s/m2"
MassFraction	"kg/kg"
MassHeatCapacity	"kJ/kg/K"
MassReactionRate	"kg/hr/m3"
MassSecondMoment	"kmol/kg"
MassSpecificFlow	"kg/hr/m2"
MassSpecificVolume	"m3/kg"
MassThirdMoment	"Mmol/kg"
MolarDensity	"kmol/m3"
MolarEnthalpy	"GJ/kmol"
MolarEntropy	"kJ/kmol/K"
MolarFlux	"kmol/s/m2"
MolarGibbs	"kJ/kmol"
MolarHeatCapacity	"kJ/kmol/K"
MolarReactionRate	"kmol/hr/m3"
MolarVolume	"m3/kmol"
MolarWeight	"kg/kmol"
MoleFlow	"kmol/hr"
MoleFraction	"kmol/kmol"
Moles	"kmol"
MolesPerMass	"mol/kg"
MomentOfInertia	"kg m2"
Momentum	"tonne m/s"
Percent	"%"
Power	"kW"
Pressure	"bar"
PressureDifference	"bar"
PressureDiffSmall	"N/m2"
PressureDropCoeff	"m1.5 kg0.5/hr/bar0.5"
PressureDropCoeff2	"bar hr2/kg/m3"
SurfaceTension	"N/m"
Temperature	"C"
TemperatureDifference	"C"
ThermalConductivity	"W/m/K"
Time	"hr"
TimeMin	"min"
TimeSec	"s"
Velocity	"m/s"
Viscosity	"cP"

VolFlowPerRPM	"m3/hr/rpm"
Volume	"m3"
VolumetricFlow	"m3/hr"
VolumetricFlux	"m3/s/m2"

Time Units of Measurement

There are two settings for time units of measurement, both available from the Run Options dialog box. These are:

- Simulation time units.
- Display time units.

Simulation Time Units

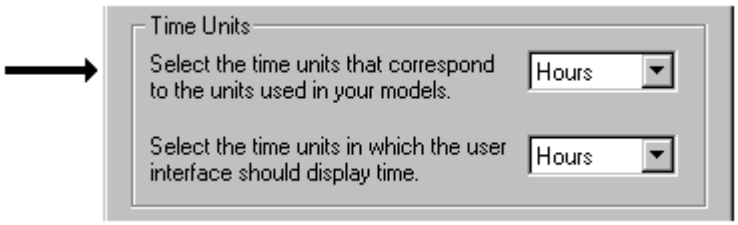
Simulation time units are the units you use by convention in your models. The default is hours, which are the time units in the supplied model libraries. If your models are not based on hours, change this setting to the required units.

The available simulation time units are hours, minutes and seconds.

To change the simulation time units:

1 From the Run menu, click Run Options.

2 In the Run Options dialog box, go to the Time Units field and select new simulation time units from the upper of the two boxes.



- Time Units in Control Models.

Display Time Units

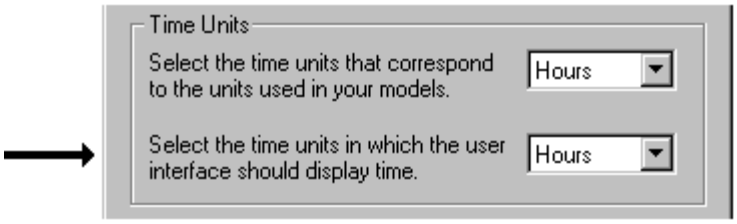
Display time units are the units Aspen Custom Modeler uses to display time in the user interface, for example on history tables and plots. If necessary, Aspen Custom Modeler automatically converts from the simulation time units to the display time units.

The available display time units are hours, minutes and seconds.

To change the display time units:

1 From the Run menu, click Run Options.

2 In the Run Options dialog box, go to the Time Units field and select new display time units from the lower of the two boxes.



Reset Specifications from the Status Window

To reset specifications changed from default settings defined in models:

1 From the View menu, click Status Window.

2 Do one of the following:

To	Do This
Reset all fixed and free settings	Click Fixed Changes Tab and click Reset.
Reset all Initial and unInitial settings	Click Initial Changes Tab and click Reset.

Activate a Task

A task is a set of instructions that you can create in your AspenTech product. A task defines a sequence of actions that take place during a dynamic simulation, such as changing feed flow rate.

After you have created a task, you must activate it, so that it is used in the current simulation.

Caution: Do not change estimated or reconciled variables with a task while you are performing a dynamic estimation run; and do not change decision or control variables with a task when you are performing a dynamic optimization run.

You can activate tasks for:

- A flowsheet, where the task applies to the entire flowsheet.
- A block or stream, where the task applies to the selected item.

Activate a Flowsheet Task

To activate a flowsheet task:

1 In the All Items pane of the Simulation Explorer, click Flowsheet.

2 In the Contents pane, click the appropriate task with the right mouse button, and then click Activate:

A check mark appears in front of the Activate command and the task icon

changes to display a blue check mark, which indicates that the task will be used in the simulation:

Caution: Do not change estimated or reconciled variables with a task while you are performing a dynamic estimation run; and do not change decision or control variables with a task when you are performing a dynamic optimization run.

Activate a Block or Stream Task

To activate a block task:

1. In the Flowsheet window, click a block, then click the right mouse button, and then click Explore.
   The Exploring dialog box for the block appears.
2. In the Contents pane, click the appropriate task with the right mouse button, and then click Activate:

   A check mark appears in front of the Activate command and the task icon changes to display a blue check mark, which indicates that the task will be used in the simulation:

Caution: Do not change estimated or reconciled variables with a task while you are performing a dynamic estimation run; and do not change decision or control variables with a task when you are performing a dynamic optimization run.

Run a Script

A script is a set of instructions, written in Microsoft Visual Basic, to automate the setup of a simulation.

You can run scripts for:

- A flowsheet, where the script applies to the entire flowsheet A library script also applies to the flowsheet.
- A block or stream, where the script applies to the selected item

If errors occur when you run a script, and if a suitable debugger has been installed on your machine, you can debug the application.

The Modeler.acml library delivered with ACM contains some standard scripts you can call.

Run a Flowsheet Script

To run an existing flowsheet script:

1. In the All Items pane of the Simulation Explorer, click Flowsheet.
2. In the Contents pane, click the appropriate script icon with the right mouse button.
3. Click Invoke.
   The script is run. After completion, the Scripting dialog box appears. Click OK to close this dialog box.

Run a Block or Stream Script

To run a script for a block or stream:

1. In the Flowsheet window, click a block or stream, then click the right mouse button, and then click Explore.
   The Exploring dialog box for the block or stream appears.
2. In the Contents pane, click the appropriate script with the right mouse button, and then click Invoke.
   The script is run. After completion, the Scripting dialog box appears.
3. Click OK to close this dialog box.

Run a Library Script

To run an existing script that is stored in a library:

1. In the All Items pane of the Simulation Explorer, click Custom Modeling or any library node, and then click Scripts.
2. In the Contents pane, click the appropriate script icon with the right mouse button and then click Invoke Script.

   The script is run. After completion, the Scripting dialog box appears. Click OK to close this dialog box.

Delivered Library Scripts

The Modeler.acml library delivered with ACM contains some standard scripts you can call. This can be done either by finding them in the Explorer in the Scripts folder of the library or by invoking them from a script of your own using the InvokeLibraryScript method.

ListLargeDerivatives

This script will list out in the simulation messages window the variables with large derivative values in the current simulation. The tolerance used on the derivative value is by default 0.01 of the value of the variable but this can be specified as an argument to the script.

```
e.g. InvokeLibraryScript
"modeler.acml","listlargederivatives",4
```

will use a tolerance of 0.0001.

Handle Scripting Errors

When you run a script, errors may occur, for example, if you try to access a variable that does not exist, the system raises an error. If a suitable debugger has been installed on your machine, a dialog box will prompt you to debug the application.

If you click	Then
No	An error is reported in the Simulation Messages window.
Yes	The Microsoft Development Environment is displayed. At the prompt asking if you want to open a project, click No. The line containing the error in your script will be highlighted. You can use the debugger facilities to analyze the error.

Remember: When you have finished your debug session, on the Debug menu, select Detach All Processes. If you do not do this, closing the debugger will shut down your Aspen Modeler application.

Check the Status of a Simulation

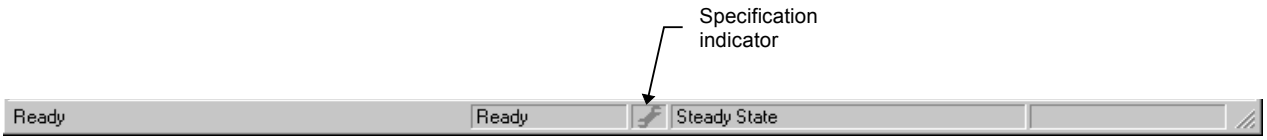
To check the status of a simulation:

1. Make sure the Flowsheet window is active by clicking in it.
2. From the View menu, click Status Window.

   The Status Window appears, giving details of the current run status.

   – or –

   Double-click the specification indicator on the Status bar:



Check for Structural Singularities in a Simulation

To check for structural singularities in a simulation:

1. From the View menu, click Status Window.
2. On the General Tab of the Status Window, click Check.
3. In the Legality area, the symbol indicates whether structural singularities exist:

This symbol	Means

	No structural singularities.
	Structural singularities present.

Work with Specification Analysis Information

If your simulation will not solve, the Status bar displays a symbol other than the green square, or a warning message appears in the Simulation Messages window.

You can use Specification Analysis to recommend changes to the specification status (Fixed, Free, or Initial) of as many variables as are required to make the simulation solve.

You can restrict the list of variables that are recommended for change by confirming the current specification status of any variables that must remain unchanged.

Click the following links to display information on Specification Analysis:

- Access Specification Analysis.
- Use recommendations from Specification Analysis.
- Confirm the current status of variables that must remain unchanged.
- List variables that cannot be used for Specification Analysis for all simulations.
- Using Specification Analysis for initialization and dynamic runs.

Access Specification Analysis

To obtain recommendations on which variables to change to make the simulation solvable:

1 Do one of the following:

- From the View menu, click Specification Analysis.

  – or –

- From the Status window, click Analyze.

2 In the Specification Analysis dialog box, click Analyze Now.
The Specification Analysis dialog box shows details of each variable that is recommended to be changed: the variable name, its current specification status, its recommended specification status, its value, and the DOF (Degrees of Freedom) of the parent block or stream in which the variable occurs.

Use Specification Analysis Recommendations

When you click Analyze Now, the Recommended Changes list displays the names of variables whose specification status can be changed as indicated to make the simulation solvable.

You can do any of the following to variables in the Recommended Changes list:

To	Do this
Accept the recommended changes	Click the variable name(s) and then click Accept.
Change the value of a variable	1. Click the variable name, and then click Properties. 2. In the Properties dialog box, enter a new value for the variable and then click Accept.
Reject the suggested changes	Transfer the variables to the Do Not Change Spec list. To do this: 1. In the Recommended Changes list, click the name of the variable(s) whose specification status you want to remain unchanged. 2. Click the Transfer button. 3. Click Analyze Now again.

Note: To obtain further information, as well as details of any equations which may be causing problems, click Details.

Confirm Variables' Current Status

In the Do Not Change Spec list, include both fixed and free variables whose current specification status you know to be correct, and which therefore should not be included in the Recommended Changes list.

You can do the following to the Do Not Change Spec list:

To	Do this
Add variables	In the Add text box at the bottom of the Do Not Change Spec list, type a variable name and then click Add. **Note:** You can also use wildcards – or – Drag and drop variable names from Variable Find and from tables directly into the list.
Remove variables	Click the variable(s) to be removed and then click Remove – or – Click Remove All.
Rename variables	Click the variable name, click Edit, and type a new name.

Convert Variables into Expressions

To convert a variable on the Do Not Change Spec list to an expression:

1 Click the chosen variable with the right mouse button.

2 Do one of the following:

To convert the variable to an expression that gives	Click
All variables in the owning block or stream	Block/Stream
All elements in the owning array (if any)	Whole Array
All variables of the same variable type	Variable Type

Exclude Variables from Specification Analysis for All Simulations

You can exclude chosen variables from Specification Analysis for *all* simulations. To do this:

1. Click Settings.
2. On the Specification Analysis tab, add the variables that you want to exclude from Specification Analysis for all simulations.
   These values are stored for your Windows user profile and are used whenever you run your AspenTech product.

Use Specification Analysis for Initialization and Dynamic Run Modes

For the Initialization and Dynamic run modes, the initialization specification needs to be given after the steady-state specification, that is, after you have chosen which variables should have their specification status set to Fixed. The initialization specification requires that you select which variables should have their specification status set to Initial or RateInitial. To do this:

1. When the steady-state specification is complete, click Analyze Now.
   The Initial Specification dialog box is displayed.
2. You can choose one of the following options:

Option	Result
Set the specification status of all state variables to Initial	The specification process is completed.
Set the specification status of all state variables to RateInitial	The specification process is completed.
Use Specification Analysis	Enables you to use Specification Analysis to make recommendations to complete the initialization specification. Specification Analysis will recommend which variables should have their specification status changed to Initial or RateInitial.

Generate a DLL File for Procedures

To use external routines, you must:

1 Make sure that you obey the external procedure calling conventions for the language you are using.

2 Generate the DLL by:

- Using existing external code.
- Creating new external code.

Important

- If your routines make calls to other routines in a library, or in other Fortran files, see Generate a DLL for Procedures that call other External Routines.
- If you need to change the procedure code contained in a DLL, you can unload the DLL. The unloaded DLL will then be re-loaded when the next run that requires it is performed.

Use Existing External Code

To use existing external code:

1 Within the procedure definitions for the implementation property, include the file name for the code. Then:

To generate code for	Do this
All procedures	From the Tools menu, click Generate Procedure Code.
A single procedure	In the All Items pane of the Simulation Explorer, click Procedures. In the Contents pane, click the appropriate block with the right mouse button and then click Generate Code.

The Generate Procedure Code dialog box appears with Wrappers in the Generate Files area selected by default.

2 In the Makefile area, select Generate and Execute.

Note: If you need to modify the generated MakeUserCode makefile, do not select Execute.

3 If necessary, change the path to your working directory and click OK. The makefile, MakeUserCode (and also wrapper routine if you are using Fortran), and DLL are generated.

4 If you did not select Execute in step 2, you can do this after you have edited the makefile by entering the following command from the MS-DOS prompt :

```
-f MakeUserCode
```

For future runs, you need to supply only the DLL file with the input.

Fortran Users The argument list for your external routines must contain the parameter IFlag. If you are using your routines for properties calculations, then your argument list must also contain the parameter IType.

If IFlag or IType is missing, add them to avoid the error "unsatisfied external symbols" when you generate the dynamic link library.

Create New External Code

To create new external code:

1 Decide whether to generate code for all procedures, or for a single procedure:

To generate code for	Do this
All procedures	From the Tools menu, click Generate Procedure Code.
A single procedure	In the All Items pane of the Simulation Explorer, click Procedures. In the Contents pane, click the appropriate block with the right mouse button and then click Generate Code.

The Generate Procedure Code dialog box appears with Wrappers in the Generate Files area selected by default.

2 Select Templates, to create a template into which you can insert your code.
Write or insert your code in the template.

3 When you have finished adding your code to the template, select Generate and Execute in the Makefile area.

Note: If you need to modify the generated MakeUserCode makefile, do not select Execute.

4 If necessary, change the path to your working directory, and then click OK.
The template and the makefile, MakeUserCode (and also the wrapper routine, if you are using Fortran), and DLL are generated.

5 If you did not select Execute in step 3, you can do this after you have edited the makefile by entering the following command from the MS-DOS prompt :

```
nmake -f MakeUserCode
```

For future runs, you need to supply only the DLL file with the input.

Generate a DLL for Procedures that call other External Routines

To make calls from one routine to routines in other Fortran files or in a library that is not used directly by your AspenTech product:

1 Generate the wrapper and MakeUserCode in the normal way.

2 Before generating the DLL, open the MakeUserCode file.

3 Find the section headed **Problem specific part**.

4 To add an object library to the link, add a definition for USERLIBS naming the library, and a definition for USERLIBPATHS, specifying the path to the library, prefixed by $(LIB_PATH).
For example:

```
USERLIBS = My_Objects$(OLB)
USERLIBPATHS = $(LIB_PATH)"C:\Development\Lib"
```

5 To add extra objects to the dll, add a definition for USERTARGETS naming the objects to be added.
For example:

```
USERTARGETS = My_First$(OBJ) My_Second$(OBJ)
```

If these objects need to be compiled, add rules to the end of the file to do that.
For example:

```
   My_First$(OBJ): My_First.f
        $(ECHO) Compiling My_First.f
        $(FORTRAN) $(FDEBUG) $(F_OPTC) My_First.f
#
My_Second$(OBJ): My_Second.f
        $(ECHO) Compiling My_Second.f
        $(FORTRAN) $(FDEBUG) $(F_OPTC) My_Second.f
```

Example of Edited MakeUserCode File

The following example assumes:

- The user has procedure definitions for Median, Average and Range in Custom Modeling and has used Generate Procedure Code to create this makefile
- The user's code for Average and Range are supplied in a single source file called UserCode.f
- Two additional subroutines are called by the user's code, called My_First and My_Second, supplied by the user as FORTRAN source files with similar names. These need compiling and linking into the DLL.
- There are calls in the user's code to subroutines in gpp.dll and a user-supplied library stats.lib

```
#>>> This marks additional comments added to help you
understand the example


################## Problem specific part
##############################
#
```

```
#>>> The dll to be created: specified in the LIBRARY: part
of the
#>>> procedure definition
Libraries    = Example$(DLL)
#
#>>> This object list is modified from that generated by
the
#>>> procedure code export
#>>> The sources Range and Average are in UserCode.f,
#>>> so Range$(OBJ) and Average$(OBJ) have been replaced by
#>>> UserCode$(OBJ).
#>>> The two wrapper functions remain so that ACM can
#>>> call these subroutines.
Library1_Objects = MedianI$(OBJ) Median$(OBJ) \
        AverageI$(OBJ) RangeI$(OBJ) UserCode$(OBJ)
#
#>>>---- start of definitions added by the user
#
#>>> List of objects which need to be compiled separately,
#>>> and added to the dll that are NOT called directly by
ACM.
#>>> This has been added by the user.
USERTARGETS = My_First$(OBJ) My_Second$(OBJ)
#
#
#>>> Location(s) of libraries specified in USERLIBS, added
by the
#>>> user.
#>>> The first is the usual location of gpp.dll and gpp.lib
#>>> The second is the location of the 'stats.lib'
#>>> Note the use of double quotes (redundant in the second
case)
USERLIBPATHS = $(LIB_PATH)"C:\Program
Files\AspenTech\AMSystem 12.1\bin" \
               $(LIB_PATH)"C:\Development\TestLibrary"
#
#
```

```
#>>> Additional object libraries that contain subroutines
called by
#>>> the code in the dll
#>>> gpp.lib is supplied by AspenTech and permits linking
into
#>>> gpp.dll
#>>> stats.lib is an ordinary object library. Added by the
user.
USERLIBS = gpp.lib stats.lib
#
#>>>--- end of definitions added by the user
#
build: $(Libraries)
#
Example$(DLL): $(Library1_Objects) $(USERTARGETS)
      $(ECHO) Building Dynamic Link Library Example$(DLL)
      $(C_LINK) $(LINKDEBUG) $(C_OPTL) $(DLLOPT)
$(Library1_Objects)\
      "$(ATUTCFORPATH)libatdll$(OLB)" \
      $(ALT_EXECUTABLE)"Example$(DLL)" $(USERLIBPATHS)
$(USERLIBS) \
       $(SYSLIBS)
      $(ECHO) Dynamic Link Library built
#
MedianI$(OBJ): MedianI.f
      $(ECHO) Compiling MedianI.f
      $(FORTRAN) $(FDEBUG) $(F_OPTC) MedianI.f
#
Median$(OBJ): Median.f
      $(ECHO) Compiling Median.f
      $(FORTRAN) $(FDEBUG) $(F_OPTC) Median.f
#
AverageI$(OBJ): AverageI.f
      $(ECHO) Compiling AverageI.f
      $(FORTRAN) $(FDEBUG) $(F_OPTC) AverageI.f
#
RangeI$(OBJ): RangeI.f
```

```
	$(ECHO) Compiling RangeI.f
	$(FORTRAN) $(FDEBUG) $(F_OPTC) RangeI.f
#
#>>>------ Start of rules changed or added by the user
#
#>>> User has replaced rule for Average$(OBJ) here with one
for
#>>> UserCode$(OBJ), and removed the rule for Range$(OBJ)
UserCodeI$(OBJ): UserCodeI.f
	$(ECHO) Compiling UserCodeI.f
	$(FORTRAN) $(FDEBUG) $(F_OPTC) UserCodeI.f
#
#>>> User has added extra rules to compile the fortran for
the two
#>>> routines called by his procedures that are not in
either
#>>> stats.lib or gpp.lib
My_First$(OBJ): My_First.f
	$(ECHO) Compiling My_First.f
	$(FORTRAN) $(FDEBUG) $(F_OPTC) My_First.f
#
My_Second$(OBJ): My_Second.f
	$(ECHO) Compiling My_Second.f
	$(FORTRAN) $(FDEBUG) $(F_OPTC) My_Second.f
#
```

Unload DLLs from Simulations

To enable easy debugging of DLLs containing your own code, you can unload selected DLLs from a simulation, and automatically reload them the next time you run the simulation.

To unload DLLs from a simulation:

1. From the Run menu, click Unload User DLLs.
2. In the Unload User DLLs dialog box, select the DLL(s) you want to unload and click OK.
   The simulation restarts.

The unloaded DLLs are re-loaded the next time you run the simulation that requires them.

Work with Snapshots and Results

A snapshot is a saved set of values and specifications, for all variables in a simulation. You can use them to rewind simulations or to copy selected values to your simulation. Snapshots are taken automatically by the system, but can also be requested.

When you change the structure of a simulation, for example, by adding a controller, all snapshots become results. You can no longer use them for Rewind, but you can copy values from them.

Kept Snapshots and Results

A kept result is a result which has been marked to be kept in the simulation input file in ASCII form. This means that you do not need access to any result files to be able to access a kept result. You access a kept result in the same manner as other results. When a snapshot is marked as kept, it indicates that when the snapshot is archived it will be saved as a kept result.

Snapshots and results are displayed graphically within your Aspen Modeler product:

Icon	Meaning
	Kept snapshot
	Snapshot
	Result
	Kept result

Use Snapshots for Rewind and Restart

You can use any snapshot in a rewind operation. It can be used in any run mode and will cause the simulation to be rewound to the state stored in the snapshot. The value of all variables in the simulation will be set from the snapshot. Delay and procedure workspace information will be restored. If the run mode is dynamic the simulation time will be set to the time stored in the snapshot.

You can Restart any dynamic simulation. This will cause the system to rewind to the snapshot that was taken automatically at the start of the dynamic run. This is usually the snapshot labeled Dynamic Initialization. The simulation time will be set to 0.

You can Rewind and Restart from the Run toolbar or Run menu.

Access Snapshots and Results

To access snapshots and results use the Snapshots item on the Tools menu or the camera icon on the tools toolbar. This will display the snapshot management dialog. From this dialog box, you can access:

- Snapshots from the current simulation

- Results from the simulation input file, kept results.

The different types of snapshot and result are distinguished in the list using icons.

From this dialog box you can:

- Rewind using any snapshot. Once you close the dialog the simulation time will be set to the time in the snapshot. You cannot rewind using results.
- Copy values from any snapshot or result. This copies the values of all variables in the snapshot to variables that match by name in the current simulation. The simulation time is not changed.
- Rename or delete any snapshot or result.
- Flag as kept any snapshot or result which isn't already flagged as kept. When a snapshot is marked as kept it indicates that when the snapshot is archived it will be saved as a kept result.

From this dialog you can also access an advanced copy dialog box, which allows you to carry out more complex copy operations.

Save (Take) a Snapshot

Snapshots are taken automatically when any run is carried out. They are given a name which indicates why it was taken. In dynamic run mode several snapshots can be taken. One is taken at time 0 labeled Dynamic Initialization but you can then choose to take snapshots at regular timed intervals for the duration of the dynamic run. For some dynamic solvers which carry out reinitializations e.g. Gear snapshots will be taken at the reinitialization if requested on the Snapshot management dialog.

You can also take a snapshot at any time and give it a name of your choice.

Snapshots can be produced when:

- Snapshots are automatically saved in the current simulation.
- Snapshots are saved at reinitialization during a dynamic run.
- Automatically named snapshots, called Timed Snapshots, are saved during a dynamic run.
- You save custom snapshots.

Snapshots and results are, by default saved in files in the working directory on the server machine. You can request that any snapshot or result can be kept in the simulation input file. They can be accessed like any other snapshot or result.

Automatically Saved Snapshots

By default, for steady-state, initialization, estimation, and optimization runs, snapshots are saved automatically before (described as Initial Specification in the Snapshot Management dialog box) and after running the simulation.

For dynamic runs, snapshots are saved before and after the initialization phase (described as Dynamic Initialization in the Snapshot Management dialog box) of the run.

The snapshot management dialog Create tab allows you to control the automatic generation of snapshots. From this dialog you can:

- Switch off generation altogether.

- Control how often snapshots are generated during a dynamic run.
- Limit the number of automatically generated snapshots.

When a limit has been set, old automatically generated snapshots are deleted before new ones are created to ensure the maximum number specified is not exceeded.

Save Snapshots for a Dynamic Run at Reinitialization

For dynamic runs, you can save snapshots at reinitialization. To do this:

1 From the Tools menu, click Snapshots.

2 Click the Create tab on the Snapshot Management dialog box, and in the Scheduling area, select the Take a Snapshot on Simulation Reinitialization check box.

3 Click Close to exit the Snapshot Management dialog box.

Note: A simulation can have any number of snapshots associated with it. You can later delete snapshots if you wish.

Save Timed Snapshots for a Dynamic Run

For dynamic runs, you can save automatically named snapshots, called Timed Snapshots, at regular time intervals.

To save timed snapshots:

1 From the Tools menu, click Snapshots.

2 Click the Create tab on the Snapshot Management dialog box, and in the Scheduling area, select the Take Regular Snapshots check box. If necessary, change the default time interval and select whether the snapshots are to be taken for ever, or (to avoid filling up the disk), for a specified maximum time.

3 Click Close to exit the Snapshot Management dialog box.
Snapshots labeled with the current simulation time are saved during the run, at the time intervals you specified.

Note: A simulation can have any number of snapshots associated with it. You can later delete snapshots if you wish.

Save Snapshots Manually in All Run Modes

For all run modes, you can produce a named snapshot at any time during a run. To do this:

1 From the Tools menu, click Snapshots.

2 Click the Create tab on the Snapshot Management dialog box, and in the Manual area, enter a unique description of the snapshot in the text box.

3 Click Close to exit the Snapshot Management dialog box.
A snapshot with the unique description you supplied is saved during the run. For dynamic runs, the time at which you took the snapshot is also

saved. If you have the Always Mark Named Snapshot as Kept check box set on the snapshot page of the Settings dialog box, your named snapshot will be automatically marked as Kept.

Note: A simulation can have any number of snapshots associated with it. You can later delete snapshots if you wish.

Use a Snapshot

You can use a snapshot to:

- Copy variable values from the snapshot into a simulation, based on a search/pattern-matching of variables.
- Rewind a simulation in any run mode. This restores the state stored in the snapshot.
- Rewind a dynamic simulation (using Timed Snapshots only) to a previous time in the current results sequence. The current simulation time for the run is set to the time in the snapshot.
- Restart a dynamic run. The Restart command automatically rewinds to the first snapshot that can be found in the dynamic run.

Note: Automatically named snapshots taken after this time are deleted because their simulation times are outside the current simulation time range.

For detailed information and examples of using a snapshot to set the values for a simulation, see Copy Values from a Snapshot or Result.

Use a Result

You can use a result when copying values to the current simulation.

For detailed information about using a result to set the values for a simulation, see Copy Values from a Snapshot or Result.

Copy Values from a Snapshot or Result

You can copy values from any snapshot or result to the current simulation. This copies values when the name of the variable in the snapshot or result matches the name of a variable in the current simulation. This means that, for a snapshot, all variables in the simulation will receive a value, but for a result this may not be the case as the result may have a different structure. A report is given telling you which values were copied. No other information is copied.

You can also use the advanced copy facilities to refine the copy operation. These allow you to, for example copy variable values between variables in blocks with different names and copy only to variables with a spec of free.

To use values from a snapshot or result to set the current values for the variables in your simulation:

- From the Tools menu, click Snapshots.

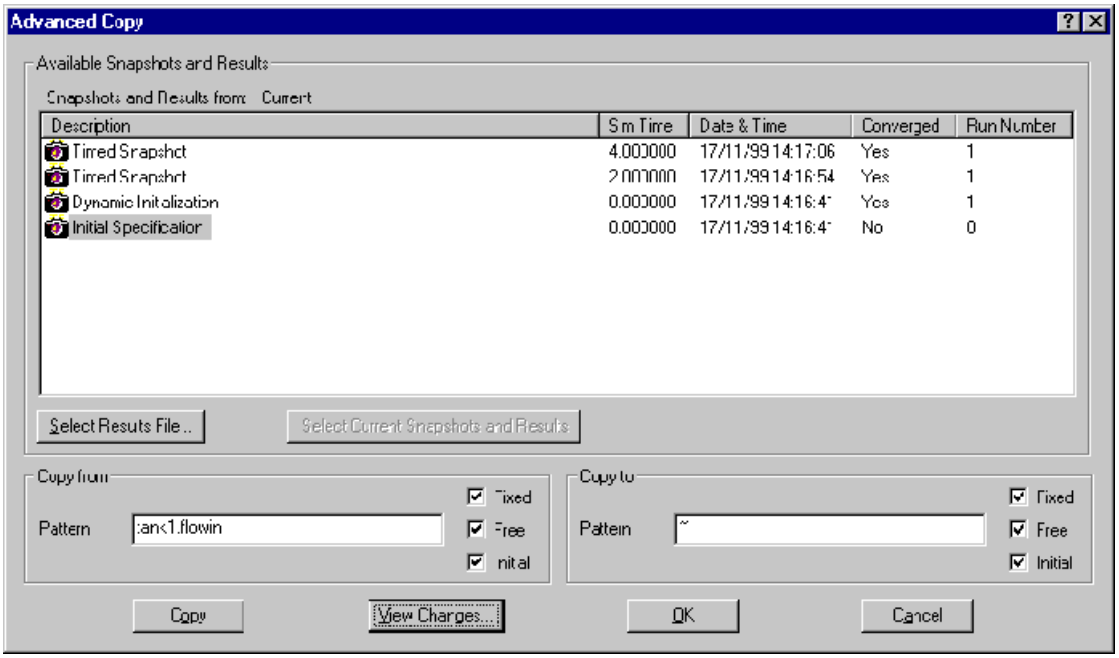
If you simply want to copy all matching variables between the snapshot or result and the simulation, use the Copy To button. If you want to perform a more complex Copy, click the Advanced Copy button. In the Advanced Copy dialog box, use the Copy From and Copy To boxes to control how data from a snapshot is copied into the simulation.

Note: You cannot use wildcards in the Copy From box.

Example of Copying Variables from a Snapshot

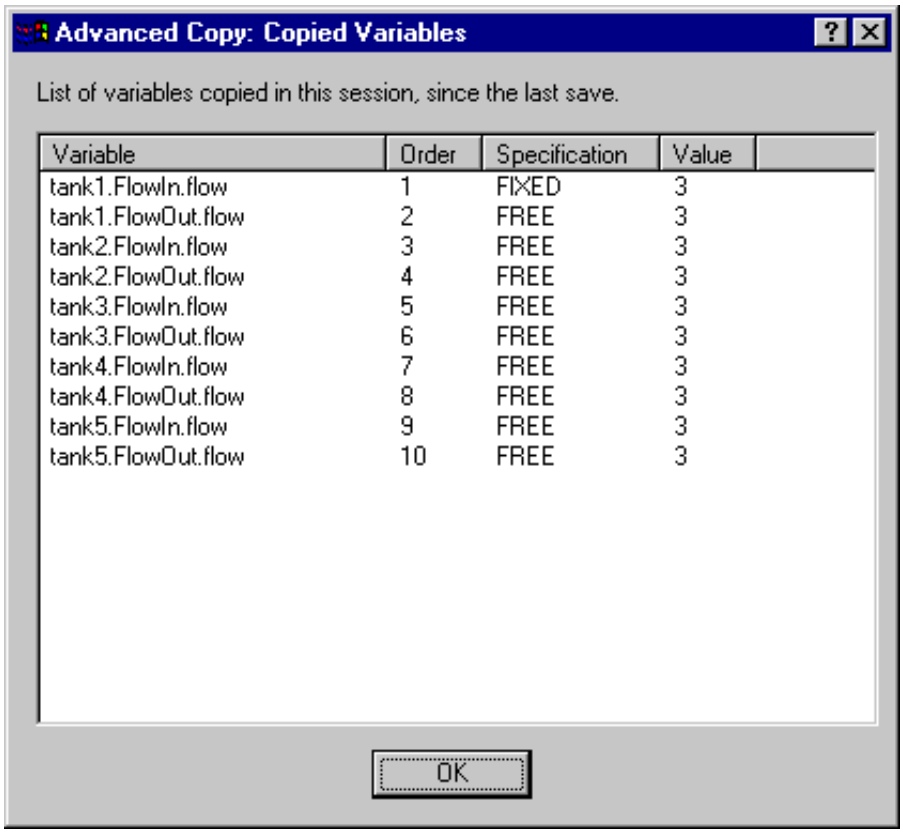
This example uses one of the files supplied with your AspenTech product.

1. Open the file fivetank.acmf.
   If you installed in the default location, this will be in the folder C:\Program Files\AspenTech\Aspen Custom Modeler 121\Examples\5tank.
2. From the Tools menu, click Snapshots, and then in the Snapshots Management dialog box, click the Advanced Copy button.
3. In the Available Snapshots and Results list, click Initial Specification to select the snapshot. This snapshot contains the values for all the active variables when they were initially loaded.
4. In the Copy From box, type Tank1Flowin.
5. In the Copy To text box type ~ and ensure the Fixed check box is selected, as well as Initial and Free:



6. Click Copy.
   A dialog box appears telling you there are 10 matches:
7. Click OK.

8 To see which variables matched, click View Changes:



9 In the Advanced Copy dialog box, click Cancel to lose your changes.

Explanation

In the Copy From box, variables in tank1.flowin were selected. There is only one variable in tank1.flowin, that is, tank1.Flowin.Flow.

The path for this variable relative to tank1.Flowin is Flow. So the variables that will be copied to are those whose complete path matches ~.Flow.

Tip: If you enter this string in the Variable box of Variable Find, you will get the same list you obtain in the Advanced Copy dialog box.

Example of Copying Variables with an Explicit Path

This example copies only the Flowin.Flow variables in the file fivetank.acmf:

1 Using the fivetank.acmf example, from the Tools menu, click Snapshots, and then, in the Snapshots Management dialog box, click the Advanced Copy button.

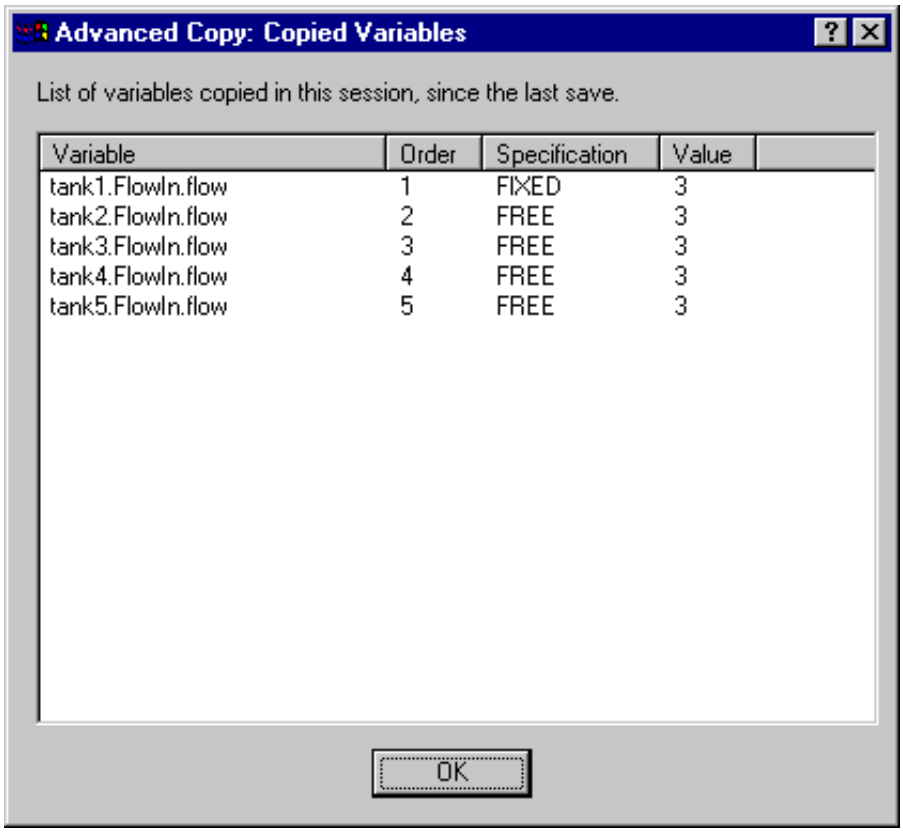
2 In the Available Snapshots and Results list, click Initial Specification to select the snapshot.

3 In the Copy From box, type Tank**1**Flowin.Flow.

4 In the Copy To text box type ~.Flowin.Flow and ensure the Fixed check box is selected, as well as Initial and Free.

5 Click Copy.
A dialog box appears telling you there are 5 matches.

6 Click OK, then click View Changes:



Note: If you did not cancel the Advanced Copy dialog box in the first example and restart, your list will contain these matches appended to the original list, because the changes have accumulated in the Advanced Copy dialog box.

Explanation

An explicit variable path was given in the Copy From box. The path of the variable relative to itself is an empty string. Therefore, nothing is prefixed to the Copy To pattern and it is used unmodified to match destination variables.

Examples of Unexpected Results

The following patterns can give unexpected results. The example file used is fivetank.acmf, a file that is supplied with your AspenTech product.

Example 1

If you copy from tank1.flowin to the inlet flows of the other tanks using tank1.Flowin in the Copy From box, and ~.Flowin.Flow in the Copy To box, the result will be no matches.

The Copy From variable is tank1.flowin.flow with a relative path of **flow**. This results in a complete Copy To pattern of **~.Flowin.Flow.Flow**, which does not match any variables.

Alternatively, if in trying to perform the same task you enter tank1.Flowin.Flow in the Copy From box and ~ in the Copy To box, 35 matches will be found and View Changes will list all the active variables.

This is because an explicit variable path gives an empty relative path, so the Copy To string is used unmodified. The ~ in the Copy To box therefore matches all the active variables in the simulation.

Import and Export Snapshots and Results

From the Snapshot Management dialog box, you can import and export snapshots and results to and from the current simulation:

- When the currently selected snapshot or result is **exported**, it is written in ASCII form to a file with the extension **asnp**. Files with this extension can be imported into any other simulation.
- When a result is **imported**, it is displayed in the Snapshot Management dialog box as a kept result. This means that when you save the simulation, the result will be written into the simulation input file.

Display Simulation Messages

To view messages about the state of the current simulation and any significant events or errors, make sure the Simulation Messages window is displayed. To switch the Simulation Messages window on and off:

- From the View menu, click Messages.
  Messages scroll up as new messages arrive from the system.

To display a menu of operations you can perform from the Simulation Messages window:

- Click the right mouse button anywhere in the Simulation Messages window.

Change Solver Reporting Level

To change the level of detail in the information you receive at run time:

1 Make sure the Simulation Messages window is displayed.

2 In the Simulation Messages window, click the right mouse button and then point to Solver Reporting Level.

3 From the list, click a value to specify the level of output during solution. The values range from None (fatal error messages only) to Maximum (very detailed run-time output information).

Use Homotopy

To use homotopy for a steady-state or an initialization run, you must define which fixed variables to use and their current and target values, specify the options for homotopy, and enable homotopy.

Caution: Do not change the homotopy variables with the Simulation Access eXtensions (SAX) while you are performing a homotopy run.

To use homotopy:

1 From the Tools menu, click Homotopy.

2 In the Homotopy dialog box, add the fixed or initial variables to be used to the Homotopy list box. You can either:

- Type the name of a fixed variable in the Variable text box.
- – or –
- Click Find and drag and drop fixed variables from Variable Find onto the Homotopy list. You can also drag and drop fixed variables from tables.

Tip: To define multiple Homotopy sections, click Script. This enables you to create a script containing the current homotopy specification. Running such a script will restore a previously created homotopy specification in place of the current one.

3 To change the values and set targets of variables in the Homotopy list, double-click a variable name and enter the new value and the target value in the Change Values dialog box.

4 To specify the options for homotopy steps, click Options.

5 Type the values for the options you want to change, and then click OK.

6 Select the Enable Homotopy check box. The next run will have homotopy enabled, and will use the fixed variables and options you have specified.

Notes:

- To see whether you have homotopy enabled for the current run, look at the Status bar.
- Messages in the Simulation Messages window from the homotopy control that refer to values of homotopy variables are in the *base units* for the variable, not the selected Units of Measurement.

Homotopy

Homotopy provides a way of moving from one converged solution to another solution with different values for one or more of the homotopy variables. It is

a useful technique where it is difficult to obtain convergence for a particular specification, but where you already have a converged solution for a different specification.

Homotopy works by moving along a path to the new solution, and solving a number of interim points along the way.

Let HOM1 be the vector of values for the homotopy variables at the point you have already solved, and HOM2 is the vector of values for the point to which you want to move. In a homotopy simulation, your AspenTech product attempts to solve a number of points at values for the homotopy variables of:

Homotopy = HOM1 + *theta* * (HOM2 – HOM1)

Where:

Theta	=	Homotopy parameter. This is a number that is moved from 0 to 1 at the successive solutions. When *theta* is 1, this corresponds to the specification HOM2. You can control the way theta changes between successive solutions.

Change Updating of the User Interface

Normally, the user interface updates after each step of a dynamic simulation. However, you can change the frequency with which the user interface is updated. Low frequencies can improve simulation times, especially where the simulation server and the user interface are on the same machine, but this can introduce a delay between the current state of the solution and the user interface.

To change the update frequency of the user interface:

1 From the Run menu, click Run Options.

2 In the Run Options dialog box, change the Synchronization Frequency box to reflect the behavior you require:

For	Select
Normal behavior	Full
Infrequent updates of the user interface	Low
Medium number of updates of the user interface	Medium
Frequent updates of the user interface	High

Note: You can also use Automation to change the user interface update frequency, using the SyncSteps property.

Reviewing Results

Forms for Reviewing Results

You can view results using the following forms:

- Plots, which display results graphically
- Tables, which display results in a grid
- ActiveX Control Forms, (ocx forms) which are forms you can create using reusable software components such as text boxes, check boxes, list boxes, and command buttons that have been developed by software vendors

Note: You can make any of these forms the default for viewing the results of a model.

Form Configuration Data in Aspen Custom Modeler 12.1

Aspen Custom Modeler 12.1 now allows you to edit instances of forms as well as the form definitions. For example, if you define a model-level form, it is possible to redefine or customize it for each block using that model.

Customizations

Customizations can include the form size, variables displayed in it and other elements. It is also possible to copy a model which contains a customized table form from a library to Aspen Custom Modeling without loosing customizations.

For example, an instance of a form [e.g. B1.ResultsPlot] might have an entirely different definition to that of the original form in the model type. To reset it to the original, right-click the form and select Reset Definition. This applies to inherited forms, forms in libraries, etc.

Customizations are stored, so next time you open a form for a particular block, the customizations are still present. They are also stored in the input file when the file is saved.

The following conditions apply to input files:

- Aspen Custom Modeler 12.1 reads 10.2 input files correctly but 10.2 looses any form definitions in an 12.1 input file.
- Aspen Custom Modeler 12.1 will not load 10.2 binary input files or libraries.

Form Type Definitions

Form Types have been removed from Aspen Custom Modeler 12.1 and Form type definitions no longer appear in libraries. In Aspen Custom Modeler10.2 and earlier versions if you followed the procedure:

1 Add Form definition to add a custom form definition <MyFormDef>

2 Create library

3 Load library

4 Then in the library you would see a custom form definition <MyFormDef>

5 If you attempted to open this form you would get a disabled instance of <MyFormDef>

6 To open an instance of this form you would first have to add an instance of it either to the flowsheet or to a model definition

It was possible but uncommon to implement forms such that they are intended to be instantiated by double-clicking them in the library without placing them either on the flowsheet or in a model, and not be disabled as described in step 5.

This is no longer possible. The workarounds are

1 Place the instances of your forms in a model in the library. You could potentially gather them all together under a model created for the purpose

2 Place instances of your forms in the flowsheet of any input file

Create scripts which use the new automation method LaunchCustomFormView.

Note: It is no longer possible or necessary to use the 'Add Form Definition' item in the ACM explorer under custom modeling.

When an original form definition changes, the block-level instances of that form are automatically updated, unless they have already been customized.

To update any customized instances, right-click the forms and select 'Reset Definition'.

Adding OCX Forms to a Model

To add an ocx form to a model:

1 Select the model definition

2 Select 'Add Form' - give the form a name <MyForm> and choose the ocx.

3 Close the disabled instance of the form which appears

4 In an instance of the model, select <MyForm> to view the form.

Adding OCX Forms to a Flowsheet

To add an ocx form to a flowsheet:

1 In the flowsheet, select 'Add Form' - give the form a name <MyForm> and choose the required ocx.

The automation method LaunchLibraryFormView will still work where the input file has been modified by hand to include a flowsheet-level custom form, but users should now migrate to the new method LaunchCustomFormView.

Work with Plots

Plots are used to display a graphical representation of results. Your AspenTech product enables you to:

- Create new plots
- Change the properties of plots

Create a Plot

You can create three types of plots:

- Run-time plot: a plot of a set of variables against time
- Profile plot: a plot of a series of variables, for example, temperature profile, against an index (the number of trays within a column or the number of points used to discretize a partial differential equation - an equation that contains both spatial and time derivative terms, see below) or against another series of variables, for example, the length of a reactor bed:

$$\frac{\partial T}{\partial t} = \frac{k}{C_p \rho}\left(\frac{\partial^2 T}{\partial x^2}\right)$$

- Phase Plot: a plot of a set of variables against another variable on the X axis. The data plotted is time based but time is not shown on the plot.

Create a Run-Time Plot

You can create a run-time plot in two ways:

To	Do this
Plot a set of variables from different blocks	Create a flowsheet plot
Pre-configure plots for all blocks that use the same model	Create a model plot

Create a Profile Plot

You can create a profile plot in two ways:

1 To plot a set of variables from different blocks, create a flowsheet-level plot.

a) From the Tools menu click New form or click the New form button on the toolbar.

b) In the New Flowsheet Form dialog box, enter a name for the form and select Profile Plot from the Available Form types.

c) The new plot appears.

2 To pre-configure plots for all blocks that use the same model, create a model-level plot.

a) In the Simulation Explorer, ensure the Custom Modeling folder (or your required library) is expanded. If necessary, expand Models, and click the required model.

b) In the Contents pane of the Explorer, double-click Add Form.

c) In the Add Form Instance dialog box, enter a name for the form and select ProfilePlot from Available Form Definitions.

d) A plot appears, and a new icon appears in the Contents pane, showing a new plot template.

Note: To view a model-level profile plot, create a block which uses that model and open the plot for that block.

Adding Variables

The procedure for adding variables to the plots is very similar in both cases, with the following difference:

- In flowsheet-level plots, you need to prepend the block name to the variable names, e.g. B1.X.
- In model-level plots, you just need to give the variable name within the model, e.g. X.

Create a Phase Plot

You can only create a phase plot by first creating a Run-Time plot. Once you have a Run-Time plot displayed you can convert it into a phase plot by either dropping a suitable variable onto the X axis or using the plot property page.

To access the plot property page:

1 Click the right mouse button on the plot, then click Properties.

2 Click the Variable tab, and then type the required variable name into the Specify X-Axis variable field and click the associated radio button.

You can switch back to time on the X axis by either clicking the X-axis as time radio button on this page or using the Time on X-Axis menu item on the plot context menu.

Phase plots have a time range facility which allows you to select the time range of the data that will be plotted.

To access this:

1 Select Time Range... from the phase plot context menu. This displays a dialog that allows you to select a start and end time for the plot.

2 Check the Use current simulation time check box, the plot is updated when new data is generated.

The Profile Editor

To add variables to the plot template:

1 Click the right mouse button (or press the context menu key) on the plot and then click Profile Variables.

2 The Profile Editor dialog appears.

There are three distinct styles of profile plot. These are:

- 2d Plots
- 3d Time Plots

- 3d Surface Plots

2d Plots

The x-axis represents an index of the variables in the y-axis profile, or you can optionally define an x-axis profile. It is useful to define an x-axis variable when the y-axis profile variables are distributed over distance, and items are not equally spaced.

It would be possible to have this spacing vary with time. You may plot more than one profile on the y-axis.

- To create this type of plot select the 2d plot button on the profile editor.

3d Time Plots

These are very like 2d plots; the profile variables are plotted on the z-axis, the x-axis behaves as in the 2d plots described above and the y-axis represents time; the plot surface is extended as time progresses. However, you may plot only one profile on the z-axis.

- To create this type of plot select the 3d plot button on the profile editor and in the Y - axis profile box, select the Time button.

To see an example of a 3d time plot, see the FiveTanks example.

3d Surface Plots

3d surface plots depict two dimensional arrays against two indices, the x- and y- axes. Profiles can be assigned to both of these axes, for example In a heated slab of material you could plot temperature against the x and y distances across the slab. You may plot only a single surface at a time.

- To create this type of plot select the 3d plot button on the profile editor and in the Y - axis profile box, select the Specify profile button.

You can optionally supply different time settings for the plot, or specify a time interval, so that profiles are added at this interval during a dynamic run. If you pre-define the profile intervals, the plot refreshes as the data becomes available.

- To see an example of a 3d surface plot, see the Heated Slab example.
- See also Interacting with 3D Profile Plots.

Adding Variables to a Profile

To plot variables on a profile plot, you create a profile, add variables to that profile and then assign that profile to your chosen axis.

Creating a profile

1 In the profile builder box of the profile editor, click the  button, or click on the list of profiles and press Insert.

2 Enter a name for the profile; double click the name to edit it.

3 If the profile is to represent a 2d surface, right-click the profile and select Mark Profile as 2D [or press Ctrl-D] (the dimension is shown in the last column of the profiles list). Otherwise, the profile will be one-dimensional; the default setting.

Add variables to that profile

1. Highlight the profile you wish to add variables to.
2. In the Profile Variables list, click the  button or press Insert, then type the name of the variable you wish to place in this profile.
3. Repeat until you have added all the variables you wish to this profile. You can change the order of these variables by dragging them up and down the list, or by pressing Alt and the Up or Down arrow keys.

Assign that profile to your chosen axis

1. Ensure that your profile is highlighted and click the >> Plot button. This assigns the profile to the y-axis of a 2d plot or the z-axis of a 3d plot.
2. The 'Profiles to Plot' list shows the profiles which will be plotted on the vertical axis; you can change the order in which they are plotted by dragging them up and down the 'Profiles to Plot' list.
3. To remove a profile from this list highlight it and then click the  button or press Delete.

Hints & Tips

Click the Find Variables... button to search for variables in the current context.

- You can drag and drop variables from the variable finder or a table to the profile editor.
- If you drop variables on the (top) list of profiles then a new profile is created and those variables assigned to it.
- If you drop variables on the (bottom) list of variables then those variables are added to the currently selected profile.
- If you drop a single member of an array on the profiles list, it assumes that you want to plot the whole array and substitute a wildcard, e.g. dropping B1.X.Value(3) on the top list creates a new profile called New Profile with the variable B1.X.Value(*).
- You can drag profiles from the top list to, for example, table definitions and other instances of the profile editor. The variables that make that profile up are dragged, not the profile name. You can copy a profile by dragging it from the list of profiles and dropping it in that same list.
- You can use wildcards.

Using Wildcards

The following table details wildcard options:

To	Enter this
Plot the contents of the array T in the block R1 in a flowsheet plot	R1.T
Plot the temperatures of all stages in a column in a model plot stage	(*).T
Plot a one-dimensional distribution variable T in the block R1 in a flowsheet plot	R1.T.value

Plot a slice over one dimension of a two-dimensional variable T	T.value(*,2)
Plot a 2d slice of a three-dimensional variable Temp	Temp.value(*,2,*)
Plot the heights of liquid in the fivetank example, create a profile with one variable or create a profile with five variables	Tank*.h (or) ~.h Tank1.h, Tank2.h, Tank3.h, Tank4.h, Tank5.h
To specify a 2d surface of Temperature in a heated slab, create a profile with one variable or create a profile with several variables	T(*,*) T(*,0), T(*,1), T(*,2), T(*,3)

Note: When using wildcards to specify a 2d surface, the following restrictions apply:

1 Only use the * wildcard (not ~ or ?).

2 In each case the wildcards must represent a continuous range of integers, e.g. 3,4,5,6.

Specify x-axis variables

The x-axis of both 2d and 3d plots and the y-axis of 3d surface plots represents an index of the variables profile, or you can optionally define an x-axis profile. It is useful to define an x-axis variable when the y-axis profile variables are distributed over distance, and items are not equally spaced. It would be possible to have this spacing vary with time.

To specify variables that are plotted on these axes, create a profile in the normal way, e.g. X Length with variables X.Value(*), and click the >> X Axis button to assign this profile to the x-axis, or the >> Y Axis button to assign this profile to the y-axis.

To remove this assignment click the  button or press Delete.

Time Settings for Profile Plots

On the Time Settings tab of the profile editor you can specify different times for the updating of the plot.

These settings are:

Continuous update

- On 2d plots and 3d surface plots, the plot is redrawn as new data becomes available.
- On 3d time plots, a new line is added to the y [time] axis as new data becomes available.

Specify Times

- On 2d plots and 3d time plots, there is one line on the y-axis for each of the specified times.

- On 3d surface plots, the plot is updated as each of the specified times is reached.

Specify Interval

- The plot is updated at the specified interval, beginning with the given time. The Max. time points setting determines how many different times can be plotted at once.
- On 2d plots and 3d time plots, there is one line on the y-axis for each of the specified times - old times are removed from the plot.
- On 3d surface plots, the Max. time points setting is ignored.

Interacting with 3D Profile Plots

You can interact with the chart as it's running to examine data more closely or visually isolate a part of the chart. The interactions described here affect the chart displayed inside the body of the Chart; other chart elements like the header are not affected.

Select Chart Interactions or Additional Options below for more details:

- Chart Interactions
- Additional options

Chart Interactions

Rotation

1 Hold down both mouse buttons (or the middle button on a 3-button mouse), and:

- to rotate freely, move the mouse in the desired direction, or,
- to constrain rotation along an axis, press the "x", "y", "z", or "e" key and move the mouse perpendicular to that axis.

Translation

1 Press Shift, hold down both mouse buttons (or the middle button on a 3-button mouse), and move the mouse to change the positioning of the chart inside the plot window.

Scaling

1 Press Ctrl, hold down both mouse buttons (or the middle button on a 3-button mouse), and move the mouse down to increase chart size, or up to decrease chart size.

Zooming

1 Press Ctrl, hold down the left mouse button, and drag mouse to select zoom area and release mouse button.

Reset to Automatic Scale and Position

1 Press the "r" key to remove all scaling, translation, and zooming effects. Rotation is not removed.

Additional Options

Additional options are provided by right-clicking on the chart. These are:

Option	Effect
Redraw	Redraws the chart without refreshing the data from the server
Zoom In	When the y axis represents time, zooms in on that axis
Zoom Out	Undoes a zoom in operation
Zoom Full	Zoom as far out as possible and refreshes the chart data
Time range	Allows you to specify the range of times on which to zoom in and the size of the interval between data points in that range
Properties	Allows you to customize the appearance of the chart, the chart titles, shading, etc.
Reset form properties	Resets the definition of this instance of this form back to the original
Copy as metafile	Copies a picture of the chart to the clipboard in enhanced metafile format
Copy as bitmap	copies a picture of the chart to the clipboard in bitmap format
Reset position	Undoes the scaling, translation, zooming and rotation effects described above
Profile variables	Displays the profile editor for configuring profile plots
Profile table	Creates a table of the data displayed in the plot

Work with Tables

Tables are used to display a tabular representation of results. Your AspenTech product enables you to:

- Create new tables
- Open existing tables
- Change the properties of tables
- Reset variable attributes to their default values
- Ramp variable values

Create a Table

You can create either:

- A flowsheet table: a table of a set of variables from different blocks.
- A Hierarchy table:a table of a set of variables in a hierarchy.
- A model table: a table of all the blocks that use the same model
- A Time History table: a table of an existing plot, showing a set of values for a variable, recorded over a period of time, at communication intervals set in the Run Options dialog box.
- A Profile Table**:** A table of the historical data used to generate a profile plot, showing a set of values for a profile, recorded over a period of time.

Note: You can also create a flowsheet table using Variable Find.

Create a Flowsheet Table

A flowsheet table is used to tabulate one or more variables from anywhere in the flowsheet.

To create a flowsheet table:

1. From the Tools menu, click New Table.
2. In the Enter Flowsheet Table Name dialog box, enter a name for the new table.
   The new table appears.
3. Do one of the following:

   Open one or more tables containing the variables you want.

   – or –

   Create a list of variables using Variable Find.
4. Drag and drop the required variables onto the table.

Note: You cannot drag and drop a stream variable marked with the **<** or **>** character.

Tip: Instead of using drag and drop, you can type the variables names. To do this, click the right mouse button on the table, then click Properties. Click the Variable tab, and then type the required variable names in the Variable Name box.

Create a Hierarchy Table

1. In the Simulation Explorer, make sure the Flowsheet icon is expanded. Find the node beneath this which represents the required hierarchy. Click the required hierarchy and then in the Contents pane, double-click Add Form.
2. In the Add Form Instance dialog box, type a name for the form. In the Available Form Definitions list, click Table and then click OK.
   A table appears, and a new icon appears in the Contents pane.
3. To add variables to the table template, drag and drop variables from the AllVariables Table for the hierarchy.
4. By default the table will only show the value and spec attributes of a variable. To display additional attributes, change the properties of the table by clicking the right mouse button, and then click Properties.
5. To quit the table, click the Close button.

Create a Model Table

To create a model table:

1. In the Simulation Explorer, make sure the Custom Modeling folder or your required library is expanded and if necessary, expand Models. Click the required model and then in the Contents pane, double-click Add Form.
2. In the Add Form Instance dialog box, type a name for the form. In the Available Form Definitions list, click Table and then click OK.
   A table appears, and a new icon appears in the Contents pane.
3. To add variables to the table template, drag and drop variables from the AllVariables Table for the model.

Note: Both the table template and AllVariables table are gray.

4. By default the table will only show the value and spec attributes of a variable. To display additional attributes, change the properties of the table by clicking the right mouse button, and then click Properties.
5. To quit the table template, click the Close button.

Important note: To view results, you must first click a block (model instance). Any variable assignments such as value, spec, description, made in a model will be displayed in the block table. Similarly, array variables displayed as a pattern in the model table, will be shown expanded in the block table because the expansion relies on parameter assignments and these are done in blocks.

Create a Table from Variable Find

To create a table from Variable Find:

1. From the Tools menu, click Variable Find, and then click Find.
2. Hold down the CTRL key, and then click each variable you want to select from the list at the bottom of the Variable Find dialog box, and then click Table.

Tip: To select all the variables in the list, press CTRL+A.

3. In the Variable Find dialog box, type a name for the table and then click OK.
   The newly generated table appears and a new icon appears in the Contents pane for Flowsheet, showing the new table.

Open a Table

You can open either:

- A flowsheet table: a table of a set of variables from different blocks.
- A model table: a table of all the blocks that use the same model.

Open a Flowsheet Table

To open an existing flowsheet table:

1 In the All Items pane of the Simulation Explorer, click Flowsheet.

2 In the Contents pane, double-click the icon for the table to display it.

Open a Model Table

To open an existing model table:

1 In the All Items pane of the Explorer, make sure the Custom Modeling folder or your required library is expanded, and if necessary, click the plus sign (+) in front of Models.

2 Click the name of the relevant model.

3 In the Contents pane, double-click the icon for the table to display it.

Change the Properties of a Table

To change the properties of a table:

1 Make sure the table is open and in the active window. Click the right mouse button, and then click Properties.
The Table Edit Properties dialog box appears.

2 Transfer an attribute to the Selected Attributes area by doing one of the following:

- Click the relevant attribute and then click the right arrow button to transfer attribute to the Selected Attributes area


  – or –

- Double click the relevant attribute. The attribute is automatically transferred to the Selected Attributes area.

3 Click OK to apply the changes and close the dialog box.

Reset Variable Attributes from a Table

To reset variable attributes back to their default values:

1 Open a table displaying the variables whose attributes you wish to reset.

2 Click the right mouse button on the cell you want to reset.

3 Click Reset to Default.
The value changes to the default value.

Note: You can also reset variable attributes from Variable Find.

Ramping Variable Values from a Table

For any fixed variable on a table, the context menu has an item labeled Ramp.

This brings up a dialog which allows you to start ramping the value of the variable. You can set the duration and slope, the start time being the current time.

Note: The value field of the variable changes to red while it is being ramped.

Ramps are cancelled on:

- rewind,
- restart,
- reset,
- any manual time change.

Add a Control Form to a Model

To add a control form to a model:

1. Create a Microsoft ActiveX control form, use the Compile to Native Code option to compile the form to an .ocx file, and add the new .ocx file to the PC registry.
   For information on these procedures, see the Microsoft documentation on ActiveX.

   Note: If you include the "Aspen Custom Modeler 12.1 Type Library"/"Aspen Dynamics Type Library" in the Visual Basic project references, and use implements IAspenModelerEvents in the body of your control, then you will be able to receive events. Events are generated for conditions such as run mode or UOM set changing, runs starting or stopping. Events are automatically delivered to all user forms which implement IAspenModelerEvents. See the example 'Handling Events in a Visual Basic Form'.

2. Highlight the required model in Simulation Explorer.
3. Double-click Add Form and enter a name for the new form. Select Custom OCX... in the form type list.
4. From the Select Control list, click the required ActiveX control and then click Select. A new icon appears, showing a new instance of the form type.

Note: The form is not sized to that of the OCX used. To save the new sizing information, you must re-size the form and then close it.

Make a Form the Default Form for a Model

To make a form (table, plot, or OCX form) the default for a model:

1. In the Simulation Explorer, click the form that you want to be the default for the model.
2. Click the right mouse button, and then click Make Default.
   The chosen form is now the default form for viewing results.

Define Patterns

You can use patterns, which may contain wildcards, in a string text, to produce a match for an Advanced Copy operation:

Symbol	Meaning
*	Finds zero or more characters for the current name field
~	Finds zero or more name fields
?	Finds any single character present in the specified name field

In the Advanced Copy dialog box, you can also use results from an existing block to provide starting values for a new block.

Work with Variable Find

You can use Variable Find to create a list of variables that you specify. From this list you can create a script or a table, or change the properties of a variable.

Click the following links to display information about working with Variable Find:

- Accessing Variable Find
- Using Variable Find

Access Variable Find

You can access Variable Find in several ways:

- From a block or stream on a flowsheet
- From the Status window
- From the Tools menu

Access Variable Find from a Block or a Stream

To access Variable Find from a block or a stream:

- In the flowsheet window, click a block or stream, then click the right mouse button and click Find.
  The Variable Find dialog box appears, showing the name of the block or stream in the Find In text box.

Access Variable Find from the Status Window

To access the Variable Find from the Status window:

1. From the View menu, click Status Window.
2. On General Tab, in the Find area, click the required spec property that you want to use for your search criterion.
   The Variable Find dialog box appears with the spec you requested in the Specification text box.

Use Variable Find

You can use Variable Find to create a list of variables that you specify. From this list you can create a script or a table, or change the properties of a variable.

To use Variable Find:

1. From the Tools menu, click Variable Find.
   The Variable Find dialog box appears.
2. Specify the required options for your search criteria and then click the Find button to perform the search.
   A list of variables and parameters matching the search criteria is created, with a header containing Name, Value, Spec, and Variable Type.

Tip: You can use the Find or Find More buttons as follows:

To	Click
Append a new list of variables matching a Search criterion to a list of variables from a previous Find operation	Find More
Clear the previous list and create a new list	Find

3. When you have completed your search, click the Close button to quit Variable Find.

Note: To reset variable attributes back to their default value, select the Reset to Default check box next to the relevant attribute, and then click OK or Apply.

View Results for a Block or a Stream

To view block or stream results:

1. In the flowsheet window, click the block or stream, then click the right mouse button and then point to Forms.
2. Click the form (table, plot, or ActiveX control) you want to use for displaying results. By default, each block has a table called AllVariables, which shows all the variables for the block.
3. To close the form, click the Close button.

View Results for Flowsheet Variables

To view results for flowsheet variables:

1. In the All Items pane of the Simulation Explorer, click Flowsheet.
2. In the Contents pane, double-click the LocalVariables table.

   All variables defined in the flowsheet Constraints section are displayed.
3. To close the table, click the Close button.

View Results for an Optimization Run

To view results for an optimization run, do one of the following:

1. Depending on where the variables are defined:

For variables that are	Do this
Defined in the flowsheet Constraints sections	In the All Items pane of the Simulation Explorer, click Flowsheet and then double-click LocalVariables table.
Defined in models	Use Variable Find to find variables of the type SlackVariable, and variables with the spec Objective and/or Vary. You can then create a table from the resulting list.

2. To display Lagrange multipliers, click the table with the right mouse button, and then click Properties.
3. Select the ULagrange and Llagrange attributes from the Available Attributes list and click the right arrow to transfer them to the Selected Attributes list.
4. Click OK to apply the changes and close the dialog box.

   The table now includes columns for Lagrange multipliers.

Results Displayed for Failed or Interrupted Runs

Results are available in the form of tables and plots, or through automation. They show the values of variables in your flowsheet. The values displayed depend on:

- Type of simulation (such as dynamic or optimization)
- Status of simulation (such as converged or interrupted)

For example, consider a steady-state simulation. If the simulation converges, the results comprise variable values from the converged solution. But what variable values are presented when the simulation fails to converge?

To find out what variables are displayed for each simulation status, click the following simulation types:

- Steady-state or initialization simulation
- Dynamic simulation
- Estimation simulation
- Optimization simulation

Values of Variables for Steady-State or Initialization Simulations

If the steady-state or initialization simulation converges, then the variables take their converged values.

If the simulation fails to converge or is interrupted, then the values of the variables depend on the simulation status, as follows:

Status of Simulation	Values of Variables
Fails to converge	Variables take their unconverged values. See note, below.
Interrupted	Variables take their unconverged values from the solvers at the point of interrupt.

Note: If the run fails to converge, it will fail in one of the groups in the decomposition. Some variables will satisfy your equations (those appearing in groups before the failed group). The variables in the unconverged group will be the unconverged values at the last iteration of the solver. Variables in the subsequent groups have the values from the start of the run.

Values of Variables for Dynamic Simulations

If the dynamic simulation converges, then the variables take their converged values.

If the simulation fails to converge or is interrupted, then the values of the variables depend on the simulation status, as follows:

Status of Simulation	Values of Variables
Fails to converge	Variables take their unconverged values.
Interrupted	Variables take their unconverged values from the solvers at the point of interrupt.

Values of Variables for Estimation Simulations

If the estimation simulation converges:

- All variables except those estimated are reset to their values at the start of the simulation
- Estimated variables take their final converged values that fit the measured data

If the simulation fails to converge or is interrupted, then the values of the variables depend on the simulation status, as follows:

Status of Simulation	Values of Variables
Fails to converge	All variables except those estimated are reset to the values from the start of the simulation. Estimated variables take their unconverged values.
Interrupted	All variables take their current unconverged values from the solvers at the point of interrupt.

Values of Variables for Optimization Simulations

If the optimization simulation converges, then all variables take their converged values. For dynamic optimization, these are the values at the final optimization time.

If the simulation fails to converge or is interrupted, then the values of the variables depend on the simulation status, as follows:

Status of Simulation	Values of Variables
Fails to converge	All variables take their unconverged values. For dynamic optimization, these are the unconverged values at the final optimization time.
Interrupted	All variables take their current unconverged values from the solvers at the point of interrupt.

Access Decomposition Information

To access decomposition information:

1 In the All Items pane of the Simulation Explorer, click Diagnostics.

The active decompositions appear in the Contents pane.

Note: Decompositions will only appear in the Simulation Explorer after you have run a simulation.

2 To list the groups in a decomposition, double-click the required decomposition.

3 To list the equations in a group, double-click the required group.

4 To list the variables in an equation, double-click the required equation.

Tip: To avoid performance problems with large simulations, you can choose to view only the first unconverged block under the decomposition node of Diagnostics. To do this:

- In the All Items pane of the Simulation Explorer, click Diagnostics with the right mouse button and then select Show First Unconverged Only.

Access Global Parameters

Global parameters are defined in the modeling language, outside of type definitions. For example a global parameter could be the Universal Gas constant, acceleration due to gravity constant, and so on.

To access global parameters:

1 In the All Items pane of the Simulation Explorer, click Simulation.

2 In the Contents pane, click AllGlobals.

Table AllGlobals appears. If necessary, you can change the values of the parameters.

3 To close the table, click the Close button.

View Degrees of Freedom Information

You can view information about the number of equations and unknown variables for the following:

- Blocks and streams
- Simulation

View Degrees of Freedom Information for a Block or a Stream

To view degrees of freedom information for a block or a stream:

1 In the Flowsheet window, click a block or stream, then click the right mouse button, and then click Properties.

The Block Properties form appears, showing detailed information about the block or stream.

2 Click OK to close the form.

View Degrees of Freedom Information for a Simulation

To view degrees of freedom information for a simulation:

1. In the All Items pane of the Simulation Explorer, with the right mouse button click Simulation, and then click Properties.
2. Click the Degrees of Freedom tab to display detailed information about the degrees of freedom, equations, and variables.
3. Click OK to close the form.

Print

To print an item on the screen:

1. Click the item that you wish to print and from File menu, click Print.
2. Click OK to send the item to the printer.

Tip: If you want to see how the printed document will look before you print it, click the File menu and then click Print Preview.

Work with External Applications

You can paste results from your simulation into other applications and paste data from other applications into forms. For more information, refer to:

- Copy results into other applications
- Copy information from external applications

Paste Results Into Other Applications

To paste results in to other applications, for example, a Microsoft® Excel spreadsheet:

1. Open a table for a flowsheet, block, or stream.
2. Click the right mouse button on the relevant cell value and then click Copy.
3. From the Edit menu in the other application, click Paste Special and then select Paste Link and click an option in the As list. For example, in Microsoft Excel, select Paste Link, and then click As Text and click OK.
4. Repeat steps 2 and 3 until you have copied all the relevant data to the other application.

Paste Data From Another Application

To paste data from another application, for example, a Microsoft® Excel spreadsheet into a table:

1. Open the application containing the data.

2 From the Edit menu in the other application, click Copy.

3 In the table, click the right mouse button on the cell where you want the information to appear, and then do one of the following:

To	Click
Copy the value to a cell and ignore any changes to the original data	Paste
Copy the value to a cell and reflect any changes made to the original data	Paste Link

4 Repeat steps 2 and 3, until you have copied all the relevant data from the other application.

Work with Time History for Variables

By default, during a dynamic simulation, time history is not recorded for variables. This means that when a plot is displayed during a run, no data from before the time you display the plot is shown.

However, you can choose to record the time history for variables. You can then create time history tables and view and change the recorded information.

Record Time History for Variables

There are three ways to record a time history for variables in a dynamic simulation:

- From the Run Options dialog box, for all variables
- From a table
- From Variable Find

Record Time History for All Variables from Run Options

To record time history for all variables

1 From the Run menu, click Run Options.

2 In the Run Options dialog box, select the Record History for All Variables check box.

3 To apply your changes, click one of the following:

To	Click
Keep the dialog box open	Apply
Close the dialog box	OK

Note: When a plot is displayed, all variables on the plot are automatically switched to record on if they are not already on. They remain on after the plot is closed.

Record Time History for Variables from a Table

To record time history for a variable from a table:

1. In the Flowsheet window, click a block and then click the right mouse button. Point to Forms and then click the relevant table.

Tip: The AllVariables table shows all the variables in the block.

2. Change the properties of a table to include the Record attribute.
3. For the chosen variable, in the Record column, click False and select True, to switch on the recording of time history.
4. To close the table, click the Close button.

Note: When a plot is displayed, the selected variables on the plot will be automatically switched to record on if they are not already, and they remain on after the plot is closed.

Record Time History for Variables from Variable Find

To record time history for variables from Variable Find:

1. Use Variable Find to create a table of all the variables for which you would like to have a time history.
2. Select all the variables and then click Properties.
3. In the Record column, click False and select True, to switch on the recording of time history.
4. To close the table, click the Close button.

Note: When a plot is displayed all the variables on the plot are automatically switched to record on if they are not already, and they remain on after the plot is closed.

Create Time History and Profile Tables

History tables are based on existing plot definitions, so you can show an existing plot as a history table. Any changes made to variables (including their order) in a history table is reflected in the associated plot, and vice versa. Profile tables are a special type of history table based on profile plots rather than time plots.

You can create time history and profile tables:

- From the Tools menu
- From the Simulation Explorer

Create Time History or Profile Tables from the Tools Menu

After you have recorded time history for variables, you can create a flowsheet-level history table, or a profile table. To do this:

1. From the Tools menu, click New History Table, or New Profile Table.
2. In the dialog box, type a name for the new table and click OK.
3. To add variables to the table, drag and drop from tables, or from Variable Find, directly onto the table, and:

   For history tables:

   click the new table with the right mouse button, then click Properties. In the Properties dialog box, ensure the Variables tab is selected, type the name of a variable in the text box, then click the Add button.

   For profile tables:

   click the new table with the right mouse button, and select Profile Variables.

Create Time History or Profile Tables from the Simulation Explorer

After you have recorded time history for variables, you can create a history or profile table for any level you want in the simulation hierarchy. To do this:

1. In the All Items pane of the Simulation Explorer, click the level for which you want to display a history table (for example, Flowsheet).
2. In the Contents pane, double-click Add Form.
3. In the Add Form Instance dialog box, type a name for the new form, then from the Available Form Definitions list, select the table type you require and then click OK.
4. To add variables to the new table, drag and drop from tables, or from Variable Find, directly onto the new table, and:

   For history tables:

   click the new table with the right mouse button, then click Properties. Ensure the Variables tab is selected, type the name of a variable in the text box, then click the Add button.

   For profile tables:

   click the new table with the right mouse button, and select Profile Variables.

View Time History and Profile Tables

To view an existing time history table:

1. In the All Items pane of the Simulation Explorer, click the folder in which the time history or profile table was created.

2 In the Contents pane, click the name of the table you want to open, then click the name again with the right mouse button, and click Show as History or Profile Table.

Tip: You can also view a time history or profile table by opening the associated plot and then clicking the right mouse button on the plot and clicking Show as History or Profile Table.

Change Time Settings for a History or Profile Table

To change the times for which data is displayed in a history or profile table:

1 Right-click the table and select Properties.

2 Click on the Time Settings tab.

Copying and Exporting Time History Data

From a time history table, you can copy history data to the clipboard for use in other Microsoft Windows applications, or write it to an external file for data entry to DMCplus®. To do this:

1 Open the time history table.

2 Click the history table with the right mouse button, and then:

To	Click
Copy the whole table or selected rows to the clipboard	Copy
Write history data to an external file	Send to File

Modeling

Creating Types

Create a Component List

A component list contains two types of information:

- A list of component names.
- A list of options associated with these components. Typically this is used to store options for calculating physical properties for mixtures of these components.

A component set is a simplified version of a component list that does not include a list of physical property options.

In general you should use a full component list when using a physical properties package such as Aspen Properties, and a component set when you are not using a physical properties package.

Tip: You can convert a component set into a component list and vice versa. To do this, select the required component list or set in the Contents pane for Component Lists, click the right mouse button and click Convert. The Convert dialog box appears asking whether you want to convert the selected set to a list or vice versa. Click Yes to carry out the conversion.

Create Component List

A component list is a list of component names that are validated against a physical properties system. To create a component list:

1 In the All Items pane of the Simulation Explorer, click Component Lists.

2 In the Contents pane, double-click Add Component List.

3 In the Create Component List dialog box, enter a name for the new component list.

The new component list appears in the Contents pane.

4 Double-click the new list.

The Build Component List dialog box appears.

5 From the Components list, select the components you require by doing one of the following:

Double-click required component.

– or –

Click the component and then click the right arrow button.

Tip: To select all the components, click the right double arrow button in the Build Component List dialog box.

6 When you have selected all the required components, click OK.

Create a Component Set

If you don't need to use a physical property package for your simulation, but still want to use the ComponentList parameter to size arrays etc. you can create one or more Component Sets (Component Sets are simply sets of strings).

To create a component set:

1 In the All Items pane of the Simulation Explorer, click Component Lists.

2 In the Contents pane, double-click Add Component List.

3 In the Create Component List dialog box, enter a name for the new component list and then check the IsComponentSet option.

The new component list appears in the Contents pane.

4 Double-click the new list.

The Build Component Set dialog box appears.

5 In the Edit or Add Component text box, type a name for the component and then click the Add button.

6 When you have added all the required component names, click OK.

Create an Icon

To create an icon:

1 In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded and, if necessary, expand Models, and then click the required model.

2 In the Contents pane, double-click Add Icon.

3 In the Create Icon dialog box, enter a name for the icon. The Icon Editor window and the Icon Draw toolbar appear.

4 In the draw pane of the Icon Editor window, use the Icon Draw toolbar to draw the icon. To make an icon appear as a 3D shape, click the icon and then click the right mouse button to access the pop-up menu:

Icon	3D Options
Box	Cylinder (upright) Cylinder (sideways)
Triangle (polygons)	Cone
Ellipse	Sphere

You can also import a bitmap by using the Import DXF item on the File menu. This creates a DXF tab from which the bitmap can be dragged and dropped onto your icon as many times as you want. Note that the bitmap can be resized once placed.

5 From the Port tree view, click the port you want to add. Hold down the mouse button and drag the selected port to the desired position on the icon perimeter. The port's position determines where it appears when it is placed on a flowsheet.

6 From the File menu, click Save.

7 To quit the icon editor, click the Close button.

Note: If you do not save your changes, the Save Icon Confirmation dialog box appears. If you want to save your changes click Yes. If some of the ports are not on the perimeter of the icon, the AutoPort confirmation dialog box appears. If you wish to apply the AutoPort command, click Yes.

Create a Library

To create a library:

1. From the File menu, click Create Library.

   The Save As dialog box appears.

2. Type a name for the file in the File Name box and then click Save.

The Create Library command consolidates the contents of the Custom Modeling library into a binary .acml library file.

Note: To achieve minimum library file size you should use the Create Library command with an empty flowsheet.
When the Create Library command is used any libraries which are attached to the current simulation are also attached to the new library. When the new library is used these attached libraries must be available and be the same version as was used when the new library was created otherwise the library will fail to load. Any libraries which are not needed should therefore be removed from the simulation before you create the library. You can do this by removing them from the Libraries statement which appears at the top of the simulation input file or from the Libraries tab on the Settings dialog. If you want to remove the default libraries set for the application e.g. Modeler.acml you can use the command line option /Nodefaultlibraries. Care should be taken not to remove libraries which are being used otherwise you will get errors on reloading the simulation.

Create a Model

To create a model:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and then click Models.
2. In the Contents pane, double-click Add Model.
3. In the Create Model dialog box, enter a name for the new model.
4. In the Text Editor window, enter the information for the new model.

Tip: For a syntax reminder, click the right mouse button in the Text Editor window, then point to Insert and click the required reminder option to insert the boilerplate text.

5. From the Build menu click Compile.

   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.

6. To quit the editor, click the Close button.

Create a Model Folder

You can create nested folders under the Custom Modeling folder in the Simulation Explorer.

To create a model folder:

1. In the All Items pane of the Simulation Explorer, click Custom Modeling.
2. In the Contents pane, double-click Add Model Folder.

Notes

- You can add new models and copy existing models to the folders you have created.
- You can create any number of levels of model folder using Add Model Folder from within each model folder.

Create a Parameter Type

To create a parameter type:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and then click Parameter Types.
2. In the Contents pane, double-click Add Parameter Type.
3. In the Create Parameter Type dialog box, enter a name and click the parameter type option.
4. In the Text Editor window, enter the information for the new parameter type.
5. From the Build menu click Compile.

   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
6. To quit the editor, click the Close button.

Create a Port Type

To create a port type:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and then click Port Types.
2. In the Contents pane, double-click Add Port Type.
3. In the Create Port Type dialog box, enter a name for the new port type.
4. In the Text Editor window, enter the information for the new port type.

Tip: For a syntax reminder, click the right mouse button in the Text Editor window, then point to Insert and click the required reminder option to insert the boilerplate text.

5 From the Build menu click Compile.

The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.

6 To quit the editor, click the Close button.

Create a Procedure

To create a procedure:

1 In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and then click Procedures.

2 In the Contents pane, double-click Add Procedure.

3 In the Create Procedure dialog box, enter a name for the new procedure.

4 In the Text Editor window, enter the information for the new procedure.

5 From the Build menu click Compile.

The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.

6 To quit the editor, click the Close button.

Create a Simulation

To create a simulation:

- From the File menu, click New.
  If you are running a simulation, a dialog box may appear prompting you to save changes or stop the current simulation. Click the appropriate response.
  The Simulation Explorer and flowsheet windows revert to their original state.

You can now create a new simulation.

Create a Stream Type

To create a stream type:

1 In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and then click Stream Types.

2 In the Contents pane, double-click Add Stream Type.

3 In the Create Stream Type dialog box, enter a name for the new stream type.

4 In the Text Editor window, enter the information for the new Stream Type.

5 From the Build menu click Compile.

The syntax is checked for errors, and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.

6 To quit the editor, click the Close button.

Create a Model Task

To define a disturbance or other sequence of actions to occur during a dynamic run, you need to create a task.

Caution: Do not change estimated or reconciled variables with a task while you are performing a dynamic estimation run; and do not change design or control variables with a task when you are performing a dynamic optimization run.

To create a task for a model:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded and, if necessary, expand the Models folder. A list of all models in the current simulation appears.
2. Click the model to which you want to add a task.
3. In the Contents pane, double-click the equal icon for the model.
4. In the Text Editor window, add the information for the new task to the model text.
5. From the Build menu, click Compile.

   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
6. To quit the editor, click the Close button.

Important note: Before the task you have created can be used in a simulation, you will need to activate the task. To activate a model task, you must first click a block (model instance).

Create a Variable Type

To create a variable type:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and then click Variable Types.
2. In the Contents pane, double-click Add Variable Type.
3. In the Create Variable Type dialog box, enter a name for the new variable type.
4. In the Text Editor window, enter the information for the new variable type.

Tip: For a syntax reminder, click the right mouse button in the Text Editor window, then point to Insert and click the required reminder option to insert the boilerplate text.

5. From the Build menu click Compile.

The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.

6 To quit the editor, click the Close button.

Create a Model Script

To automate actions in models, you need to create scripts. To create a model script:

1 In the All Items pane of the Simulation Explorer, make sure the Custom Modeling is expanded, and, if necessary, expand the Models folder.

2 Click the required model and then in the Contents pane, double-click Add Script.

3 In the Create Script dialog box, enter a name for the new script.

4 In the Text Editor window, type the information for the new script. When you have finished typing your script, you can do one of the following:

To	Do this
Run the script	Click the right mouse button in the Edit window, point to Build, and then click Invoke Script. The script is run and saved. After compilation, the Scripting dialog box appears. Click OK to dismiss this box.
Save the script	From the File menu, click Save Text, and then close the Text Editor.

Important: To run a model script, you must first click a block (model instance).

Further information: Scripts are a simple method for defining parameter and variable properties, that is, spec, value, upper and lower limits. The script language uses Microsoft® Visual Basic® Scripting Edition (VBScript). This is supplied automatically with your installation.

If you want to use the more powerful features of VBScript, you can obtain help from the Microsoft web site as a free download.

Create a Library Script

To create a script that you can store in a library for use in your simulations:

1 In the All Items pane of the Simulation Explorer, click Custom Modeling and then click Scripts.

2 In the Contents pane, double-click Add Script.

3 In the Create Script dialog box, enter a name for the new script.

4 In the Text Editor window, type the information for the new script. When you have finished typing your script, you can do one of the following:

To	Do this

Run the script	Click the right mouse button in the Edit window, point to Build, and then click Invoke Script. The script is saved then run. After running, the Scripting dialog box appears. Click OK to dismiss this box.
Save the script	From the File menu, click Save Text. When you close the Text Editor, you will be prompted to save any further changes.

When you have created the scripts, you will need to create a library to contain them. You can then invoke the scripts but you cannot change them.

Further information: Scripts are a simple method for automating activities, such as changing parameter and variable properties. The scripting language is Microsoft® Visual Basic® Scripting Edition (VBScript). This is supplied automatically with your installation.

If you want to use the more powerful features of VBScript, you can obtain full documentation on the VBScript language from the Microsoft web site.

Editing Types

Edit an Icon

To edit an icon:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and if necessary, expand the Models folder. A list of all models in the current simulation appears.
2. Click the model whose icon you want to edit.
3. In the Contents pane, double-click the icon you want to edit.
4. Make the necessary changes in the Icon Editor. You can add, delete, or modify the graphical primitives; add or remove bitmaps from the icon; and place ports on the icon. Bitmaps can be added using the Import DXF item on the file menu or existing DXF tabs in the icon editor. You can drag and drop the required bitmap from its tab once it has been imported.
5. From the File menu, click Save.
6. To quit the editor, click the Close button.

Note: If you do not save your changes, the Save Icon Confirmation dialog box appears. If you want to save your changes, click Yes. If some of the ports are not on the perimeter of the icon, the AutoPort confirmation dialog box will appear. If you wish to apply the AutoPort command, click Yes.

Edit a Model

If the model you want to edit is not in the Custom Modeling library, first copy the model to Custom Modeling.

If you want to edit a sub-model, you must also copy any library model that uses this sub-model to the Custom Modeling library, so that the changes can be used in the current simulation.

To edit a model in the Custom Modeling library:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded and, if necessary, expand the Models folder. A list of all models in the current simulation appears.
2. Click the model you want to edit.
3. In the Contents pane, double-click the equals icon for the model.
4. Make the necessary changes in the Text Editor.

Tip: For a syntax reminder, click the right mouse button in the Text Editor window, then point to Insert and click the required reminder option to insert the boilerplate text.

5. From the Build menu, click Compile.

   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
6. To quit the editor, click the Close button.

Note: If other models, for example submodels or inherited models, may be affected by the change you have made, use the Rebuild option to update all user-defined models in the Custom Modeling library.

Copy a Model to the Custom Modeling Library

To copy a model to the Custom Modeling library:

1. In the All Items pane of the Simulation Explorer, expand the library that contains the relevant model and expand the Models folder.

   A list of all the models in the library appears.
2. Click the required model and then from the Edit menu, click Copy.
3. In the All Items pane, make sure the Custom Modeling folder is expanded and then click Models.
4. From the Edit menu, click Paste.

   A copy of the model appears in the Contents pane.

Edit a Model Script

To edit a model script:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling folder is expanded, and if necessary, expand the Models folder. A list of all models in the current simulation appears.

2 Click the required model.

3 In the Contents pane, double-click the required script icon.

4 In the Text Editor window, make the necessary changes to the script information.

5 From the File menu, click Save Text.

6 To quit the editor, click the Close button.

Important note: To invoke a model script, you must first click a block (model instance).

Further information: Scripts are a simple method for defining parameter and variable properties, that is, spec, value, upper and lower limits. The script language uses Microsoft® Visual Basic® Scripting Edition (VBScript). This is supplied automatically with your installation.

If you want to use the more powerful features of VBScript, you can obtain help from the Microsoft web site as a free download.

Edit a Library Script

To edit a script that you have stored in a library:

1 In the All Items pane of the Simulation Explorer, click Custom Modeling and then click Scripts.

2 In the Contents pane, click the required script, then click the right mouse button and click Edit.

3 In the text Editor window, make the necessary changes. When you have finished typing your script, you can do one of the following:

To	Do this
Run the script	Click the right mouse button in the Edit window, point to Build, and then click Invoke Script. The script is saved then run. After running, the Scripting dialog box appears. Click OK to dismiss this box.
Save the script	From the File menu, click Save Text. When you close the Text Editor, you will be prompted to save any further changes.

When you have edited the scripts, you will need to create a library to contain them. You can then invoke the scripts but you cannot change them.

Further information: Scripts are a simple method for automating activities, such as changing parameter and variable properties. The scripting language is Microsoft® Visual Basic® Scripting Edition (VBScript). This is supplied automatically with your installation.

If you want to use the more powerful features of VBScript, you can obtain full documentation on the VBScript language from the Microsoft web site.

Edit a Parameter Type

To edit a parameter type:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and then click Parameter Types.
2. In the Contents pane, double-click the parameter type you want to edit.
3. Make the necessary changes in the Text Editor.
4. From the Build menu, click Compile.

   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
5. To quit the editor, click the Close button.

Edit Physical Properties Options

To edit physical properties options:

1. In the All Items pane of the Simulation Explorer, click Component Lists.
2. In the Contents pane, click the right mouse button on the required list and then click Physical Properties Options.
3. From the left pane, select the option that you want to change.

   The right pane shows the required options.

Note: The values for OPSET and FREE-WATER depend on what is available in your Properties Plus input file.

4. When you have selected the required options, click OK.

Edit a Port Type

To edit a port type:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded and then click Port Types.
2. In the Contents pane, double-click the port type you want to edit.
3. Make the necessary changes in the Text Editor.

Tip: For a syntax reminder, click the right mouse button in the Text Editor window, then point to Insert and click the required reminder option to insert the boilerplate text.

4. From the Build menu, click Compile.

   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
5. To quit the editor, click the Close button.

Edit a Procedure

To edit a procedure:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded and then click Procedures.
2. In the Contents pane, double-click the procedure you want to edit.
3. Make the necessary changes in the Text Editor.

Tip: For a syntax reminder, click the right mouse button in the Text Editor window, then point to Insert and click the required reminder option to insert the boilerplate text.

4. From the Build menu, click Compile.

   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
5. To quit the editor, click the Close button.

Edit a Stream Type

To edit a stream:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and if necessary, expand Stream Types.
2. Click the stream you want to edit.
3. In the Contents pane, double-click the Equal icon for the stream.
4. Make the necessary changes in the Text Editor.

Tip: For a syntax reminder, click the right mouse button in the Text Editor window, then point to Insert and click the required reminder option to insert the boilerplate text. Edit a Stream Type

5. From the Build menu, click Compile.

   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
6. To quit the editor, click the Close button.

Tip: For a syntax reminder, from the Insert menu, click the required reminder option to insert the boilerplate text.

Stream Display Properties

Display properties, i.e. color and linestyle, can be changed for a stream type and also for any individual stream on a flowsheet. Changing stream display

properties affects all existing streams of that type, except those which have been changed individually.

To change stream display properties:

1. Locate the node for the stream type in the Explorer. This is located in the Stream Types folder under the Custom Modeling node.
2. Select the Properties item on the context menu for the required stream type.
3. Select the Stream Properties tab of the Properties dialog to change the color, linestyle and to select whether or not the ID label is shown. Note, you cannot change the properties for system stream types Connection and ControlStream, or for stream types in libraries.

Tip: To change stream display properties on a flowsheet, use the Color & LineStyle item on the Stream Context menu. The dialog displayed allows you to change properties and provides the option to reset the stream back to the default for the stream type.

Edit a Model Task

To edit a model task:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and then, if necessary, expand the Models folder.
2. Click the model that contains the task you want to edit.
3. In the Contents pane, double-click the Equal icon for the model.
4. In the Text Editor window, make the necessary changes to the task information.
5. From the Build menu, click Compile.

   The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.
6. To quit the editor, click the Close button.

Important note: To invoke a model task, you must first click a block (model instance).

Edit a Variable Type

To edit a variable type:

1. In the All Items pane of the Simulation Explorer, make sure the Custom Modeling library is expanded, and then click Variable Types.
2. In the Contents pane, double-click the variable type you want to edit.
3. Make the necessary changes in the Text Editor.

Tip: For a syntax reminder, click the right mouse button in the Text Editor window, then point to Insert and click the required reminder option to insert the boilerplate text.

4 From the Build menu, click Compile.

The syntax is checked for errors and messages about the compilation are displayed in the Simulation Messages window. Changes are saved automatically after a successful compilation.

5 To quit the editor, click the Close button.

Note: If other models, for example submodels or inherited models, may be affected by the change you have made, use the Rebuild option to update all user-defined models in the Custom Modeling library.

Rename a Type

To rename a type:

1 In the All Items pane of the Simulation Explorer, click the type you want to rename, then click the right mouse button and click Rename.

2 Type the new name and then press ENTER.

Important

- The name can contain up to 27 characters, but it cannot contain spaces, or non-alphanumeric characters (for example, -\ /:*?"<>|).
- Do not create a type which has the same name as a built-in type. This can have undesirable results.
- You cannot rename a type that is being used. For example, you cannot rename a sub-model, if a model using this sub-model is currently part of the flowsheet.

Delete a Type

To delete a type:

1 In the All Items pane of the Simulation Explorer, click the type you want to delete, then click the right mouse button, and click Delete.

2 When prompted, click Yes to confirm the deletion.

Important You cannot delete either of the following:

- A type that is being used. For example, you cannot delete a sub-model, if a model using this sub-model is currently part of the flowsheet.
- The in-built (System) icon for a model

Rebuild User-Defined Types

If you change a user-defined type, other types may be affected by the change you have made. For example, a change to a model or variable type can affect

the sub-models or inherited models that use them, and can cause syntax errors.

To prevent this problem, you can use the Rebuild option to update all user-defined types in the Custom Modeling library. You can do this in either of the following ways:

In	Do this
All Items pane of the Simulation Explorer	Click Custom Modeling with the right mouse button, and then click Rebuild.
Text Editor window	Click with the right mouse button, point to Build and then click Rebuild.

All user-defined types in the Custom Modeling library are updated.

Structures

How to Use Structures

To define and use a structure, carry out the steps detailed in the following topics:

- Define a structure type
- Create an Instance of a Structure Type
- Reference a Structure Type from a Model
- Creating a Block or Stream with a Structure Reference

Define a Structure Type

This can be carried out using the Add Structure icon in the explorer that is contained in the Structure Types folder under Custom Modeling or any user defined folder. Once you give a name, the text editor will be shown allowing you to enter the content of the structure type.

You can add any of the following:

- Variable or Parameter Type definitions
- Parameter declarations
- Variable declarations
- Set declarations
- Assignments to variable or parameter properties
- Declarations of Structure instances.
- Equations
- Procedure calls
- Comments

You can add scripts or forms to a structure type in the same way you add them to models.

Create an Instance of a Structure Type

You can create an instance of a structure type by either using the Create Instance menu item on the explorer context menu for a structure type (also accessible using the quick key ctrl I) or by dragging and dropping a structure type onto the flowsheet icon in the explorer.

Note: You cannot drag and drop them onto the flowsheet window.

The system prompts you for a name. Instances of structure types are placed in folders under the flowsheet node in the Explorer. If the structure type used is in the Structure Types folder, the instances are placed in a Structures folder. If the Structure Type is in a user defined folder the instance is placed in a folder with the same name but under the flowsheet.

You can access forms and scripts by navigating to the structure folder under the flowsheet and highlighting the required structure instance in the Explorer tree view.

Reference a Structure Type from a Model

To use a structure in a model you construct a reference to the structure type

e.g. Rxn1 as External Powerlaw;

where Powerlaw is the name of a structure type and Rxn1 is the reference you will be using in this model. Having defined this reference you can use it to access parameters and variables in the structure type.

e.g. Rxn1.HeatofReaction

This could then be used wherever local parameters or variables are used. However note you cannot assign a value to a parameter or variable in a structure from a reference. You can only do that in the structure type itself.

Creating a Block or Stream with a Structure Reference

When you create a block or stream on the flowsheet that has a reference to a structure type you need to specify which structure instance will be referenced. Until you do this, the system will be unable to find the parameters or variables in the structure you referenced.

You can specify the structure to be used from the AllVariables table for the block or stream. This will have an entry with the same name as the reference you gave, Rxn1 in the above example.

The value field has a drop-down list containing the names of all the structure instances of the type you used in the reference definition statement. In the above example this was Powerlaw. You can select any one and the references will be completed.

You can also set up a default for structure instances in the model

e.g. Rxn1 as External Powerlaw("Reactions.Reaction1");

or

Rxn1 : ("Reactions.Reaction1");

Choosing Solver Options

Access Solver Options

To change solver options:

1. In the All Items pane of the Simulation Explorer, click Simulation.
2. In the Contents pane, double-click Solver Options.

   The Solver Properties dialog box appears.
3. Click the required tab, select the options you want and then click OK.

Tip: You can also access solver options by clicking Solver Options from the Run menu.

Change Tolerances for Non-Linear Solvers

To change tolerances for a non-linear solver:

1. In the All Items pane of the Simulation Explorer, click Simulation.
2. In the Contents pane, double-click Solver Options.

   The Solver Properties dialog box appears.
3. Click the Non-Linear Solver tab and in the Mode box, select Tolerances.
4. Type the information for the options you want to change and then click OK.

Define Tolerances for a Simulation

To define tolerances for a simulation:

1. In the All Items pane of the Simulation Explorer, click Simulation.
2. In the Contents pane, double-click Solver Options.

   The Solver Properties dialog box appears.
3. Click the Tolerances tab.
4. Type the information for the options you want to change and then click OK.

Specify an Integrator for a Simulation

To choose an integrator for a simulation and specify the related integrator options:

1. In the All Items pane of the Simulation Explorer, click Simulation.
2. In the Contents pane, double-click Solver Options.

   The Solver Properties dialog box appears.
3. Click the Integrator tab.
4. Type or select the options you want to change and then click OK.

Specify a Linear Solver

To choose a linear solver and specify the related solver options:

1. In the All Items pane of the Simulation Explorer, click Simulation.
2. In the Contents pane, double-click Solver Options.

   The Solver Properties dialog box appears.
3. Click the Linear Solver tab.
4. Type or select the options you want to change and then click OK.

Specify a Non-Linear Solver

To choose a non-linear solver and specify the related solver options:

1. In the All Items pane of the Simulation Explorer, click Simulation.
2. In the Contents pane, double-click Solver Options.

   The Solver Properties dialog box appears.
3. Click the Non-Linear Solver tab and in the Mode box, select General.
4. Type or select the options you want to change and then click OK.

Specify Procedure Tearing Options

To specify the options for procedure tearing:

1. In the All Items pane of the Simulation Explorer, click Simulation.
2. In the Contents pane, double-click Solver Options.

   The Solver Properties dialog box appears.
3. Click the General tab and in the Mode box, select Tearing.
4. Select the options you want to change and then click OK.

Troubleshoot Non-Linear Equations

To diagnose why your non-linear solver equations fail to solve:

1. In the All Items pane of the Simulation Explorer, click Simulation.

2 In the Contents pane, double-click Solver Options.

The Solver Properties dialog box appears.

3 Click the Non-Linear Solver tab.

4 Use the three diagnostic options in the Diagnostics pane to diagnose convergence of your equations. Click on the button (top) for a description of these diagnostics.

Specify an Optimizer

To choose an optimizer and specify the related solver options:

1 In the All Items pane of the Simulation Explorer, click Simulation.

2 In the Contents pane, double-click Solver Options.

3 In the Solver Properties dialog box, click the Optimizer tab.

4 Type or select the options you want to change, and then click OK.

Specify Homotopy Options

To specify options for steps in a Homotopy run:

1 In the All Items pane of the Simulation Explorer, click Simulation.

2 In the Contents pane, double-click Solver Options.

3 In the Solver Properties dialog box, click the Homotopy tab.

4 Type the values for the options you want to change, and then click OK.

Important: To use the options you have specified, you must have enabled homotopy and specified which fixed variables to use.

Specify an Open Non-Linear Solver

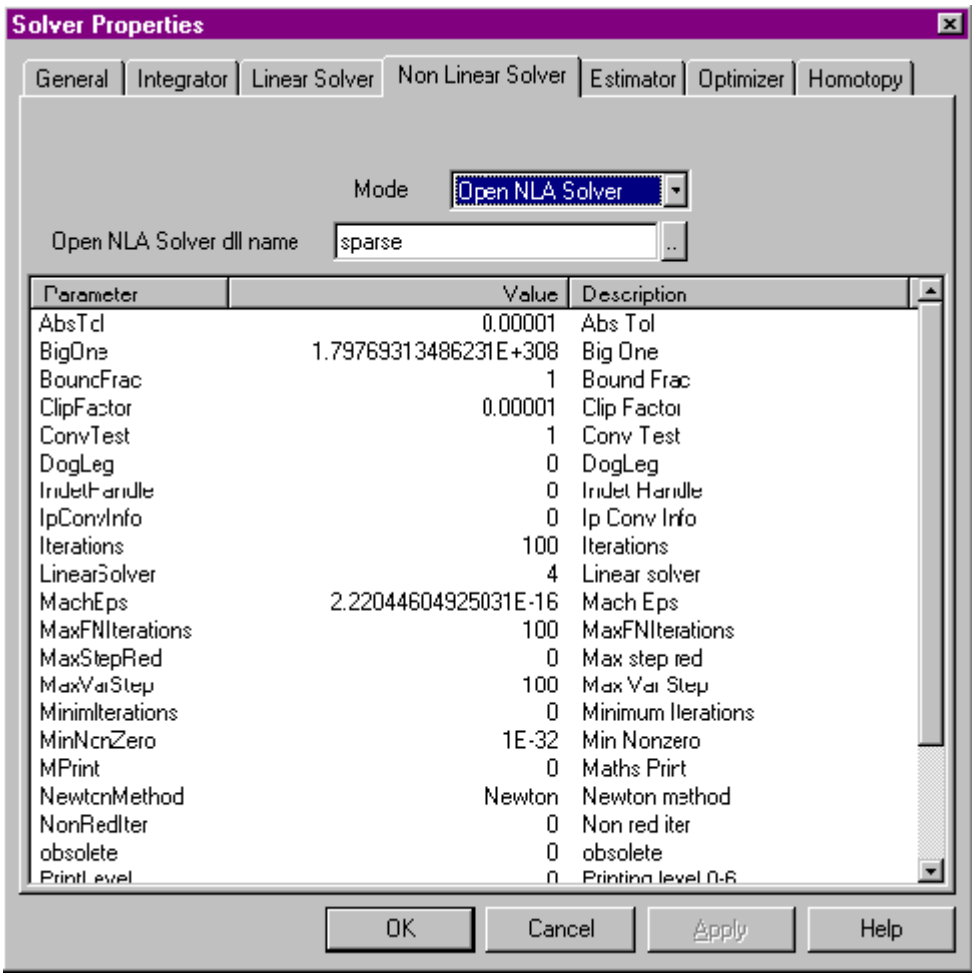
To specify an open non-linear solver:

1 From the Run menu click Solver options, or in the Contents pane of the Simulation Explorer, double-click Solver Options.

2 Click the Non-Linear Solver tab. In the Mode list select Open NLA Solver.

3 To specify the open solver DLL file, type the path and file name of the DLL or click the browse button  to locate and select the required DLL file.

Note: We recommend that the open solver DLL file is placed in the \bin directory. An example of the parameters and default settings corresponding to a selected DLL is shown below:



4 To edit a parameter value, click the parameter name in the left column and enter the new value when prompted. To apply the changes to the solver, click the Apply button.

To revert to the internal non-linear solver, select General, Diagnostics, or Tolerances in the Mode list.

Note: You can specify an open solver and change the parameters through automation. If the open solver or parameters change through automation then those changes will be automatically reflected in this dialog box.

Using Remote Servers

Accessing Server Configurations

To view or change the existing server configuration:

- From the Tools menu, click Server Configuration.

  The Server Configurations dialog box appears, from which you can add or change servers, remove servers, and edit existing server configurations.

For information on setting up remote servers, see Defining a Remote Server.

Defining a Remote Server

To specify a remote simulation engine host from a client PC:

1 In the Server Configurations dialog box, click Add.

The Servers dialog box appears.

2 In the Current Configuration Name box, type a name.

This can be the name of the computer where the simulation server software will run, or it can be a name with meaning to you only.

3 Click Add Host

The Host Properties dialog box is displayed.

4 In Host Name box, type the name of the remote computer where the simulation server software is installed.

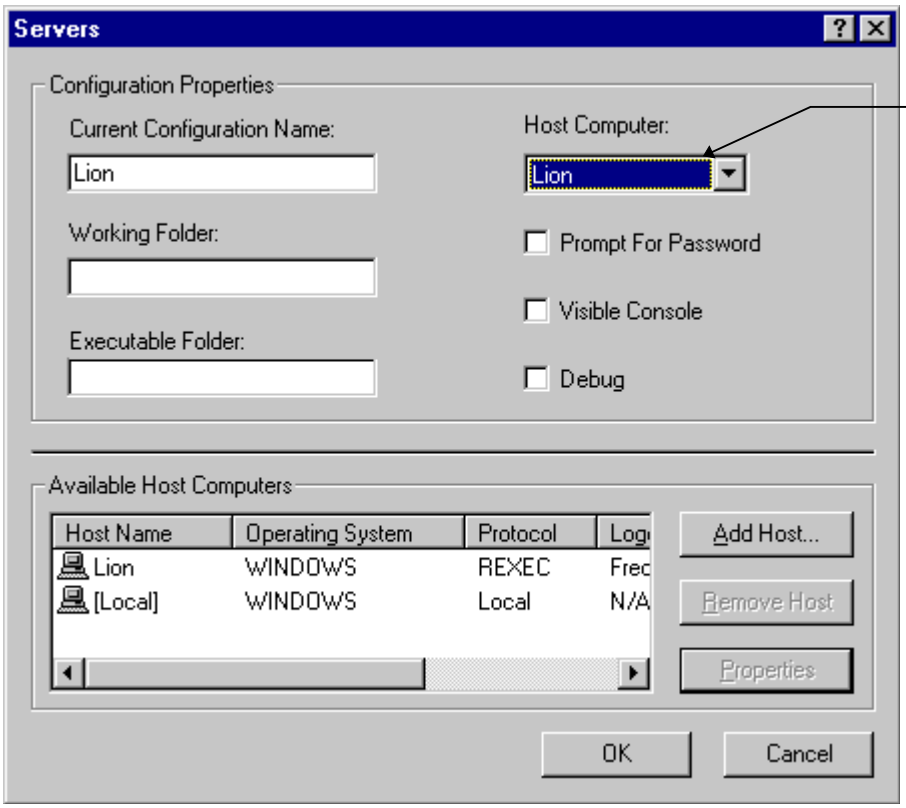
Note: Take care with the accuracy of both case and spelling of the name. The name is used to resolve the network address of the computer, and must be in a form recognizable to your Domain Name Server. If you do not understand the term Domain Name Server, seek advice from your system administrator.

5 In the User Name box, type the user name of the user under whom the simulation engine will run on the remote computer. You must supply a value for this field.

6 In the Password box, type the password for the user and click OK.

Note: If you prefer not to have the password stored by the system, you may leave this field blank.

7 In the Servers dialog box, select the name of the host you just added in the Host Computer list:



8 In the Working Folder text box, type the path of the root folder on the remote machine where run time data (that is, simulation snapshots and plot data) will be stored, for example, **C:\Program Files\AspenTech\Working Folder\Aspen Custom Modeler 121\UserName**.

Note: Your snapshots and plot data will appear in sub-folders below this working folder on the **remote** machine.

9 In the Executable Folder text box, type the path of the folder where the simulator executables have been installed on the remote machine, for example **C:\Program Files\AspenTech\AMSystem 121\bin**.

10 Select any or none of the optional check boxes, as appropriate:

If	Select
You want the system to prompt for the remote password rather than use the stored one which has been associated with the host	Prompt For Password
You want the console windows of the servers to be visible and you are running them on the local host	Visible Console

11 Click OK to accept your changes and close the Servers dialog box.

12 In the Server Configurations dialog box, click OK.

You are notified that you will have to restart your AspenTech product for the changes you have made to take effect.

After you have restarted, you can run simulations from your PC on the remote simulation server you specified.

2 Five Tanks Example

This example models five open tanks in series. The key assumptions are:

- Flow out of each tank is proportional to the square root of the level of fluid in the tank
- Fluid at constant density
- Tank geometry is a regular sided vertical cylinder
- No overflow condition

Running the Five Tanks Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder:
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\Fivetank

To run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\FiveTank

To run the simulation:

1. From the File menu, click Open.
2. Open the FiveTank folder. If you have copied the files to the example working folder, this is:
   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121
3. Double-click the file **fivetank.acmf**.

You can run the simulation dynamically and change the feed flow rate to the first tank using an AllVariables table. To do this:

- In the Flowsheet window, double-click the stream FeedStream that enters Tank1. Enter a new value for the variable displayed as **Flow**.

You can see the simulation respond to your step change by displaying a plot. To do this:

1. In the All Items pane of the Simulation Explorer, click Flowsheet.
2. In the Contents pane, double-click the plot **TankVolumes** to display it.

3 Double-click the tank icons on the flowsheet. This gives a more detailed plot of key tank data.

4 Click Run and watch the simulation respond to your step change.

Advanced Features in Five Tanks Example

The advanced features in this example are:

- Using Microsoft Excel
- Running from Microsoft Visual Basic
- Tasks
- Flowsheet level Profile Plots

Using Microsoft Excel

While Aspen Custom Modeler is running a simulation, you can use Microsoft Excel to display a real-time profile plot of the tank levels.

1 From Excel, open FiveTank.xls.

2 If prompted, select the options to enable macros and to update linked information.

3 Select the Tank Level Plot sheet, and click the Run button to run the simulation.

Capturing Real Time Data using Excel

You can use Visual Basic within Excel to capture simulation results to a table.

To run the example, load the FiveTank.acmf file and the FiveTank.xls Excel file.

1 In the FiveTank.xls Excel file, click the worksheet called Import Data.

2 You can change the default values of the fields.

3 By default, the volume of fluid in Tank5 is imported from the file FiveTank.acmf. The data is read into the spreadsheet every two seconds of real time.

4 In the Excel spreadsheet, click the Run button to start the simulation running and to start the data reading into the spreadsheet.

   You can see time / value pairs appearing in a column in the spreadsheet.

When you have enough data, click the Pause button.

You can now use the time / value data to define a plot, or to use in further spreadsheet calculations.

Applying Step Changes at Specified Times Using Excel

You can use Visual Basic in Excel to apply a number of step changes in fixed variables at specific times. To do this, load the FiveTank.acmf file and the FiveTank.xls Excel file.

1. Make sure you have run the FiveTank example.
2. In the FiveTank.xls Excel file, click the worksheet called Schedule.
3. The worksheet Schedule contains a number of time / value pairs representing the desired value of feed flowrate at a specific time.
4. You can alter the values of the feed flow rate applied at specific times by editing the values in the columns. You can also add to the number of time / value pairs.
5. In the Excel spreadsheet, click the Run Schedule button to run the simulation and apply the step changes at the specific times.

Running from Microsoft Visual Basic

You can control the simulation from Microsoft Visual Basic. This enables you to easily make changes to the key input data values and get results back into a single dialog box.

To load the files into Visual Basic:

- Double-click FiveTank.vbp

To run the control application:

- In Visual Basic, click the Start button.
  - Use the spin box to alter the feed flow rate to Tank1.
  - Tank volumes are reported in the control panel. The color of the text labels for the tank volume indicate low alarm and high alarm conditions.

Tasks

You can run a task that is supplied with the example. This task covers the main task features available in Aspen Custom Modeler.

To do this, you need to activate the task:

1. At the start of a dynamic run, in the All Items pane of the Simulation Explorer, click Flowsheet.
2. In the Contents pane, double-click the task named **TestTask** to activate it.

Now when you run a dynamic simulation, the instructions in the task are carried out. These instructions show a number of different disturbances you can make to a dynamic run.

You can look at the results of the simulation through predefined plots.

To display the TankVolumes plot :

1. In the All Items pane of the Simulation Explorer, click Flowsheet.
2. In the Contents pane, double-click the plot **TankVolumes** to display it.

3 Click Run and watch the simulation respond to your step change.

Each tank on the flowsheet contains a plot named TankPlot.

Flowsheet Level Profile Plot

The example contains a flowsheet level profile plot called HeightsProfile.

First activate the task **TestTask**.

1 At the start of a dynamic run, in the All Items pane of the Simulation Explorer, click Flowsheet.

2 In the Contents pane, double-click the task named **TestTask** to activate it.

To display the HeightsProfile plot:

1 In the All Items pane of the Simulation Explorer, click Flowsheet.

2 In the Contents pane, double-click the plot HeightsProfile to display it.

3 Click Run and watch the simulation respond to your step change.

4 Right-click the plot and select Profile Variables... to open the Profile Editor and see how the plot has been configured. Click the 3d button to turn this plot into a 3d plot, then click OK to apply your change.

3 Absorber Example

This example is for users of Aspen Custom Modeler with Properties Plus. You therefore have to have Aspen Plus installed to use this example.

Note: If you are using a Simulation Engine installation from a client machine, make sure that you run Aspen Plus using the same server and the same working folder as you have configured for Aspen Custom Modeler.

This example simulation is from *Equilibrium-Stage Separation Operations in Chemical Engineering*, by Ernest J. Henley and J.D. Seader, Example 12.8, page 466. A simple, 6-stage-absorber is used to remove heavy components from a light gas stream with a high-molecular weight oil.

There is a pressure controller and a level controller on the column.

This example illustrates these areas of Aspen Custom Modeler:

- How to use models within models, or hierarchy
- How to use scripts to help initialize models in steady-state
- How to write equations which are automatically normalized
- How to use local or simplified physical properties
- Model polymorphism based upon the structure of the flowsheet
- Model polymorphism based upon parameters
- How to start a dynamic simulation

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\Absorber

To run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\Absorber

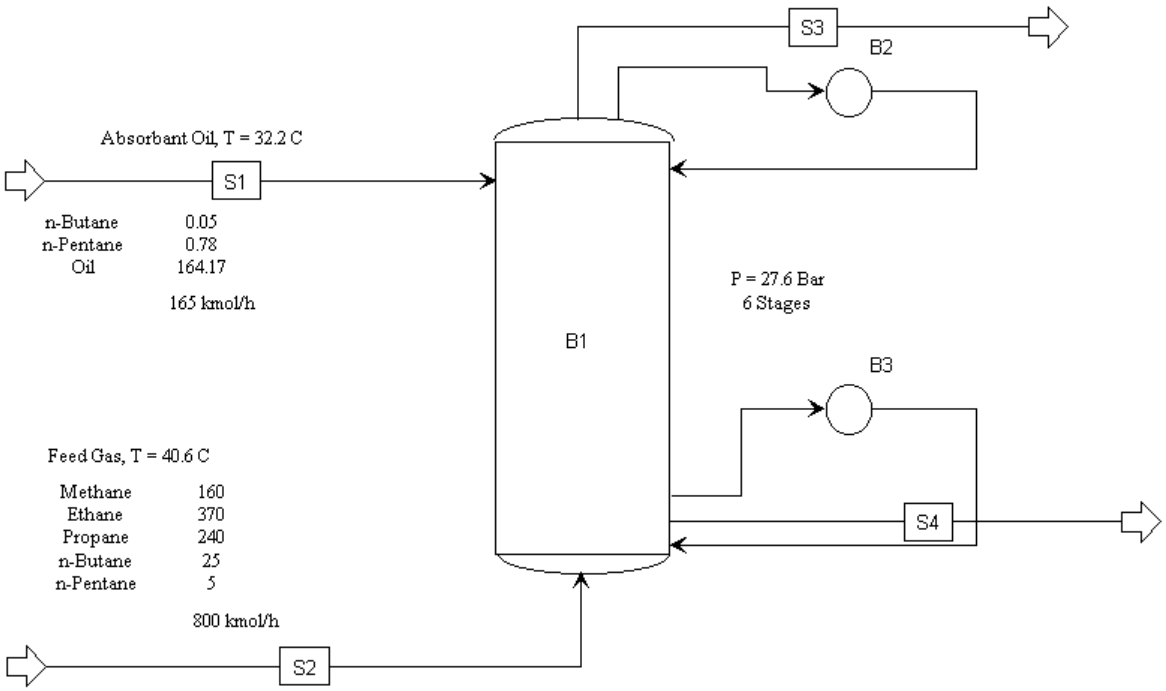
Models Used in the Absorber Example

There are several models in the simulation:

Model Name	Used to
antoine_model	Calculate coefficients for a simple ideal model for equilibrium k values based upon a simplified Antoine's equation
column	Model a complete absorption column
enthalpy_model	Calculate coefficients for a simple component enthalpy model for the vapor and liquid phases
props	Calculate physical properties based upon the simplified equations or rigorous physical property calls from Properties Plus
tray	Model a single stage
tray_hydraulics	Calculate the holdup and flow rates for the vapor and liquid phases for each tray

The tray model is the heart of the simulation, and includes material and energy balance equations for an equilibrium stage model. The equations are written so that they are self-normalized by the flows in the column. This helps speed the solution of the problem. The column model organizes each of the models into the absorber simulation.

Note the structure of each of the models. For example, the tray model and column model include IF structures around logical ISCONNECTED for each of the ports. This illustrates how the model can convert its equations to new equations depending upon the flowsheet connectivity, and is an example of model polymorphism. In addition, a parameter called proptype is used to switch the models between rigorous and simple property models, and is another example of polymorphism.



Setting Up Properties Plus for the Absorber Example

You need to set up Properties Plus for the simulation. You only need to do this once:

1 Locate the example folder Absorber. If you copied the files to the example working folder, this is:

C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\Absorber

2 Load the file Absorber.bkp in to Aspen Plus and run it.

Note: This is a properties only input file, so you will not see a flowsheet in Aspen Plus.

3 From the File menu, click Save As and save the simulation as an Aspen Plus document (.apw file), then exit Aspen Plus.

Now you can run the example.

Running the Absorber Example

When you have set up Properties Plus, you can run the Absorber example. To do this:

1 In Aspen Custom Modeler, from the File menu, click Open.

2 Open the Absorber folder. If you copied the files to the example working folder, this is

C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\Absorber

3 Click the example file **Absorber.acmf** then click Open.

4 In the All Items pane of the Simulation Explorer, click Flowsheet to select it.

5 In the Contents pane, double-click the task named FeedDisturbance to activate it.

6 In the Contents pane, double-click the script InitializeSimulation to run it. The InitializeSimulation script:

- Fixes the feed compositions, flows, temperatures, and pressures in each of the feed streams. Notice also that we use a stream model.
- Initializes the temperature, liquid and vapor flows in the absorber. The specabsorber script fixes the pressure and temperature in the absorber, and frees the duty for each stage.
- Runs the simulation. A steady-state run at this point is at a fixed temperature of 50°C. This simulation converges very quickly, and shows the power of using simplified properties and initialization.
- Switches the simulation to rigorous properties and an adiabatic duty for each tray.
- Runs a steady-state simulation again.
- Switches the specifications of the absorber to dynamic specifications, where tray coefficients are used to calculate the pressure, and the vapor and liquid product flows are fixed.

Now you can add controllers.

Adding Controllers to the Absorber Example

When you have set up Properties Plus, and run the Absorber example, follow these steps to add controllers:

1 From the ControlModels folder of the Modeler library, place a PID controller on the flowsheet. This will be a pressure controller to maintain the vapor flow based on the pressure in the top stage of the absorber.

2 Use the built-in stream ControlSignal to connect these variables:

Connect from	To
B1.p(1)	PID variable PV
PID variable OP	B1.vapor_out

3 Place another PID controller on the flowsheet. This will be a **level controller** to manipulate the liquid flow based on the bottom stage liquid

level. Use ControlSignal to connect the new PID controller:

Connect from	To
B1.Stage(6).Sieve.Liquid_Height	PID variable PV
PID variable OP	B1.liquid_out

4 Double-click each controller to access the controller face-plate, and then click the Configure button  . For each controller, click the Initialize Values button to set up default values for the controller.

5 Enter the following tuning parameters:

Tuning Parameter	Pressure Controller	Level Controller
Gain	5 %/%	5 %/%
Integral Time	5 min	10000 min
Controller Action	Direct	Direct

5 When you have added the controllers, run the simulation in steady-state mode.

6 Change the run mode to Dynamic and start the run.

7 View changes in the column temperature profile using the Flowsheet plot TempProfile.

4 Double Effect Evaporator Optimization Example

This example models a two-stage evaporator system.

A glycol/water solution enters the first vessel and is heated by steam. The glycol is concentrated and the vapor offtake from the first vessel is used to heat the second evaporation stage running at a lower pressure.

The Objective expression to be maximized is defined in the Flowsheet Constraint definition.

The decision variables are given a specification Vary. For this simulation, the decision variables are:

- Outlet Pressure for the Vapor stream
- Valve position for the process feed stream, effectively the process feed flow rate
- Valve position for the steam feed stream, effectively the steam feed flow rate

The following constraints are applied to the simulation in the Flowsheet Constraints definition:

- Minimum concentration in Evaporator 2 is 0.12 kg/kg
- Minimum pressure in Evaporator 2 is 30.0 bar
- Maximum temperature in both evaporators is 100.0 Celsius

Running the Double Effect Evaporator Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\EvaporatorOpt

To prepare to run this example:

- Copy the files in the example directory to a convenient working folder, for example:

C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\EvaporatorOpt

To run this example:

1. Open the file DoubleEffectEvap. If you copied the files to the example working folder, these files are located in:

   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\EvaporatorOpt

2. Run the simulation in steady-state mode.
3. Change the mode to Optimization, and run the simulation again.

5 Heated Metal Cube (PDE) Example

This simulation models a cube of a material heated across three surfaces.

The heat flux through the material slab is modeled in three dimensions. The heat flux equation is a second order, partial derivative. Partial derivatives are used in three spatial dimensions to model the cube of material.

Tasks are provided that simulate raising the temperature of three of the surfaces of the cube and raising the temperature of three of the edges of the cube. You can combine these heating effects to experiment with different dynamic temperature profiles in the cube.

A Microsoft Excel spreadsheet file is provided, which shows temperature contours at four slices in the cube.

Running the Heated Cube Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\HeatedCube

To run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\HeatedCube

To run the example, follow these steps:

1. From the File menu, click Open.
2. Open the HeatedCube folder. If you copied the files to the example working folder, these files are located in:

   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\HeatedCube

- Double-click the example file **HeatedCube.acmf**.

- In the All Items pane of the Simulation Explorer, click Flowsheet.
- In the Contents pane, you can activate a combination of the tasks WarmEdge1, 2, and 3, and WarmSurface1, 2, and 3. You may wish to edit these tasks to alter the start times, so that the heating effects start at different times.

Note: After you have edited a task, you must compile and then activate it. To do this, click the task with the right mouse button and then click Compile. After compiling a task, activate it by clicking it with the right mouse button and then clicking Activate.

3 Open the Microsoft Excel spreadsheet file HeatedCube.xls to display the temperatures inside the cube. If a dialog box appears asking if you want to update your workbook, or your macros, click YES.

Note You need Excel 7.0 or later to use this file.

4 In Aspen Custom Modeler, click Run to start the dynamic run then in Excel, watch the plots change as the surface temperatures increase. You may need to pause the simulation to allow Excel time to update the surface plot.

You can also click it with the right mouse button and then click Activate.

6 Heated Metal Slab (PDE) Example

This simulation models a slab of a material heated and cooled on opposite sides. For example, this simulation could represent the heat flow through the wall of a vessel, heated or cooled with a jacket.

The heat flux through the material slab is modeled in two dimensions. The heat flux equation is a second order, partial derivative. Partial derivatives are used in two spatial dimensions to model the slab of material.

Two tasks are used to model transient effects during dynamic simulation. The tasks model a pulse of warm fluid passing over two of the edges of the slab. These pulses start at the same time, but the period is different, so the pulses become out of phase as the simulation proceeds.

A Microsoft Excel spreadsheet file is provided to give a surface plot of the temperature in the two spatial directions modeled.

Running the Heated Slab Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, the files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\HeatedSlab

To run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\HeatedSlab

To run the example:

1. From the File menu, click Open.
2. Open the HeatedSlab folder. If you copied the files to the example working folder, these files are located in:

   C:\Program Files\AspenTech\Working Folders\Aspen

Custom Modeler 12.1\HeatedSlab

3 Double-click the file **HeatedSlab.acmf**.

4 In the All Items pane of the Simulation Explorer, click Flowsheet.

5 In the Contents pane, click Flowsheet and activate the task Side2

6 In the mode box on the Run Control toolbar, change the run mode to Dynamic.

7 Before running the simulation, open the Microsoft Excel spreadsheet file HeatedSlab.xls to display the temperatures inside the slab. If a dialog box appears asking if you want to update your workbook, or your macros, click YES.

Note You need Excel 95 or later to use this spreadsheet file.

8 Start the dynamic run and watch the Excel surface plot change as the temperatures change on two opposite edges of the slab. You may need to pause the simulation to allow Excel time to update the surface plot.

Viewing a Plot of the Corner Temperatures

You can view a plot in Aspen Custom Modeler of the corner temperatures of the heated slab. To view this plot:

1 In the All Items pane of the Simulation Explorer click Flowsheet.

2 In the Contents pane, double-click the plot called CornerTemperatures.

Viewing a Plot of Temperature Throughout the Slab

In Aspen Custom Modeler you can view a 3d profile plot of the temperature throughout the heated slab.

To view this plot:

1 In the All Items pane of the Simulation Explorer click Flowsheet.

2 In the Contents pane, double-click the plot called TemperatureProfile.

3 Right-click the plot and select Profile Variables... to see how the various axes have been configured.

7 Reactor Dynamic Estimation Example

This example models a simple reactor. There are three components in the system and two main reactions.

The components are represented by A, Q, and S.

The reactions are:

K3

$A \rightarrow S$ (1)

K1 K2

$A \rightarrow Q \rightarrow S$ (2)

Where both reactions are rate-based.

The estimation simulation calculated estimated values of K1, K2, and K3 from experimental measurements of the concentrations of A and Q.

Running the Reactor Dynamic Estimation Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, the files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\ReactorDynEst

To run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\ReactorDynEst

There are three ways to run the estimation example:

- Using the Estimation dialog box
- Using the supplied Microsoft Visual Basic script

- From external Microsoft Visual Basic in the supplied Microsoft Excel spreadsheet

Using the Estimation Dialog Box

To run the example using the Estimation dialog box:

1. From the File menu, click Open.
2. Open the file **ReactorDynamicEstSetupComplete.acmf**. If you copied the files to the example working folder, this file is located in:

   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\ReactorDynEst
3. To inspect the setup of the estimation experiments, from the Tools menu, click Estimation.
4. Click the Estimated Variables tab and note that it shows those variables which are to be estimated.
5. On the Dynamic Experiments tab select the dynamic experiment listed on it and click the Edit button to view its details.
6. In the dialog box, note the following three tabs:

   Measured Variables tab. Lists variables for which there is observed data – in this case, the reaction rate constants.

   Fixed Variables tab. Lists variables which have prescribed values in the experiment. There are no fixed variables in this example.

   Initial Variables tab. Lists those variables which are given an initial value in the experiment and enables you to modify the initial value. There are no initial variables in this example.
7. Change the run mode to Estimation and run the simulation. Wait for the solution to complete.
8. To observe the results on the Estimated Variables tab, from the Tools menu, click Estimation.
9. To check the accuracy of the plot, on the Measured Variables tab of the Estimation dialog box, double-click the measured variables to edit them, and observe the deviations. You may also plot these results here.

From a Microsoft Visual Basic Script

To run the example from a Visual Basic script:

1. From the File menu, click Open.
2. Open the ReactorDynEst folder. If you copied the files to the example working folder, these files are located in:

   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\ReactorDynEst
3. Open the file ReactorDynamicEst.acmf
4. Perform an initialization run.
5. In the All Items pane of the Simulation Explorer, click Flowsheet to select it.

6 In the Contents pane, double-click the script called **est_dyn_cat**.

7 The estimation simulation starts.

Results of the simulation are displayed in the Simulation Messages window.

From External Visual Basic in an Excel Spreadsheet

To run the example from external Microsoft Visual Basic in a Microsoft Excel spreadsheet:

1 From the File menu, click Open.

2 Open the folder ReactorDynEst. If you copied the files to the example working folder, these files are located in:

C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\ReactorDynEst

3 Open the file ReactorDynamicEst.acmf.

4 Perform an initialization run.

5 If you have Excel 97 or later, open the spreadsheet ReactorDynamicEst.xls.

– or –

If you have Excel 7.0, open the spreadsheet **ReactorDynamicEst95.xls**. This is in Examples\Off95ExcelSheets.

Note: If a dialog box appears asking if you want to enable macros, click Enable Macros.

6 On the Results sheet, click the button Run Estimation on the spreadsheet.

7 After the simulation completes, the results are put into the Excel spreadsheet.

The Excel spreadsheet has some additional features:

You can	To do this
Analyze the results of the Estimation run	Select the Analysis worksheet. You can check the results of the dynamic simulation with the estimated values against the original experimental results.
Check the sensitivities of the estimated variables to the measured variable	Select the Sensitivity worksheet.
Check and alter the Estimation Solver Options	Select the Estimation Solver Options worksheet. Click Get to read in the current values of the estimation solver options. Click Put to apply values.

EUROKIN Test Problems

EUROKIN F1 is a European co-operation initiative on kinetics research. Further details of EUROKIN can be obtained at the web site at http://www.dct.tudelft.nl/eurokin/.

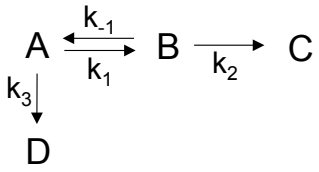
EUROKIN has developed four test problems for software packages capable of fitting kinetic parameters to experimental reaction data. AspenTech has created solutions for each of these test problems, and these solutions are delivered with Aspen Custom Modeler.

- EUROKIN Test Problem 1 – Parameter estimation problem with batch data and CSTR data
- EUROKIN Test Problem 2 – Fixed bed with transport phenomena
- EUROKIN Test Problem 3 – Methanol synthesis models
- EUROKIN Test Problem 4 – Tubular reactor startup

EUROKIN Test Problem 1

This problem is to fit kinetic parameters to experimental data from two different reactor types, a batch reactor and a stirred tank reactor operated at steady state.

The reaction scheme is as follows:



All reactions occur only on the catalyst surface. The reaction A → B is a reversible reaction, reactions B → C and A → D are irreversible reactions.

The kinetics of the catalytic reactions can be described using the following Langmuir-Hinshelwood Hougen-Watson type equations.

$$r_1' = \frac{k_1 K_A \left([A] - \frac{[B]}{K_{eq1}} \right)}{1 + K_A[A] + K_B[B]} \qquad r_2 = \frac{k_2 K_B [B]}{1 + K_A[A] + K_B[B]} \qquad r_3 = \frac{k_3 K_A [A]}{1 + K_A[A] + K_B[B]}$$

The temperature dependency of k_1, k_2 and k_3 are described using the Arrhenius equation:

$$k_1 = k_1^0 \exp\left(\frac{-E_1}{R\,T}\right) \qquad k_1 = k_2^0 \exp\left(\frac{-E_2}{R\,T}\right) \qquad k_1 = k_3^0 \exp\left(\frac{-E_3}{R\,T}\right)$$

K_A, K_B and K_{eq1} are defined functions of temperature. k_1, k_2, k_3, E_1, E_2, and E_3 are the reaction parameters to be estimated.

Concentration versus time data is available from 4 isothermal batch reaction experiments. Also outlet concentration data is available for 10 stirred tank reactor experiments.

This example uses all of this data to estimate the reaction parameters.

- Running EUROKIN Test Problem 1

Running EUROKIN Test Problem 1

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder:

C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\EUROKIN

To run this example:

1 Copy the file EUROKIN1acmf from the examples folder to a convenient working folder.

2 Start Aspen Custom Modeler, and from the File menu, click Open.

3 Select the file EUROKIN1acmf.

4 Click the run button to run the simulation.

EUROKIN Test Problem 2

This problem is to fit reaction constants for a catalytic reaction with mass transfer limitations, using experimental results from a tubular reactor.

The reactions scheme is:

A + B -> C

A + C -> D

All reactions take place in the liquid phase at the outer surface of spherical catalyst particles.

The kinetics are described by:

Reaction	Equation
Reaction of A with B to C	$R_1 = \frac{k_1 K_{1m} K_{2m} [A]_s [B]_s}{\left(1+\sqrt{K_{1m}[A]_s}+K_{2m}[B]_s+K_{3m}[C]\right)^3}$
Reaction of A with C to D	$R_2 = \frac{k_2 K_{1m} K_{3m} [A]_s [C]}{\left(1+\sqrt{K_{1m}[A]_s}+K_{2m}[B]_s+K_{3m}[C]\right)^3}$

The subscripts *s* for [*A*] and [*B*] denote concentrations at the surface of the catalyst. Mass transfer limitations in the liquid to solid film are an important factor in the overall process rate, so the surface concentrations will differ from those in the bulk of the liquid.

Other equations describing the reactions and the reactor are:

Mass Balance/Transfer	Equation

Mass balance for B	$\frac{d[B]}{dx} = -1442([B]-[B]_s)$
Mass balance for C	$\frac{d[C]}{dx} = 28.8(R_1-R_2)$
Mass transfer of A	$9.88([A]-[A]_s) = (R_1+R_2)$
Mass transfer of B	$50.2([B]-[B]_s) = R_1$

12 experiments were performed in a tubular reactor. The inlet concentrations and length of the reactor were varied across the experiments. The outlet concentrations were measured.

The kinetic rate constants, k_1, k_2, K_{1m}, and K_{2m} are fitted to the experimental data.

- Running EUROKIN Test Problem 2

Running EUROKIN Test Problem 2

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder:

C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\EUROKIN

To run this example:

1. Copy the file EUROKIN2acmf from the examples folder to a convenient working folder.
2. Start Aspen Custom Modeler, and from the File menu, click Open.
3. Select the file EUROKIN2acmf.
4. Click the run button to run the simulation.

EUROKIN Test Problem 3

The problem compares 20 different possible kinetic models for describing the reaction of carbon monoxide and hydrogen to produce methanol:

$$CO + 2H_2 \leftrightarrow CH_3OH$$

Both the experimental data and the proposed rate equations were taken from the literature [Berty, J.M.; *The State of Kinetic Model Development*; Chem. Eng. Progr., pp. 61-67 (Feb. 1988)].

Reaction rate values are available from 27 experiments in a continuous stirred tank reactor. The temperature and partial pressure of each reactant is varied between these experiments.

The kinetic constants in each of the 20 models are fitted to this data. A script is used to automatically run all 20 estimations, and report the results in an Excel spreadsheet.

- Running EUROKIN Test Problem 3

Running EUROKIN Test Problem 3

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder:

C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\EUROKIN

This example makes use of Microsoft Excel to produce tabulated results. You need Excel 97 or later installed to run this example.

To run this example:

1. Copy the file EUROKIN3acmf from the examples folder to a convenient working folder.
2. Start Aspen Custom Modeler, and from the File menu, click Open.
3. Select the file EUROKIN3acmf.
4. In Explorer Select Flowsheet, and then double click the script RunEstimation to run the simulation.
5. The script will automatically perform 20 estimation runs, each using a different kinetic model. It will automatically open an Excel spreadsheet and write the results of each run to this spreadsheet.

EUROKIN Test Problem 4

This problem estimates reaction kinetics, heat transfer and mass transfer parameters for an industrial tubular reactor using measured plant data from the start-up of the reactor.

The plant reactor system consists of a feed mixer, a pre-heater and a series of seven horizontal vessels with baffles. The reactor is insulated and located outdoors in an open structure. Thermo-elements are located at several positions between the reactor inlet ($z = 0$) and reactor outlet ($z = 1$). The relative position (z) of a thermo-element is calculated by taking the ratio of the volume from the reactor inlet up to the particular thermo-element and the total reactor volume, according to the model assumptions.

The whole scheme of reactions carried out in the tubular adiabatic reactor is simplified to one overall reaction that describes the consumption of the main reactant B:

A + B -> C

This exothermal reaction is carried out in the liquid phase at a pressure level sufficiently high to avoid boiling. Reactant *A* is fed in excess, because reactant B should be totally converted at the reactor exit.

Data has been collected during a reactor startup. The data set contains the dimensionless time, the dimensionless fluid velocity, the dimensionless reactant B concentration at the reactor inlet, the dimensionless reactor inlet temperature, and the dimensionless reactor temperature at various axial reactor positions.

The objective is to fit optimum values of the following model parameters to this data:

γ = dimensionless activation temperature

Da_r = Damköhler number

Pe_{mr} = Péclet number for mass dispersion

Pe_{hr} = Péclet number for heat dispersion

U^* = dimensionless heat transfer coefficient

The experimental data and the reactor model are taken from the open literature [Verwijs, J.W., H. van den Berg, K.R. Westerterp; *Startup of an Industrial Adiabatic Tubular Reactor*; AIChE J., Vol 38, pp. 1871-1880 (Dec. 1992)].

- Model Equations for EUROKIN Test problem 4
- Running EUROKIN Test Problem 4

Model Equations for EUROKIN Test Problem 4

The plant reactor is an adiabatic tubular reactor and its dynamic behavior can be described by three, dimensionless partial differential equations.

Mass balance for component *B*:

$$\frac{\partial \Gamma_B}{\partial \sigma} = -\phi_v \cdot \frac{\partial \Gamma_B}{\partial z} + \frac{1}{Pe_{mr}} \cdot \frac{\partial^2 \Gamma_B}{dz^2} - Da_r \cdot e^{(\gamma \cdot (1-\frac{1}{\theta})} \cdot \Gamma_B$$

The reaction is exothermic, and the reactor vessel is heated up by the fluid during start-up. As a result, the reactor vessel acts as a heat sink during startup, and influences the dynamic behavior of the system. Therefore, the heat transfer between the fluid and the reactor vessel is taken into account in the model.

Energy balance for the fluid:

$$\frac{\partial \theta}{\partial \sigma} = -\phi_v \cdot \frac{\partial \theta}{\partial z} + \frac{1}{Pe_{hr}} \cdot \frac{\partial^2 \theta}{dz^2} - U^* \cdot Da_r \cdot (\theta - \theta_w) + \Delta\theta_{adr} \cdot Da_r \cdot e^{(\gamma \cdot (1-\frac{1}{\theta})} \cdot \Gamma_B$$

Energy balance for the reactor vessel, neglecting heat transport through the tube wall by conduction in the axial direction and heat losses to the surroundings:

$$\frac{\partial \theta_w}{\partial \sigma} = Da_r \cdot U^* \cdot \omega_h \cdot (\theta - \theta_w)$$

Initial Conditions:

$$\sigma = 0; \quad \Gamma_B(z,\sigma) = \Gamma_B(z,0) = 0$$
$$\theta(z,\sigma) = \theta(z,0)$$
$$\theta_w(z,\sigma) = \theta_w(z,0) = \theta(z,0)$$

Boundary conditions:

$$\sigma \geq 0 \quad and \quad z = 0 \quad ; \quad \Gamma_B(0,\sigma) = \psi_B$$
$$\theta(0,\sigma) = \nu_\theta$$

$$\sigma \geq 0 \quad and \quad z = 1 \quad ; \quad \frac{\partial^2 \Gamma_B}{\partial z^2} = \frac{\partial^2 \theta}{\partial z^2} = 0$$

Where the following dimensionless quantities are:

σ	=	dimensionless time
$\Gamma_B(z,\sigma)$	=	dimensionless reactant B concentration at axial position z and time σ
ϕ_v	=	dimensionless fluid velocity
z	=	dimensionless reactor length, [-]
Pe_{mr}	=	Péclet number for mass dispersion
Pe_{hr}	=	Péclet number for heat dispersion
Da_r	=	Damköhler number
γ	=	dimensionless activation temperature
$\theta(z,\sigma)$	=	dimensionless reactor temperature at axial position z and time σ
$\theta_w(z,\sigma)$	=	dimensionless reactor vessel temperature at axial position z and time σ
$\Delta\theta_{adr}$	=	dimensionless adiabatic temperature rise
ω_h	=	dimensionless heat capacity ratio
ψ_B	=	dimensionless reactant B concentration at the reactor inlet (z = 0)
υ_θ	=	dimensionless reactor inlet temperature
U^*	=	dimensionless heat transfer coefficient

Running EUROKIN Test Problem 4

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder:

C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\EUROKIN

To run this example:

1 Copy the file EUROKIN4acmf from the examples folder to a convenient working folder.

2 Start Aspen Custom Modeler, and from the File menu, click Open.

3 Select the file EUROKIN4acmf.

4 Click the run button to run the simulation.

Note: This example is very CPU intensive, and may take several hours to complete the run.

8 Steady-State Estimation of a Methanol Reactor

The reactor models these three components:

Component	Represents
A	H2O
B	CH4
A2B	CH3OH

Seven experiments are run, varying the feed flow rates to a reactor. The measured values are the outlet concentrations of the components A and B.

The results of these experiments are coded in three ways for this example:

- Using the Estimation dialog box
- Visual Basic Script at the flowsheet level in the simulation
- Excel spreadsheet (Excel 7.0 or later)

Running the Methanol Reactor Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, the files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\MethReactorSSEst

To run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\MethReactorSSEst

You can run the methanol reactor example from the Estimation dialog box, or with a Visual Basic script, or from an Excel spreadsheet.

Using the Estimation Dialog Box

To run the example using the Estimation dialog box:

1. From the File menu, click Open.
2. Open the file **MethReactorSSEstSetupComplete.acmf**. If you copied the files to the example working folder, this file is located in: C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\MethReactorSSEst
3. To inspect the setup of the estimation experiments, from the Tools menu, click Estimation.
4. Click the Estimated Variables tab and note that it shows those variables which are to be estimated.
5. On the Steady State Experiments tab, seven experiments are listed. Select an experiment and click the Edit button to view its details.

   In the dialog box, note the following two tabs:

   - **Measured Variables tab**. Lists variables for which there is observed data, as well as enabling you to edit the data. It also displays results when the estimation run is complete.
   - **Fixed Variables tab**. Lists variables which have prescribed values in the experiment and enables you to edit of those values.
6. Run the simulation. The results can be inspected on the Measured Variables tab for each experiment.

Running with a Visual Basic Script

To run this example with a Visual Basic script:

1. From the File menu, click Open.
2. Open the MethReactorSSEst folder. If you copied the files to the example working folder, these files are located in:

   C:\My Simulation\Modeler\MethReactorSSEst
3. Double-click the file **MethanolReacSSEst.acmf**.
4. In the All Items pane of the Simulation Explorer, click Flowsheet.
5. In the Contents pane, double-click the script called SSEstimation.
6. The results of the Estimation run are shown in the Simulation Messages window.

Running with an Excel Spreadsheet

To run this example with an Excel spreadsheet:

1. From the File menu, click Open.
2. Open the MethReactorSSEst folder. If you have installed Aspen Custom Modeler in the default location, this is:

   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\MethReactorSSEst
3. Double-click the file **MethanolReacSSEst.acmf**.

4 If you have Excel 97 or later, open the spreadsheet ReactorSSEst.xls.

– or –

5 If you have Excel 70, open the spreadsheet ReactorSSEst95xls. This is in Examples\Off95ExcelSheets. If a dialog box appears asking if you want to enable macros, click "Enable Macros".

6 Check the data on the Input Data page.

7 On the Results worksheet, run the estimation by clicking the Calculate button and when the estimation simulation is complete, view the results.

8 For direct access to the estimation algorithm options, select the Estimation Solver Options worksheet. Click Get to read values and Put to apply values.

9 You can analyze the results of the estimation simulation to compare the estimated solution with the measured data. After an Estimation run, select the Analyze worksheet and click Calculate.

10 You can check the sensitivities of the estimated variables to the measured variables. Select the Sensitivity Analysis worksheet and click Calculate.

Controlling NL2SOL Options

You can read and change the values of the estimation routine NL2SOL solver options in the following ways:

- Use the Estimation Solver Options worksheet in the Excel file ReactorSSEst.xls or ReactorSSEst95.xls
- Use the supplied Visual Basic form NL2SOLOpt.vbp. You need either Visual Basic 5.0 Professional or Enterprise Edition, or the Control Creation Edition.
- On the Setup tab of the Estimation dialog box, click the Options button to open the Solver Options dialog box with the Estimation tab active.

9 Catalytic Bed Regeneration Example

The catalyst in a packed bed reactor needs to be regenerated by passing superheated steam through the bed. The catalyst at all points along the bed needs to be held at 700°C for a period of time to ensure all the catalyst if fully regenerated. The process stream is at 600°C.

The temperature profile in the bed between the gas and solid in the bed is described by the following equations.

$$\frac{\partial T_g}{\partial t} = -\frac{v}{\varepsilon}\frac{\partial T_g}{\partial z} - G\_const(T_g - T_s)$$

$$\frac{\partial T_s}{\partial t} = S\_const(T_g - T_s)$$

Where:

T_g	=	Gas temperature
T_s	=	Solid temperature
t	=	Time
v	=	Steam velocity
ε	=	Bed voidage
G_const and *S_const*	=	Heat transfer constants
z	=	Bed length

The partial derivative for temperature in terms of length is approximated by the finite difference equation:

$$\frac{\partial T_{g_i}}{\partial z} \approx \frac{T_{g_i} - T_{g_{i-1}}}{\Delta z}$$

Where: i = Element number

Running the Catalytic Bed Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, the files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\Regen

To run this example:

1. Copy the files in the example directory to a convenient working folder, for example:
   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\Regen

You can run this example in several ways:

1. Loading the text input file and use Aspen Custom Modeler tables, run-time plots and profile plots to make input disturbances and view key results
2. Using a Visual Basic control panel to run the simulation without interacting with the Aspen Custom Modeler User Interface

To run this example from Aspen Custom Modeler, complete these steps:

1. From the File menu, click Open.
2. Open the Regen folder. If you copied the files to the example working folder, these files are located in:

   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\Regen
3. Double-click the file **Regen.acmf**.
4. Start a dynamic run.
5. View the effect of the change in feed temperature in the plot SolidTempPlot in the block Bed1 To view this plot, click the block Bed1 with the right mouse button then point to Forms and click SolidTempPlot.
6. You can run a predefined task to simulate changes in the temperature of the steam entering the bed:
7. In the Simulation Explorer, under Flowsheet, double-click the RegenerateCatalyst task icon to activate the task. This task simulates one cycle of catalyst regeneration.

   You can also run the simulation from Visual Basic.

Running the Catalytic Bed Example from Visual Basic

You can use Visual Basic to start an Aspen Custom Modeler session and run the simulation without interacting with the Aspen Custom Modeler User Interface.

1. Run the executable **RegenDemo.exe**.

2 Click the Start Regen Demo button. Ensure that the pathname displayed in the box at the bottom of the control panel shows the directory where you have installed the example files.

3 Wait for confirmation that the simulation is ready to run.

4 If you want to see an Excel real-time temperature profile, click the Launch Excel button.

   If a dialog box appears asking if your want to update your workbook, or your macros, click Yes.

5 Click the Run button and change the feed vapor temperature with the slider.

6 To close the simulation, in the control panel, from the File menu, click Exit. You are asked if you want to close the Aspen Custom Modeler session at the same time.

The Visual Basic source code is included in the files RegenDemo.vbp and RegenDemo.frm. These files are compatible with Visual Basic 5 and Visual Basic 6.

10 Steady-State Methanol Reactor with Re-Cycle Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, the files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\SSMeth

To run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\SSMeth

The Methanol Reactor models a methanol production process including a recycle loop.

A feed stream of Hydrogen, Carbon Monoxide, Carbon Dioxide, Methane and Water is mixed with the recycle stream and fed to a reactor operating at 533 Celsius and 120 bar.

Two reactions are modeled in the reactor:

- The methanation reaction between Carbon Monoxide and Hydrogen
  CO + 2H2 -> CH3OH
- A side reaction between Hydrogen and Carbon Dioxide.
  H2 + CO2 -> CO + H2O

Methanol and Water are extracted as a product stream. To prevent the build up of impurities a proportion of the recycle stream is purged to waste.

Running the Methanol Example

To run the Methanol example, SSMethDemo.acmf, you need to complete the following steps:

- Loading the example
- Building the flowsheet
- Connecting the blocks
- Tidying the flowsheet

- Entering the input data
- Running the simulation and reviewing results
- Changing the simulation
- Viewing changes
- Saving new specifications

Note: There is also a Methanol example with the flowsheet already built, SSMeth.acmf. If you choose to load this file, go straight to Running the Simulation and Reviewing Results.

Loading the Example

To load the example:

1 From the File menu, click Open.

2 Go to the SSmeth folder. If you copied the files to the example working folder, these files are located in:
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\SSMeth

3 Double-click one of the following:

File Name	Description
SSMethDemo.acmf	Contains text descriptions of the models required for the example, as well as variable type definitions. You need to build the flowsheet yourself.
SSMeth.acmf	Contains the models and flowsheet. Because this example is ready to run, go directly to Running the Simulation and Reviewing Results.

Building the Flowsheet

If you have opened the SSMEthDemo.acmf file, you need to build the flowsheet:

1 In the All Items pane of the Simulation Explorer window, under Custom Modeling, expand the Models folder.

2 Click the first model in the list, FEEDER. Elements associated with the model are shown in the Contents pane of the Simulation Explorer window.

3 From the Contents pane, drag and drop the FeedIcon onto the flowsheet.

4 Double-click the specification indicator on the status bar. (The specification indicator is either a green square or a red arrow.) Keep the Status window open. You will see the degrees of freedom change as you place the blocks.

5 Continue building the flowsheet:

Block	Model
B1	FEEDER
B2	MAKEUP
B3	REACTOR
B4	SEPARATOR
B5	SPLITTER

When you have placed the blocks, the flowsheet should look like this:

User Notes

You can add your own comments to blocks, streams, structures, component lists and snapshots. They can be accessed from the Properties dialog for any of these items. The Properties dialog can be accessed from the context menu for the item in the Explorer except snapshots that do not appear in the Explorer. These must be accessed from the snapshot management dialog. For blocks and streams that appear on the flowsheet the Properties dialog can be accessed from the flowsheet item context menu as well.

You can also set up a default value for all instances of a model using the following language statement in a model

UserNotes: "My model";

The text My Model will then appear for all instances of this model. User Notes can also be accessed via OLE automation.

For blocks, stream and structures it can be accessed using the flowsheet path to the item.

```
e.g.  application.msg B1.UserNotes
```

This will print out the current user notes to the simulation messages window.

For component lists it can be obtained from the component list object

```
e.g.  Set objPhysProps=
application.simulation.Flowsheet.Properties

application.msg
objPhysProps.ComponentList("Default").UserNotes
```

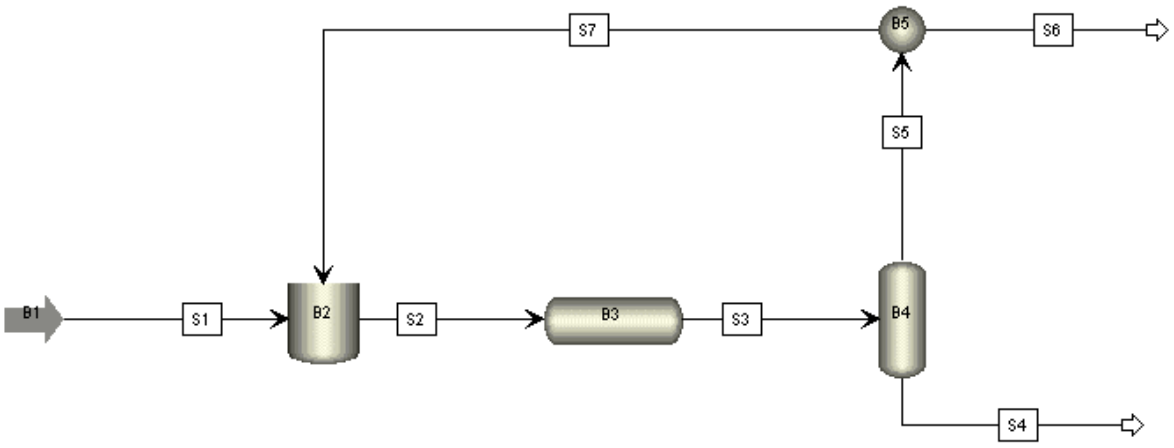
For snapshot and results use the Results object:

```
e.g.  application.msg
Application.Simulation.Results.GetSnapshot(3).UserNotes
```

Connecting the Blocks

If you are using the SSMEthDemo.acmf file, after you have placed all the blocks on the flowsheet, you need to connect them. To connect the blocks:

1. In the All Items pane of the Simulation Explorer, click Stream Types. From the Contents pane, drag the Connection icon onto the flowsheet. Blue arrows appear on the icons where ports are available.
2. Drop the Connection on a port and connect to another port as shown in the flowsheet diagram. Repeat to complete the flowsheet. The finished flowsheet should look like this:



Tidying the Flowsheet

You may need to reposition blocks on the flowsheet.

To do this:

1. Click the right mouse button on the flowsheet and then click Zoom Full to ensure the flowsheet is as large as possible.
2. To select all the blocks, hold down the mouse button and drag the cursor to round them to form a box.
3. From the Flowsheet menu, click Align Blocks.
4. From the Flowsheet menu, click Reroute Streams.

Note: If you want to use the Flowsheet as Wallpaper option, make sure the Flowsheet window is active, and then from the Window menu click Flowsheet as Wallpaper. This maximizes the flowsheet window so that it becomes the background for your

simulation.

Entering the Input Data

You can enter the required input data from:

- AllVariables tables
- Microsoft ActiveX control forms (user-defined forms)

To access these from the flowsheet:

- Click the required block to select it, then click with the right mouse button and point to Forms. From the displayed list, click the required form.

Entering Data from the AllVariables Table

Enter the data for the Feeder block from the AllVariables table. To do this:

1. In the Flowsheet window, right-click on the block and select Forms\AllVariables from the list. An AllVariables table that contains all the variables in the block appears. (Double-clicking block B1 displays the Feeder Spec.)

Tip: To change which columns are shown, click the right mouse button on the table and then click Properties. Value and Spec are usually the most useful columns.

From this table, you can alter the TOTAL flow out of the feeder block. Values to run the simulation successfully are already defined. Close the AllVariables table.

Entering Data from the User Defined Forms

Complete the data entry by using the customized forms. A user-defined form is a Microsoft ActiveX control form, designed to display data for a particular model. You can design your own forms for your own models. To access the form:

1. Make sure B3 (the Reactor block) is selected, then click it with the right mouse button, point to Forms and click ReactorSpec.
   A user-defined form appears. This form enables you to enter the correct data easily. To change data in this form, make sure you click outside each box after entering a number, otherwise the value is not retained.
2. Repeat for block B5 (the Separator block), and observe the purge ratio value.

Running the Simulation and Reviewing Results

To run the simulation and review results:

1. After all the data has been entered, you can show the user defined form for the Separator B4

The numbers change when a result is calculated.

2. Make sure the run mode box on the Run Control toolbar shows Steady State and click the Run button.
3. Use the user-defined forms to look at the results over the whole flowsheet.
4. Enter different values in the user-defined forms and run the simulation to get different results.
5. You can perform further runs, changing the value of the feed flow total and/or the component mole fractions in the AllVariables table for block B1

Changing the Simulation

You can now try altering the structure of the simulation.

1. Open the Status window, if it is not currently displayed, by doing one of the following:
   - Double-click the green square on the status bar.
   - From the View menu, click Status Window.
2. Click the Fixed Changes tab.
3. In the Flowsheet window, double-click the feed block, B1
4. In the AllVariables table, click the Spec column cell for Total.
5. Change the spec from Fixed to Free.

   The simulation is now underspecified by one. On the tab for Fixed Changes, the variable that you just changed appears on the unFixed list.
6. Open the AllVariables table for B4, the separator.
7. Change the spec for the variable meprod from Free to Fixed.

   The simulation is square again.
8. Now run the simulation again to calculate the required feed for the specified methanol production rate.

Viewing Changes

To view the changes you have made, edit the Feeder model. To do this:

1. In the Simulation Explorer, expand Custom Modeling and expand the Models folder.
2. In the Contents pane, click FEEDER with the right mouse button and then click Edit.

   The text equations for the model appear in a Text Editor window. Towards the end of the model description, the specification value for some variables are set to Free or Fixed. These are the default specifications for the model.
3. To view the changes from the default specification, double-click the specification indicator to open the specification Status window, if it is not already displayed.

Saving New Specifications

You can save the new specification values by creating scripts. To do this:

1. If the Status window is not already open, open it. To do this, from the View menu, click Status Window then click the General tab.
2. Click the Script button, supply a name for the new script and click OK.
3. In the All Items pane of the Simulation Explorer, click Flowsheet.

   The script appears in the Contents pane.
4. To edit the script, click with the right mouse button on the script icon then click Edit.

   The changes made from the default specification values are recorded in the script.
5. To apply the changes made in the script, double-click the script icon to invoke it.
6. To revert to the default specification provided in the models, in the Status window, on the Fixed Changes tab, click the Reset button.

   Now you can easily switch between the default spec and the new spec between runs.
7. You can use a Microsoft Visual Basic application to switch between a design and rating case. You can fix the total feed flow rate to the flowsheet, or switch to fixing the total methanol produced. The front end provides you with the values of the key data.

If you are running the file	Use this Visual Basic form from the SSMeth folder
ssmeth.acmf	MethanolLoop.vbp
ssmethdemo.acmf	MethanolLoopDemo.vbp

Homotopy

You can run a Homotopy example using the input file SSMeth.acmf.

What is Homotopy?

Homotopy enables you to move from one steady state solution to another steady state solution in small increments. In some circumstances, the simulation may not converge for the target steady-state solution from the current steady-state solution. Homotopy allows you to approach the target steady-state solution in stages, thereby improving the chances that the target steady-state solution is reached.

This example uses a Visual Basic Script to initialize a solution towards a set of target values for fixed variables.

When Flowsheet is selected in the All Items pane, the Homotopy script is shown in the Contents pane (this applies to SSMeth only, not SSMethDemo). You can edit this script to change the target values of the fixed variables, and alter the number of steps taken to reach the target steady-state solution.

Running the Homotopy Example

You can run the Homotopy example from the input file SSMeth.acmf.

To run the Homotopy example:

1. In the All Items pane of the Simulation Explorer, click Flowsheet. In the Contents pane, double-click the script called RunHomotopy.
2. This script initializes the homotopy target variables and enables homotopy.
3. To see the results of running this script, from the Tools menu, click Homotopy.
4. In the Homotopy dialog box, you can change the initial and target values of the variables by double-clicking the variable names in the list.
5. To start the solution, click the Run button.

The simulation solution is moved from the current solution point to the solution at the target values of three fixed variables:

1. Feed Flowrate
2. Reactor Temperature
3. Reactor Pressure

The script steps from the current solution to the target solution.

You can reset the simulation to the original values of the fixed variables by double-clicking the script called RestartHomotopy.

Running the Optimization Example

You can run an optimization simulation from the input file SSMeth.acmf.

The optimization maximizes the profit made by the reactor and recycle loop. The profit is calculated from the amount of methanol produced minus the amount of effluent purged from the recycle.

In order to maximize the profit, the optimization alters the values of the following variables:

- Reactor Temperature
- Reactor Pressure
- Purge Ratio

The optimization takes place subject to the following conditions:

- Reactor Temperature > 500 C

- Reactor Temperature < 550 C
- Reactor Pressure > 100 bar
- Reactor Pressure < 150 bar
- Effluent Methanol fraction < 0.003

You can access the definitions of the objective function to be maximized and the constraints by editing the Flowsheet Constraints in the Simulation Explorer.

To open the Optimization dialog box, on the Tools menu, click Optimization. The Decision Variables tab shows the variables which will be varied to achieve the optimum, and the Objective Variables tab shows the value of the variables which are being optimized. Run the optimization by changing the run mode to Optimization and clicking the Run button.

11 Water Hammer (PDE) Example

This example simulation consists of a pipeline terminated by a valve. If the valve is slammed shut, a pressure wave travels back up the pipe. The pipe expands due to the increased pressure. When the pressure wave reaches the pipe inlet, the pressure is relieved by back flow. The pressure wave oscillates back and forth along the pipe. The pressures decrease as energy is absorbed by friction.

Depending on the severity of the valve closure, either the flow is halted without any problems, or else the pressure in some or all sections of the pipe falls below the vapor pressure of water. The model does not simulate cavitation, but if you see the pressure falling too far below one bar, you can estimate that cavitation can occur, and you need to close the valve more gently.

With the current simulation configuration, closing the valve Cv to 100 causes no cavitation in the pipe following closure, whereas a closure to a Cv value of 10 causes extensive cavitation. The task used to close the valve uses an SRAMP function. You may want to try increasing the period over which the valve closes.

Running the Water Hammer Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, the files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\WaterHammer

To prepare to run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\WaterHammer

To run the example using a task, follow these steps:

1 Double-click the file **WaterHammer.acmf** to open it.

2 The example includes the built DLL file, wh.dll, which contains the procedure Fortran. If you want to rebuild this DLL:

   Rename or delete the existing wh.dll.

   Click the procedure with the right mouse button and then select Generate Code.

   Select the Wrappers, Generate and Execute options and then click OK.

   The DLL is rebuilt.

3 Ensure that the run mode is Steady State, and then run the simulation.

4 Double-click the two plots of pressure profiles and flows along the pipe, MovingPressureProfile and FlowsPlot.

5 Ensure the run mode is Dynamic and then run the simulation.

   At time = 0.75, the valve closes. Observe the reaction to the valve closure on the plots.

12 High Temperature Shift Reactor (PDE) Example

This example is for users of Aspen Custom Modeler with Properties Plus. You therefore have to have Aspen Plus installed to use this example.

Note: If you are using a Simulation Engine installation from a client machine, you must make sure that you run Aspen Plus using the same server and the same working folder as you have configured for Aspen Custom Modeler.

This example models a high temperature shift reactor, which is important for many industrial plants, including ammonia plants. The high temperature shift reactor example illustrates:

- Complex reaction kinetics
- Generalized solution of partial differential equations
- A flowsheet initialization strategy which includes generalized calls to initialization scripts, and homotopy scripts of each model
- A dynamic operability study of the reactor

Reactor Kinetics

The reactor uses a solid catalyst to complete the gas phase water shift reaction:

$$CO + H2O \longleftrightarrow CO2 + H2$$

The reaction is exothermic, with the heat effects determined from enthalpies which include heats of formation. The reactor is considered adiabatic because the throughputs are high and the area (which is insulated) available for heat loss is small.

Numerous kinetic relationships have been developed for the water gas shift reaction over iron based catalysts used in high temperature shift reactors. Langmuir-Hinshelwood models have been shown to be superior to power law-based models for this reaction. Consequently, Langmuir-Hinshelwood kinetics are used a sub-model.

The specific form of the rate expression used is:

$$r = \frac{k * \left(pCO * pH2O - \frac{pH2 * pCO2}{K1} \right)}{\left(DEN \right)^2}$$

Where:

$$DEN = 1 + KCO * pCO + KH2 * pH2 + KCO2 * pCO2 + KH2O * pH2O$$

The reaction rate constant, *k,* is a function of temperature, activation energy, and frequency factor, in the classic Arrhenius form:

$$k = A * e^{-E / RT}$$

Literature data indicate the activation energy in the absence of diffusion effects as 121.8 kJ/kmol. The frequency factor was determined from literature and plant data.

Similarly, the adsorption equilibrium constants are functions of temperature, heats of adsorption, and a frequency factor:

$$Ki = Ai * e^{\Delta Hi / RT}$$

$$i = CO, H2, CO2, H2O$$

The reaction equilibrium constant is a function of temperature. The water gas shift reaction equilibrium constant is a function of temperature.

$$Ki = f(T)$$

Diffusional Effects

The reactor kinetics discussed are intrinsic kinetics that are valid in the absence of diffusional or heat transfer resistances. Bulk phase concentrations (partial pressures) and temperatures are not present at the active sites within the catalyst pellets, since the pore structure can offer significant resistance to diffusion, and mild resistance to heat transfer exists between the bulk phase and the pellet surface. Little resistance to heat transfer exists within the pellets.

The diffusional effects are accounted for using the classic effectiveness factor.

$$\eta = \frac{actual\,rate\,throughout\,pellet\,with\,resistances}{rate\,evaluated\,without\,resistances}$$

Evaluating the numerator of this expression requires simultaneous integration of the rate and diffusion relationships. Simplification of the effectiveness factor calculation can be applied for conditions which cause the diffusional resistance to be large. The simplification is:

$$\eta = \frac{3}{r_p} \sqrt{\frac{K(D_e)}{k(K+1)\rho_p}}$$

Instead of calculating the effectiveness factor in the model, which would be a computational burden, the effectiveness factor is entered as a function of length as a constant profile. The effectiveness factor is primarily a function of

the catalyst pellet size, pore size, and pore size distribution. Because this is a relatively weak function of operating conditions over the normal range of these conditions, the effect of imposing it as constant profile is very small. The results of reactor simulations with the effectiveness factor profiles fixed agree very well with measured results from industrial reactors, and with results presented in the literature.

The relationship among actual reaction rate, intrinsic rate, effectiveness factor, and catalyst activity is:

$$r_{i,observed} = \eta_i \times \alpha \times r_{i,intrinsic}$$

Where:

η	=	Effectiveness factor
α	=	Relative catalyst activity (accounting for aging, sintering, pore closure)
$r_{i,intrinsic}$	=	Rate calculated with bulk fluid conditions

The catalyst activity is calculated as a parameter, updated from operating data.

The sub-model LHHW is used to model these kinetics. A Microsoft Visual Basic script in the main reactor model, CONSTANTS can be used to fix all of the kinetic parameters.

Modeling Strategy

Although the reactor can be treated as a simple equilibrium reactor, a kinetic model is necessary to reproduce dynamic behavior. The main reactor model uses the Aspen Custom Modeler domain and distribution models, thus allowing the partial differential equations for the reactor and the solid catalyst to be solved easily. Several methods can be selected for the solution of the reactor equations, such as orthogonal collocation of finite elements and finite differences.

Representation includes a partial differential equation for the convective reaction equation, convective gas energy equation, and conductive solid energy equation.

Initialization Strategy

The equations of the reactor are highly non-linear, and will require a comprehensive strategy to converge in steady state. Fortunately, Visual Basic scripts in Aspen Custom Modeler will allow easy convergence of the simulation. Several features of the reactor which are modeled are:

- The chosen state variable is concentration for the convective equation
- The reactor is adiabatic
- Axial conduction is considered down the length of the reactor in the solid catalyst

The most difficult of these features is that the reactor is adiabatic. The temperature profile of the reactor is not known and must be calculated. Unfortunately, the reaction equations are highly dependent on the

temperature, and will not converge if the temperature is freely calculated. However, if the temperature is fixed, the kinetic equations are well behaved. In addition, the density of the gas is easily estimated by the ideal gas law (high temperature gases are mostly ideal, regardless of the pressure).

Thus, the state concentrations can be estimated from the ideal gas law and the entering molefractions.

In addition, Initial specifications for the solid and gas temperature can be used. These specifications will allow an initial run to converge easily. The Initial script will prompt for the pressure drop (0.3 bar) and the temperature rise (60°C).

In a series of homotopy-type runs, you can change the specifications and perform steady-state runs. If you switch the specifications for solid and gas temperature to fixed conditions, and free the duty, a steady-state run converges easily. Then you can return the gas temperature specifications back to Initial, and fix the duty to adiabatic conditions, thus an adiabatic condition. In the last steady-state run, you can fix the solid temperature slack to 0, and return the solid temperature to specifications to Initial.

Each of these steps is accomplished with block level Visual Basic Scripts, Initialize, Homoinit, Homostep, and Homostep2. These scripts are called from a generalized flowsheet Initialize script, which can be used to converge an entire flowsheet.

Steady-State Design

The reactor operates at over 380°C and 35 bar, and uses an iron oxide/chromium oxide catalyst. The reaction is exothermic, but the reactor is not cooled. The feed composition to the reactor is mostly water and hydrogen (35% each), 15% nitrogen, 8% carbon monoxide, and 5% carbon dioxide. The reactor product is enriched with hydrogen, and much of the carbon monoxide is consumed. Thermal stress is a significant degrading factor for the catalyst. The reactor has a 3.5-meter bed of cylindrical pellets, and the average velocity is about 0.5 m/s. There is a temperature profile and composition from an operating reactor, and the estimation capabilities of Aspen Custom Modeler have been used to adjust the kinetics to plant data.

Dynamic Operability Study

The HTS reactor has a significant thermal mass because of the catalyst bed. A complete study would include the startup, shutdown, and loss of feed cases. A 20% reduction in the feed for about 20 minutes has been used, and the increase of the feed to the normal flow rate.

We want to know what is the effect of this thermal mass, and how will it affect the reintroduction of the feed to the reactor. The model is completely dynamic and includes the effect of heat-transfer to the catalyst, conduction in the bed, as well as the reaction kinetics and adsorption effects. So the results of the model will predict the same response as the reactor.

Setting Up the Interface

You need to set up the interface for Aspen Custom Modeler and Aspen Plus. However, you only need to do this once:

1 Open the Aspen Plus User Interface.

2 Select Open an Existing Simulation / More Files...

3 Open the Water Shift Reactor folder. If you copied the files to the example working folder, these files are located in:

C:\Program Files\AspenTech\Working Folders\
Aspen Custom Modeler 121\WaterShiftReactor

4 Load the file shift.inp into Aspen Plus, and run it.

Note: This is a properties only input file, so you will not see a flowsheet in Aspen Plus.

5 From the File menu, click Save As and save the simulation as an Aspen Plus document (.apw file), then exit Aspen Plus.

Now you can run the example.

Running the Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, the files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\WaterShiftReactor

To run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\WaterShiftReactor

After you have set up the interface, you can run the Water Shift Reactor example. To do this:

1 From the File menu, click Open.

2 Open the WaterShiftReactor folder. If you copied the files to the example working folder, these files are located in:

C:\Program Files\AspenTech\Working Folders\
Aspen Custom Modeler 121\WaterShiftReactor

3 Click the example file WaterShiftReactor.acmf then click Open.

4 In the All Items pane of the Simulation Explorer, ensure Flowsheet is selected.

5 In the Contents pane, click the script Initialize with the right mouse button and then click Invoke Script.

6 Enter estimates for the pressure drop and temperature increase over the reactor. You can accept the default values.

7 When the Initialize script has completed, change the run mode to Dynamic in the Run Mode box on the toolbar.

8 In the All Items pane of the Simulation Explorer, ensure Flowsheet is selected. Activate the tasks RampDown and Ramp2 by double-clicking on the task icons.

9 Click the run button to start the dynamic run.

13 Vinyl Acetate Monomer Reactor (PDE) Example

This example simulates a catalytic tubular reactor for the production of vinyl acetate monomer from ethylene and acetic acid.

It includes a general purpose dynamic catalytic tubular reactor model which models temperature and composition variations in the radial and axial directions. It models heat transfer between gas and catalyst, and heat conduction and dispersion in both radial and axial directions. Heat transfer from gas and catalyst through the reactor wall is also modeled.

The major model assumptions are:

- All heat created in the reaction on the catalyst surface passes in to the catalyst
- The coolant outside the reactor wall is an uniform temperature along the reactor length
- Constant heat transfer coefficients between gas and catalyst, and gas and reactor wall

The reactor model uses sub-models for reaction and physical property calculations. You can modify the reaction sub-model for simulation of other reaction systems. You can also export the model for use in Aspen Plus.

Running the Vinyl Acetate Monomer Reactor Example

To run the Vinyl Acetate Monomer Reactor Example:

1 In Aspen Custom Modeler, from the **File** menu, click **Open**.

2 Open the **VAMReactor** folder. If you copied the files to the example working folder, this is:

```
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 2004\VAMReactor
```

3 Click the example file **VAMReactor.acmf**; then click **Open**.

4 Click the **Run** button to run the simulation.

5 Right click on the **VAMRctr** block and select **forms**. There are a range of plots and tables that you can use to view the simulation results. These include 3D profiles for gas and reactor temperature, and for Vinyl acetate concentration.

6 Try changing the inlet conditions, such as flow rate, to see the effect on the reactor and the fraction of vinyl acetate in the product stream.

7 Switch the run mode to **dynamic** and **run**.

8 Double click on the feed stream to open the **Feed.Alvariables** table. Try the effect of changing some of the inlet conditions, such as temperature or flow rate.

9 In Simulation Explorer go to Flowsheet. Double click on the Plot OutletTPF to see a plot of the reactor outlet conditions versus time.

Note: This example uses Aspen Properties for physical properties calculations, so you must have Aspen Properties installed to run this example.

Using the Vinyl Acetate Monomer Reactor in Aspen Plus

The Aspen Custom Modeler **reactor** model can be exported for use in an Aspen Plus simulation.

To do this

1 Ensure the **VAMReactor** example is open with Aspen Custom Modeler

2 In Simulation Explorer go to **Custom Modeling, Models, PFR2D**.

3 Right click **PFR2D** and select **Export...** .

4 Change the **Save as** type to ***.msi** to save the model to an **msi** file. When prompted choose to install the model.

5 Exit Aspen Custom Modeler

6 Start Aspen Plus

7 Load the file **VAMReactor.bkp**. This file contains 3 reactor blocks:

- **REACTOR** - This uses the Aspen Plus **RPlug** model to simulate the vinyl acetate monomer reactor. It uses the Fortran code in the file USRCO2.FOR to model the reaction kinetics. Vam.dll contains this code in compiled form and dlopt.opt tells Aspen Plus to load this dll.
- **ACMRCTR** - This uses the Aspen Custom Modeler PFR2D model to simulate the reactor. The input variables have been set to values that

give equivalent behaviour to the Aspen Plus RPlug model. Low radial Peclet numbers simulate perfect mixing in the radial direction, and high axial Peclet number simulate perfect plug flow. A high gas to solid heat transfer coefficient means that the gas and catalyst will be at the same temperature, and a high catalyst heat conductivity means that the catalyst temperature is constant in the radial direction.

- **ACMRCTR2** - This uses the Aspen Custom Modeler PFR2D model to simulate the reactor. The input variables have been set to typical real world values. These are the same values that are used in the Aspen Custom Modeler simulation of the reactor in VAMReactor.acmf.

A **Dupl** block is used to ensure that all 3 reactor blocks have identical feed conditions.

8 Click the **run** button to run the simulation.

9 Look at the stream results for the outlets of the three reactor blocks. As expected **REACTOR** and **ACMRCTR** give almost identical results. The vinyl acetate mole fraction in the product is 3.84%. The **ACMRCTR2** model predicts a higher concentration of 4.62%. This is due to the higher average reaction temperature obtained when modeling radial heat transfer and the heat transfer between the catalyst and gas.

Notes: The **.msi** file is provided with this example, so you can skip steps 1-5 and instead install the .**msi** file directly by double clicking it.

This example uses Aspen Plus, so you must have Aspen Plus installed to run it

14 Dynamic Optimization Example

This example contains a model of a batch fed reactor with five state variables and two fixed variables FeedRate and CoolingRate.

There are two reactions present:

A+B→C (desired product formulation)

B+C→D (competitive reaction)

Product, represented by variable Holdup_C in the simulation, is the quantity that we want to maximize at the end of the process, at final time 30 hours by varying the feed rate.

Constraints on the optimization are a final temperature at 30 hours of between 295 and 300 °C and a path constraint on the temperature throughout the whole process to confine it to less than 520 °C. To achieve these constraints the cooling rate will also be varied.

Running the Dynamic Optimization Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, the files for this example will be in the folder
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\NonIsothermalReactor

Preparing to run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\NonIsothermalReactor

To run this example:

1. From the File menu, click Open.
2. Open the NonIsothermal folder. If you copied the files to the example working folder, these files are located in:

C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\NonIsothermalReactor

3 Double-click the file nonisothermalreactor.acmf.

Viewing the Dynamic Optimization Inputs

To view the dynamic optimization inputs:

1 Double-click the model in the flowsheet to display the table of optimization variables.

2 From the Tools menu, click Optimization

3 In the Optimization dialog box, note that Perform Dynamic Optimization is selected.

4 Select the Control Discretization tab. Note the Final Time of 30 and the Number of Elements of 15

This means that the control variables will have 15 switching points for the optimizer to vary.

5 Click the Objective Function tab in the Optimization dialog box. Holdup_C is shown. Note that the value of Holdup_C will be maximized.

Note: You can insert an objective variable by dragging and dropping it from a table.

6 Click the Control Variables tab in the Optimization dialog box.

7 Note that CoolingRate and FeedRate have been selected.

8 Highlight CoolingRate and press the Edit Initial button to see the initial estimates for the optimal control profile.

Viewing the Constraints

1 In the Optimization dialog, select the Dynamic Constraints tab.

2 Note that there is a final time and path constraint on the temperature.

3 Highlight the final time constraint on the temperaure and press the Edit button. Note that the temperature is limited to between 295 and 300.

4 Highlight the path constraint on the temperature and press the Edit button. Note that the temperature is limited to be below 520 at the end of each element. This is a discretization of the path constraint into 15 interior point constraints. .

5 Perform the optimization run.

Creating a Task to Implement the Optimal Profiles

To use a task featuring the optimal profiles calculated for the control variables:

1. Click the Re-Start button and change the run mode to Dynamic.
2. From the optimization dialog box, click the Task button.
3. When prompted, enter a name for the task.

   The task is generated.
4. Close the Optimization dialog box.
5. In the Contents pane of the Simulation Explorer, ensure Flowsheet is selected, and then double-click the task you have just created to activate it.
6. From the Run menu, click Pause At and set the pause time to 30 hours.
7. Ensure the ControlPlots plot is open.
8. Perform the dynamic run.
9. Click with the right mouse button on the plot and from the menu that appears, click Show As History.

Observe how the final temperature constraint is satisfied within tolerance.

15 CDI Example

A CDI example is supplied with Aspen Custom Modeler. This example is a model of a column and side stripper for the separation of a mixture of benzene, toluene, and xylene. Benzene and xylene are the main column's top and bottom products respectively, and toluene is the product from the sidestripper. This example includes scripts to:

- Specify the CDI input and output variables
- Generate matrices which you can then examine to determine the suitability of the input variables to control the specified output variables

Running the CDI Example

If you have installed in the default location, files for this example will be in the folder:
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\BtxCDI

To run the CDI example supplied with Aspen Custom Modeler:

1 Create a subfolder in your working folder which has the name btxcdi.

2 Copy the files btxcdi.acmf and btxallp.dll to this new subfolder. If you copied the files to the example working folder, these files are located in:

C:\Program Files\AspenTech\Working Folders\
Aspen Custom Modeler 121\btxcdi

3 Alternatively, you can copy the dll to a folder which is on your path.

4 Navigate to the btxcdi folder and open the file btxcdi.acmf.

5 Ensure the run mode is Steady State.

6 From the Tools menu, click Snapshots.

7 In the Snapshot Management dialog box, click the result labeled Steady State to select it, then click the Copy Values button.

8 On the Matches Found dialog box, click OK to copy the variables and close the Snapshot Management dialog box.

9 Perform a steady-state run.

10 In the All Items pane of the Simulation Explorer, click Flowsheet.

11 Invoke the CDI run script CDI_LINEARISE_RUN.

This script specifies the CDI input and output variables and then generates the state space matrices at the converged steady state solution. These

appear in your current working folder. The RGA and G matrices can be examined to determine the suitability of the input variables to control the specified output variables.

12 This script also generates an MDL file which can be used as input into DMCplus. The MDL file contains results from a step change in the chosen variables and can be used to design a DMCplus controller for this simulation.

Now edit a script to generate the A matrix.

Performing a Dynamic Run and Generating the A Matrix

To perform a dynamic run and generate the A matrix:

1 Change the run mode to Dynamic.

2 From the Run menu, click Pause At.

3 Type a Pause at time of 1 and click OK.

4 Perform the dynamic run.

5 Select the CDI_LINEARISE_RUN script. Click with the right mouse button on it and from the menu that appears, click Edit.

6 Edit the script so that only the A matrix is produced, that is, add this line:

```
LINEARISE.MatricesRequired "A"
```

7 Save the changes and invoke the changed script. This will generate the A matrix at time 1

The A matrix can be used to determine the linearised stability of the BTX column at time 1.

16 pH Controller Example using Simulation Access eXtensions

This example models a pH control system. An acid effluent is fed into a stirred tank and neutralized using alkali. The flow of alkali is controlled by a PID controller that measures the pH of the tank contents. A step change in the acid flow from 0.025 to 0.03 is introduced at time 5.0 to test the controller response.

This example includes a simple example of using the Simulation Access eXtensions (SAX) interface. The interface code is supplied and can be used as a starting point in the development of your own SAX interface.

Running the pH Control Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder:
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\SAX

To run this example:

- Copy the files in the example directory to a convenient working folder, for example:
  C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 12.1\SAX

To load the Simulation Access eXtensions example:

1. From the File menu, click Open.
2. Locate the SAX folder. If you copied the files to the example working folder, this is:
   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\SAX
3. Double-click the file **phcon.acmf**.

4 In the Explorer select Flowsheet and double-click the task AcidFlowStep to activate it, then double-click AcidFlowPlot to open it.

5 Run the simulation. You will see a step change in the acid flow rate at 5 hours, and the resulting change in pH in the tank with time.

Using SAX

You can run this example using SAX to introduce a disturbance in the acid feed flow rate at 5 hours. To do this:

1 Modify the makefile and create the dll as described in Modifying the Makefile and Building the DLL.

2 If you have already run the simulation restart it and ensure that the task AcidFlowStep is not active.

3 From the Tools menu, click Simulation Access eXtensions. The Simulation Access Extensions dialog box is displayed.

4 Make sure the Output Variables tab is on top by clicking it. Note that the variable CST.Acid.flow has been added to this list of variables for transmission to the SAX function.

Tip: An easy way to add variables to this list is to drag and drop them from a table.

5 Click the Procedure tab and note that the following data has been entered:

Function name:	SAXFunction
Library name:	C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\SAX\phconsax.dll (or location of your installation: change the library name if necessary)

6 Ensure that the After Dynamic Step check box is selected, so that the function is called after each step. Also select the Enable Simulation Access eXtensions check box to enable SAX and then close the SAX window.

7 In the Explorer, click Flowsheet and then double-click AcidFlowPlot to open it.

8 Run the simulation. You will see a step change in the acid flow rate at 5 hours, and the resulting change in pH in the tank with time.

Defining the SAX Interface Function

The C code to be used is in the files **sax_example.h** and **sax_example.c**.

The main function is called SAXFunction. This corresponds to the name given to the function in the SAX dialog box. It has a fixed argument list, so that it can be found when the DLL is loaded.

The body of the code is a large switch statement, with a case for each of the possible types of event. In this case, only three events are of interest: those at the beginning and end of a simulation, and the event after a dynamic step.

At the start of a simulation, which is whenever a new simulation is loaded, or a structural change is made to a loaded simulation, a number of arrays are allocated. These need to be allocated for both lists of variables being passed to SAX. For each, a list of name pointers and a list of type specifiers are allocated, with one entry for each variable in the list, and three smaller arrays for real values, integer values, and string values. Note that additional arrays may have to be allocated if we wished to manipulate variable bounds or other variable attributes. These allocations are performed in the function GenerateProblemData().

At the end of a simulation, these same data arrays are deleted by the function DeleteProblemData().

After a dynamic step, we need to check the simulation time and the value of CST.ACID.FLOW, which in this case is the only variable passed in, and change it as appropriate.

This is done by the following code:

```
case SAX_AFTER_DYN_STEP: /* after a dynamic step */
   {
   DiagPrint("\nSAXFunction, after dyn step:");
   LoadVector(nOut, OutList, OutTypes, OutReals, OutInts,
OutStrings);
   /** PHCON SPECIFIC **/
      {
      /*
* Since there is only one variable, it must be in
       * position [0] of our data arrays
       */
      int changed=0;
      if (Time>= 20.00)
         {
         if (OutReals[0]!=0.03)
            {
            OutReals[0] = 0.03;
            changed++;
            }
         }
      else
```

```
        {
        if (OutReals[0]!=0.025)
           {
           OutReals[0] = 0.025;
           changed++;
           }
        }
     if (changed)
        {
        ACM_Print(0,
           "SAX: Modified value of %s to %g at time %g",
           OutNames[0],&OutReals[0],&Time);
        UnloadVector(nOut, OutList, OutTypes, OutReals,
 OutInts, OutStrings);
        }
     }
   /** END PHCON SPECIFIC **/
   }
   break;
```

This code loads the current variable data for the variables passed into your SAX function by calling LoadVector (this function is contained in the source file), then checks the time, and the value of CST.ACID.FLOW. If this value needs to be changed, it modifies it, and causes a message to be sent to the Simulation Messages window of the client, using ACM_Print(), to say that it has done so. The changed value is then loaded back into the server by a call to UnloadVector().

This code is designed to be generic. It can be used as the starting point for your SAX applications.

Modifying the Makefile and Building the DLL

Complete the following steps to build the DLL:

1. The makefile is supplied in the examples directory as MakePhconSax. If you have not installed in the default location or have copied the examples files you will need to modify the following definitions.
2. Change the following definitions to correspond with your installation location:

ACMSAXSOURCEPATH	\Program Files\AspenTech\AMSystem 121\Procedures
ACMSAXINCLUDEPATH	\Program

Files\AspenTech\AMSystem
121\Procedures

3 Change the following definitions to correspond with the location of your example directory.

SOURCEDIR	\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\SAX\
DLLDIR	\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\SAX\

4 If you have not already done so, create the directory specified by DLLDIR, otherwise the make will fail.

5 Navigate to the directory containing the file MakePhconSax and then build the DLL by entering the following command in a DOS window:

```
nmake –f MakePhconSax
```

17 Gary Blau Estimation Example

This example is based on the 1986 paper "Nonlinear Parameter Estimation: a Case Study Comparison" by L.T.Biegler, J.J.Damiano and G.E.Blau in the AIChE Journal, Volume 32, Number 1, pages 29 to 45. The paper compares the solutions from several authors to a difficult estimation problem formulated by Dow.

The problem is the estimation of nine parameters from a complex kinetics model of a batch reactor.

Since the estimation is hard to solve, in the Aspen Custom Modeler model:

- Sensible scale factors for the estimated variables are given. These help the numerical behavior of the solvers and improve robustness of convergence.
- Both sides of the Arrhenius equations are logged and logs of the pre-exponential factor estimated rather than the pre-exponential factor; this makes convergence of the problem more robust, since the range of the logged factors is smaller, and the associated sensitivities are better scaled.

Running the Parameter Estimation Example

To run the example:

1. From the File menu, click Open.
2. Open the file **gary blau example.acmf**. If you copied the files to the example working folder, this file is located in: C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 1**2**1\BlauEstimation
3. To inspect the setup of the estimation experiments, from the Tools menu, click Estimation.
4. Click the Estimated Variables tab and note that it shows those variables which are to be estimated.

5 Click the Dynamic Experiments tab. Three dynamics experiments are listed. To view the details of an experiment, select an experiment and click the Edit button. A dialog box with three tabs appears:

- **Measured Variables tab**. This lists variables for which there is observed data. To view (and modify) the observed data, select a variable and click Edit. When the estimation is complete, the resulting table shows results and enables you to create plots.
- **Fixed Variables tab**. Lists variables which have prescribed values in the experiment. To view the details of these variables, select Edit.
- **Initial Variables tab**. Lists variables which are given an initial value in the experiment and enables you to modify that initial value.

6 Change the run mode to Estimation and run the simulation. Wait for the solution to complete. This may take some minutes.

7 To observe the results on the Estimated Variables tab, from the Tools menu click Estimation.

8 To check the accuracy of the plot, on the Measured Variables tab of the Estimation dialog box, double-click the measured variables to edit them and observe the deviations. You may also plot these results here.

Tip: Covariance and Correlation matrices are normally available from the Estimation dialog's Status tab by clicking the matrices button. Note that the correlation matrix is computed from the covariance matrix; when the leading diagonal of the covariance matrix contains negative values, it is not possible to compute a correlation matrix.

18 Open Nonlinear Algebraic Solver Example

This example shows how to write and use a DLL containing an open Nonlinear Algebraic solver. The example implements an open solver called NLEQ1S available in:

http://www.zib.de/SciSoft/CodeLib/frame_nonlin.en.html

Note: The example is given under the assumption that you have MS-Developer's Studio installed on your machine.

Building and Using the Non-linear Algebraic Solver Example DLL

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder:
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\OpenNLASolvers

The following instructions assume that the previous directory was used. If you chose to use a different one, modify the instructions accordingly.

To build this example DLL:

1. Copy the files in the example directory to a convenient working folder, for example:
   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\OpenNLASolvers
2. Obtain a copy of the file nleq1s.f (ensure you have the rights to do so) and copy it into the directory C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 1**2**1\OpenNLASolvers, replacing the dummy one provided.

3 Double-click
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\OpenNLASolvers\OpenSolvers.dsw.

This will start the MS-Dev Studio GUI.

4 Click on the MS-Dev Studio window to make it active and press F**7** This will build the OpenNLASolvers.dll in:
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\OpenNLASolvers\Debug

5 Use your Windows Explorer to locate the file:
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\OpenNLASolvers\FiveTank.acmf

and double click it. Go to the Solver Options Non Linear Solver Tab and select the Open NLA Solver mode. In the Open NLA Solver dll box type in the full name (including the path) of, or browse for, your OpenNLASolver DLL This simulation contains nonlinear blocks in both the initialization and dynamic decompositions. The open solver NLEQ1S will be used to solve any of these nonlinear blocks.

Tip: If you receive the following error message:

After NLEQ1 IERR=-999
Group 1: Open nonlinear solver failed

it means you are using the dummy NLEQ1S Fortran code provided. You have probably forgotten to put the correct NLEQ1S.f file in position.

Using the NLEQ1S Solver

To use the NLEQ1S solver provided in the example DLL:

1 From the Run menu, click Solver Options.

2 Click on the Nonlinear Solver tab.

3 In the Mode box, select Open NLA Solver.

4 In the box provided for the DLL name, type:
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\OpenNLASolvers\Debug\OpenNLASolvers.dll

Alternatively, you can search for your DLL using the search button at the right of the box.

5 Click the Apply button (this will load the DLL into ACM) and list the available parameters.

6 You can modify the parameters by clicking on them. A dialog box will open, allowing you to set a new value.

7 Click OK

From now, whenever ACM needs to solve a nonlinear algebraic system, it will use the nonlinear solver provided in the DLL.

Debugging Your Code for the Open Nonlinear Algebraic Solver Example

If your code is not behaving as you think it should, you can debug its execution as follows. The instructions are given using the same example as above.

1. Start ACM, load your problem, and load the open solver DLL. Do not yet run it.
2. Double-click **C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\OpenNLASolvers\OpenSolvers.dsw**, which will start MS-Dev Studio. In MS-Dev Studio, click on Build->[Start Debug]->[Attach to process] and select "sim_server" as the process to debug. Click on the window containing the C++ code nleq1_int.cpp and locate line 470, press "F9" (this will add a breakpoint at this line).
3. Start your run on the Aspen Custom Modeler window. The MS-Dev Studio window will be active as soon as the execution enters the open solver DLL code. You can now start debugging it as you wish.

19 Description of Open NLP Solver Example

This example shows how to write and use a DLL containing an open NLP solver. The example implements an open solver called SRQP.

SRQP is a Successive Reduced Quadratic Programming method and is one of the built-in full space solvers supplied with Aspen Custom Modeler.

Building and Using the NLP Example DLL

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, files for this example will be in the folder:
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\OpenNLPSolvers

The following instructions assume that the previous directory was used. If you chose to use a different one, modify the instructions accordingly.

To build this example DLL:

1. Copy the files in the example directory to a convenient working folder, for example:
   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 1**2**1\OpenNLPSolvers
2. Obtain a copy of the file nleq1s.f (ensure you have the rights to do so) and copy it into the directory C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 1**2**1\OpenNLPSolvers, replacing the dummy one provided.
3. Double-click
   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 1**2**1\OpenNLPSolvers\OpenSolvers.dsw.

   This will start the MS-Dev Studio GUI.
4. Click on the MS-Dev Studio window to make it active and press F**7** This will build the OpenNLPSolvers.dll in:
   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 1**2**1\OpenNLPSolvers\Debug

5 Use your Windows Explorer to locate the file:
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\OpenNLPSolvers\opt_ss_meth.acmf

and double click it. Go to the Solver Options Optimizer Tab and select the Open NLP - full space mode. In the Open NLP Solver dll box type in the full name (including the path) of, or browse for, your OpenNLPSolver DLL. This simulation is an optimization problem and the open solver SRQP will be used to solve the optimization problem in full space.

Using the SRQP Solver

To use the SRQP solver provided in the example DLL:

1 From the Run menu, click Solver Options.

2 Click on the optimizer tab.

3 In the Solver box, select Open NLP Solver.

4 In the box provided for the DLL name, type:
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 1**2**1\OpenNLPSolvers\Debug\OpenNLPSolvers.dll

Alternatively, you can search for your DLL using the search button at the right of the box.

5 Click the Apply button (this will load the DLL into ACM) and list the available parameters.

6 You can modify the parameters by clicking on them. A dialog box will open, allowing you to set a new value.

7 Click OK

From now, whenever ACM needs to solve an NLP system, it will use the NLP solver provided in the DLL.

Debugging Your Code for the Open NLP Solver Example

If your code is not behaving as you think it should, you can debug its execution as follows. The instructions are given using the same example as above.

1 Start ACM, load your problem, and load the open NLP solver DLL. Do not yet run it.

2 Double-click
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\OpenNLPSolvers\OpenSolvers.dsw,

which will start MS-Dev Studio. In MS-Dev Studio, click on Build->[Start Debug]->[Attach to process] and select "sim_server" as the process to debug. Click on the window containing the C++ code srqp_int.cpp and locate line 497, press "F9" (this will add a breakpoint at this line).

3 Start your run on the Aspen Custom Modeler window. The MS-Dev Studio window will be active as soon as the execution enters the open solver DLL code. You can now start debugging it as you wish.

20 Bayer Process Precipitator Example

Bayer Process Precipitator Example Description

This example models the precipitation stage of the Bayer process for Alumina production.

The feed is a liquor containing caustic and alumina. This enters a precipitator. Small seed crystals are fed to this precipitator, and these grow as alumina crystallizes out from the liquor.

The solids from the crystallizer are fed to a separator which removes the smaller crystals and recycles these as seed crystals. Fresh seed crystals are also added to these before they enter the precipitator. The larger crystals from the separator are the product.

This example shows how Aspen Custom Modeler can be used to model solids processes, and in particular how particle size distributions can be tracked within streams.

This example is based upon a SPEEDUP simulation originally developed by Charles Chessari at Sydney University.

Running the Bayer Process Precipitator Example

Example simulations are included in your Aspen Custom Modeler installation. If you have installed in the default location, the files for this example will be in the folder:
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\BayerProc

To run this example, follow these steps:

1 Copy the files in the example folder to a convenient working folder

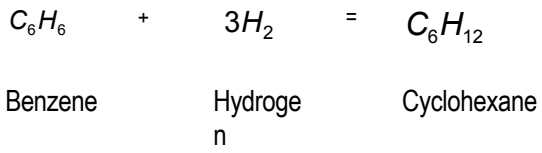
2 Double-click the file BayerProc.acmf to open it.

3 Click the run button to run the dynamic simulation.

4 The Task FeedStep steps the feed flow from 200 to 300 litres/second at time 10 hours. To see the response to this change click on a stream, right click, select Forms and select one of the tables or plots to view the stream results.

21 Exporting an Aspen Custom Modeler Flowsheet

Cyclohexane can be produced by the hydrogenation of benzene in the following reaction:



The benzene and hydrogen feeds are combined with recycle hydrogen and cyclohexane and preheated before entering a fixed bed catalytic reactor.

The reactor effluent is cooled and the light gases separated from the product stream. Part of the light gas stream is fed back to the reactor as recycled hydrogen.

The liquid product stream from the separator is fed to a distillation column to further remove any dissolved light gases and to stabilize the end product. A proportion of the cyclohexane product is recycled to the reactor to aid in temperature control.

In the Aspen Plus simulation, the reactor is modeled as a stoichiometric reactor (RStoic block) with an assumed fixed fractional conversion of benzene of 99.8%. The reactor model in an Aspen Custom Modeler simulation includes a rate expression.

$$Rate = k * C_{Benzene} * C_{hydrogen}^{\wedge 3}$$

Exporting the Cyclohexane Production Flowsheet

To be able to run this example:

- You must have Aspen Plus, Aspen Custom Modeler, and a Microsoft C++ compiler installed, tested, and running.
- Your compiler must be able to run from the command line.
- Aspen Plus must run with the same server computer and in the same working folder as you have configured for Aspen Custom Modeler.

Example problems are included as part of your Aspen Custom Modeler* installation. If you have installed in the default location, the files for this example will be in the folder:
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\FlowSheetExport

To prepare to run the example in Aspen Custom Modeler:

- Copy the files in the installation example folder to a convenient working folder.

Setting up the Properties Plus Interface

You need to first set up Aspen Properties Plus with the Aspen Modeler interface. This can be done with Aspen Plus or with Aspen Properties:

1 Open the Aspen Plus User Interface and load the Flowsheet Export feature example backup file, named Cyclohexane.bkp.

2 Run the problem. Note that this is a Properties Plus run type with no flowsheet.

3 When the Properties Plus setup is completed, from the File menu, click Save As to save as an Aspen Plus document (.apw extension) file, and name the file cyclohexane.

The cyclohexane.appdf file is automatically saved for use with Aspen Custom Modeler.

Note: Make sure that the cyclohexane.appdf file is saved in the same location as the cyclohexane.acmf file.

4 Exit Aspen Plus.

Running the Cyclohexane Production Flowsheet Export Example

To run the example in Aspen Custom Modeler:

1 Open Aspen Custom Modeler and load the Cyclohexane.acmf file. The flowsheet contains three blocks and two streams.

2 The RX_FEED block is an APlusFeed model. The RX_PROD block is an APlusProd model. These models are used to convert the Aspen Modeler system base units of measurement set (Metric) into SI units. SI units are the valid units for export to Aspen Plus.

The APlusFeed and APlusProd models are used to map the MoleFractionPort_SI port type (required for the Compiled Flowsheet Export tool) to a MoleFractionPort port type and vice versa. The MoleFractionPort port type is the port type used by the reactor model block itself and by extension, the larger flowsheet, if there was one.

The port variables to be exported and therefore to be used in the Aspen Plus flowsheet are defined by the port named MoleFractionPort_SI found in the Modeler library.

Note: All model types, port types, variable types, parameter types etc. referenced in this example exist in the delivered Modeler or Dynamics libraries.

Run the simulation in steady state run mode.

3 Review the key results from the ReactorResults table in the Simulation/Flowsheet contents folder.

Exporting the Flowsheet

To use export the flowsheet:

1 From the Tools menu, click Export Compiled Flowsheet.

The Flowsheet Export menu is displayed.

2 On the Compiled Flowsheet Export form, change the Export Directory to be the same as the directory where the Cyclohexane.bkp. file is located, for example:
E:\Aspen Products\Working Directory\FlowSheetExport

3 In the Library File box, specify the directory and the name of the Aspen Plus User Model Library (.apm extension) file to be generated. This directory is the same location as the original Cyclohexane.bkp file. For example:
E:\Aspen Products\Working Directory\FlowSheetExport\cyclohexane.apm

Note: The first time this file is specified, you will be asked if you want to create one, since the file does not yet exist.

4 In the Inputs Table box, select ReactorInputs from the list. This is the form that defines the variables that Aspen Plus will display as inputs.

5 In the Results Table box, select ReactorResults from the list. This is the form that defines the variables that Aspen Plus will display as results.

Note: Forms must have been previously created in the Flowsheet folder to be available in the list.

6 To generate the DLL and the .apm (Aspen Plus user model library) file, click OK.

7 Check the Simulation Messages window to see if the Aspen Plus model library flowsheet export and the dll generation process succeeded. The dll and the apm extension files should have been created in the specified export and library directories.

8 Exit Aspen Custom Modeler.

Running the Exported Flowsheet in Aspen Plus

To run the flowsheet you have exported:

1 Start Aspen Plus and load the Cyclohexane.bkp file. There is no flowsheet as the run type is still Properties Plus.

2 Change the run type to Flowsheet from the Data/Setup/Specification form under the Global settings section.

The flowsheet is displayed.

3 Run the simulation.

4 When the run completes successfully, reconcile streams RX-FEED and RX-PROD to provide good estimates for the Aspen Custom Modeler (ACM) reactor feed and product streams. To do this, select the stream, click the right mouse button to display the Stream menu, and then click Reconcile.

5 From the menu bar, click Library/References.

6 On the Library References form, click Browse and navigate to the directory where the Cyclohexane.apm file is located. Open Cyclohexane.apm to add the exported Aspen Custom Modeler flowsheet to the Aspen Plus model library.

Note: Before closing the Library/References form, make sure that Cyclohexane.apm is selected in the displayed list of available libraries.

7 Click OK to leave the Library references form.

A new tab named ACM Flowsheets is added to the model palette.

8 In the Process Flowsheet (PFS) window, delete the RSTOIC reactor block R-10**4**

9 Select the ACM Flowsheets tab and drag the Cyclohexane icon onto the PFS to replace block R-10**4**

10 Connect existing stream RX-FEED to the inlet of the new block (the exported ACM flowsheet) and connect stream RX-PROD to the outlet of the new reactor. The flowsheet is now complete.

11 Use the Next button, ACM reactor block pull down menu, or the Data Browser to access the ACM Cyclohexane reactor block input forms.

12 On the Connections form for the new block, click the stream field for Port ID RX-FEED and select stream RX-FEED, and click the stream field for Port ID RX-PROD and select stream RX-PROD.

Note: In INPUT mode, the Variables tab form displays the variables defined in Aspen Custom Modeler ReactorInputs form.

13 Click the Next button to run the simulation.

14 When the run completes, use the block pull down menu to view the ACM Reactor model block results from the ReactorResults form.

22 Exporting Models for Use in Aspen Plus

To run this example:

- You must have Aspen Plus, Aspen Custom Modeler, and a Microsoft C++ compiler installed, tested, and running.
- You must be able to run your compiler from the command line, i.e., ensure that the required C++ compiler directories are part of the PATH, INCLUDE, and LIB environment variables.
- Aspen Plus must run in the same server computer and in the same working folder as you have configured for Aspen Custom Modeler.

Example problems are included as part of your Aspen Custom Modeler* installation. If you have installed in the default location, the files for this example will be in the folder:
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\ModelExport

To prepare to run the example in Aspen Custom Modeler:

- Copy all the files in the example folder to a convenient working folder.

Setting up the Properties Plus Interface

You need to first set up the interface between Aspen Custom Modeler and Aspen Properties Plus. This can be done with Aspen Plus:

- Open the Aspen Plus User Interface and load the example backup file, named MyPipe.bkp.
- Run the problem. Note that this is a simple process flowsheet that simulates the flow of water through an Aspen Plus pipe model. The flowsheet contains a pipe model named B1 from the Aspen Plus model library and two streams named S1 and S2.
- Once the run is completed, from the "File" menu, click "Save As" to save as an Aspen Plus document (.apw extension) file, and name the file MyPipe. A file called MyPipe.appdf is automatically saved. This file may be used with Aspen Custom Modeler. Ensure that the MyPipe.appdf file is saved in the same location as the MyPipe.acmf file.
- Exit Aspen Plus.

Running the MyPipe Model Export Example

To run the example in Aspen Custom Modeler:

- Open Aspen Custom Modeler and load the MyPipe.acmf file. The flowsheet contains one block named MyPipe and two streams named S1 and S2.
- The MyPipe block is a model that simulates the flow of water through a pipe using partial differential and algebraic equations. The other features of this model include:
  - An external procedure call to calculate the friction factor
  - A script called "presolve" that is executed in Aspen Plus before the sequential modular calculations for this block.
  - A table called "input_par" that allows the user to input new values for the blocks input parameters in Aspen Custom Modeler and Aspen Plus
  - A table called "pressure_profile" that displays the results of particular interest to the model writer in Aspen Custom Modeler and Aspen Plus
  - An icon that is available to the user in Aspen Custom Modeler and Aspen Plus
- From the Exploring–Simulation pane, the modeling language source for the MyPipe model may be viewed by right clicking on Simulation\Custom Modeling\Models\MyPipe and by picking “Edit” from the menu. The source shows that this model uses a port type called “MaterialPort”. This port type facilitates material stream connections in Aspen Plus as it includes the flow rate, temperature, pressure, molar volume, molar enthalpy and the component mole fractions of the stream. The modeling language source for the “MaterialPort” may be viewed under Simulation\Custom Modeling\Port Types\MaterialPort.
- Note that all the model types, port types, variable types, parameter types etc. that are referenced in this example exist in the delivered Modeler or Dynamics libraries.
- Click on Simulation\Flowsheet, find the “InitMyPipe” script in the Contents pane and run the script by double-clicking on it.
- Click on Simulation\Flowsheet\Blocks\MyPipe
- From the Contents pane, open the “Input_Par” table and the “Pressure Profile” table by double clicking on these tables.
- Run the simulation in steady state run mode.
- Review the results displayed in the “Pressure Profile” table.
- In the “Input_Par” table change the value of the “Roughness” to 1e-4 and run the simulation in steady state run mode.
- Review the results displayed in the “Pressure Profile” table.
- In the “Input_Par” table change the value of the “NumElements” to 20. Note that the “Pressure Profile” table is automatically resized to 20 elements. In fact, all of the discretized variables in the model are now calculated at 20 nodes and the number of model variables and equations are changed. Run the simulation in steady state run mode.
- Review the results displayed in the “Pressure Profile” table.

Exporting the Model for use in Aspen Plus

To export the model:

- From the Exploring–Simulation pane, right click on Simulations\Custom Modeling\Models\MyPipe and pick the "Model Package Properties" wizard. Explore the various options available through the wizard for configuring the install package and accept the default options as saved in MyPipe.acmf.
- From the Exploring – Simulation pane, right click on Simulations\Custom Modeling\Models\MyPipe and pick the "Export" wizard, toggle the "Save As Type" box to "Model Installation Package(*.msi)", pick a convenient directory and click the "Save" button.
- When Aspen Custom Modeler asks whether you want to install the package, hit "Yes" and follow the installation instructions.
- Exit Aspen Custom Modeler.

Using the Exported Model in an Aspen Plus Flowsheet

- Open Aspen Plus and load the MyPipe.bkp file. The flowsheet contains one block named B1, a pipe model from the Aspen Plus model library, and two streams named S1 and S2.
- In the "Process Flowsheet" window, delete the block B1 by clicking on it and then pressing the delete key.
- From the "Library" menu, pick "References" and check the box next to the "ACM Models". A new pane called "ACM Models" will appear in the model library palette.
- Click on the "ACM Models" pane to view the MyPipe model that you exported from Aspen Custom Modeler.
- Click on the downward facing arrow next to the MyPipe model, pick the icon that was exported with the MyPipe model and place a block of this type on the process flowsheet window.
- Connect stream S1 to the inlet port of MyPipe and stream S2 to the outlet port of MyPipe.
- Double click on the MyPipe block and explore the block's input and results forms by toggling the view from "Input" to "All". The "Variables" pane will have two forms, one called "Pressure_Profile" and another called "AllVariables" that were exported along with your model. The "Parameters" pane has two forms, one called "Input_Par" and another called "AllParameters".
- From the "View" menu pick "Control Panel" and run the simulation in "Sequential Modular" mode. You'll see a message box titled VBScript that pops up before the MyPipe block is solved. This message box was displayed because the "presolve" script that was exported with the MyPipe contains VB commands that pop up message boxes when the MyPipe

model is run in Sequential Modular mode. Review the results displayed in the "Pressure_Profile" form.

- Go back to the MyPipe block's "Parameters" pane, change the value of the "Roughness" parameter from 1e-5 to 1e-4 and run the flowsheet in the sequential modular mode. Review the results displayed in the "Pressure_Profile" form.
- In the "Parameters" pane change the value of the "NumElements" parameter from 10 to 20. Note that the "Pressure Profile" form now displays 20 elements. Run the flowsheet in the sequential modular mode and review the results displayed in the "Pressure Profile" form. Note that a change in the "NumElements" parameter automatically changes the structure of the model ,i.e., the number of variables and equations in the model. This structural change results in the execution of the "presolve" script as per the conditional statements coded in the script.
- In the Control Panel window, toggle the solution mode from "Sequential Modular" to "Equation Oriented" and solve the flowsheet in "Equation Oriented" mode.
  Note that the model that was written in Aspen Custom Modeler may be run in both "Sequential Modular" and "Equation Oriented" modes in Aspen Plus and that this model was exported with automatically generated analytical derivatives for its equations.
- Exit Aspen Plus.

23 Handling Events in a Visual Basic Form

Overview of Handling Events in a Visual Basic Form

This example shows how a hidden Visual Basic form can receive event notifications generated by Aspen Custom Modeler.

The Visual Basic project ACMEvents.vbp contains one form to act as the user interface for the test application [UserInterfaceForm] and a second, hidden form to receive event notifications.

Note: If you have installed Aspen Custom Modeler in the default location, the Microsoft Visual Basic source code for the project in this example is available at:
C:\Program Files\AspenTech\Aspen Custom Modeler 12.1\Examples\ACMEvent\

For more information about Application Object events, see the Automation Reference.

Key Steps in handling ACM Events in a VB Form

To create the necessary DLL, complete the following procedure:

- Start Microsoft Visual Basic.
- Create a new project, selecting the Standard EXE type.
- Go to Project|References and in the Available References list, select Aspen Custom Modeler 12.1 Type Library.
- Create a form to handle ACM Events. Set the Visible property to false and add the following line at the top of the form's code:

```
Implements IAspenModelerEvents.
```

- In the Object list at the top of the code for this form, select IAspenModelerEvents.
- Create a code stub for each of the events in the Procedure list by selecting each of them in turn.
- Insert your event-handling code in the relevant stubs.
- Create a second form to instantiate ACM and the event-handling form.

Running the ACMEvents Example

To run the ACMEvents example:

- Start Visual Basic.
- Load the ACMEvents.vbp project.
- In VB, select Run | Start.
- ACM is started and the fivetank problem loaded. The solution is stepped once and the problem loaded. The listbox in the UserInterfaceForm shows the events as they occur.

24 Gas Permeation Module Example

This example simulates a gas permeation module used to produce oxygen enriched air from air.

It includes a general purpose gas permeation module model called GasPermModule. The retentate is the feed material that has not crossed the membrane. This flows through each of the individual cells in the module in turn. The number of cells defaults to 100 but can be changed when you use the model. The permeate is the material which has passed through the membrane, in this example the oxygen enriched air. The permeate is removed from each cell directly and mixed before leaving the module. Flow from the retentate to permeate side is driven by a higher pressure on the retentate side.

The model assumes:

- Cross flow with unhindered permeate withdrawal
- Isothermal conditions
- Ideal gas behavior
- Constant permeabilities for each component

By entering appropriate permeability values, pressures and number of cells this model can be used to simulate other gas permeation processes. It can also be exported for use in Aspen Plus. For some applications you may need to extend the model to account for the variation of permeabilities with concentration or temperature.

Setting Up Properties Plus for the Gas Permeation Example

You need to set up Properties Plus for the simulation. You only need to do this once:

1. Locate the example folder GasPermeation. If you copied the files to the example working folder, this is:
   C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 121\GasPermeation
2. Load the file GasPermeation.bkp in to Aspen Plus and run it.

Note: This is a properties only input file, so you will not see a flowsheet in Aspen Plus.

3 From the File menu, click Save As and save the simulation as an Aspen Plus document (.apw file), then exit Aspen Plus.

Now you can run the example.

Running the Gas Permeation Example

When you have set up Properties Plus, you can run the Absorber example. To do this:

1 In Aspen Custom Modeler, from the File menu, click Open.

2 Open the GasPermeation folder. If you copied the files to the example working folder, this is:
C:\Program Files\AspenTech\Working Folders\Aspen Custom Modeler 1**2**1\GasPermeation

3 Click the example file GasPermeation.acmf then click Open.

4 When the simulation has loaded click the Run button to run it.

5 On the flowsheet double click on the block called Module. This opens the Summary form for the gas permeation module. You can see for the default operating conditions the oxygen fraction in the permeate is 3**1**6%.

6 Try changing the inlet pressure (Inlet.P) or the permeate pressure to see the effect on the permeate composition.

7 Change the Spec of the permeate oxygen fraction (Permeate.z("O2")) to Fixed and the spec of the permeate pressure to Free. Change the permeate oxygen fraction to 0.3 and run to calculate the required permeate pressure. You may want to try other values for the oxygen fraction. You will find that the simulation cannot solve for values above 0.364, as this is maximum that can be achieved even with a very low permeate pressure.

25 Glossary

Glossary Definitions

.acmf file

An ASCII language file of a saved Aspen Custom Modeler simulation.

.acmd file

A binary file of a saved Aspen Custom Modeler simulation.

.ada File

An ASCII language file of a saved Aspen ADSIMsimulation.

.adb File

A binary file of a saved Aspen ADSIM simulation.

.cra File

An ASCII language file of a saved Aspen Chromatography simulation

.crb File

A binary file of a saved Aspen Chromatography simulation.

Active variable

An active variable is a variable that occurs in an equation in the current simulation.

ActiveX control

A control, for example, a check box , a list box, or a command button, that is created with the Microsoft ActiveX Control Toolbox. Controls can offer options to users or run macros to automate tasks.

Add Form icon

Click to define a form such as a plot, table, profile plot or ActiveX control form.

Add Script icon

Click to define scripts for automating specification and initialization.

Add Task icon

Click to define tasks that define a sequence of actions to take place during a dynamic simulation.

Automation

Enables applications to provide objects to other applications.

Block

A block is an instance of a model on a flowsheet.

Blocks folder

A list of all the blocks present in the current flowsheet.

Communication interval

The time interval at which data is available for plots and snapshots during a simulation.

Component Lists folder

Use the Components List folder to define a list of component names. The component names can be associated with a physical properties package or not. A component list called Default is provided.

Decomposition

The process of gathering the equations in a simulation into minimum size groups which are solved sequentially.

Degrees of freedom

The number of equations in relation to the number of unknown variables.

Diagnostics folder

The Diagnostics folder gives access to available decomposition information, that is, groups of equations with their associated variables.

Form

Table, plot, or Microsoft ActiveX control form, used to display results.

Global parameters

Global parameters are parameters that are defined in the modeling language, outside of type definitions.

Group

A collection of equations which can come from different parts of a simulation.

Homotopy

Enables you to step a steady-state solution through to another steady-state solution by defining fixed variables.

Indeterminate variables

An indeterminate variable is one that has no unique value, that is, any value will satisfy your equations.

IntegerSet

An IntegerSet is a set whose members are integers, for example, 1:10.

Kept snapshots and results

A kept results is a result which has been marked to be kept in the simulation input file in ASCII form. This means that you do not need access to any result files to be able to access a kept result. You access a kept result in the same manner as other results. When a snapshot is marked as kept it indicates that when the snapshot is archived, it will be saved as a kept result.

Model

A model is a set of equations that mathematically represent a process. On a flowsheet a model is represented by an icon.

Parameter

Parameters are used to define values that are fixed for the whole simulation, such as acceleration due to gravity, or the gas constant, or the number of trays in a column.

Partial differential algebraic equation

An equation that contains both spatial and time derivative terms, for example:

$$\frac{\partial T}{\partial t} = \frac{k}{C_p \rho}\left(\frac{\partial^2 T}{\partial x^2}\right)$$

Partial differential equation

An equation that includes spatial derivative terms.

Pause

Suspends a dynamic simulation at any time.

Phase Plot

A plot of a set of variables against another variable on the X axis. The data plotted is time based but time is not shown on the plot.

Port Type

A port type is similar to a flange on a vessel or a pipe. It defines the connectivity for models and streams. In other words, a port type defines what process variable information is passed between models and streams.

Procedure

Procedures are definitions of external routines used in models to calculate variables instead of equations.

Profile plot

A plot of a series of variables, for example, temperature profile, against an index (the number of trays within a column or the number of points used to discretize a partial differential equation) or against another series of variables, for example, the length of a reactor bed.

Recorded history

Also known as Time History. A set of values for a variable, recorded over a period of time, at communication intervals set in the Run Options dialog box.

Reset

The current solution point is removed and all variables are returned to their default values.

Restart

The next dynamic run starts from its initial conditions at time = 0.

Results

A result is a set of data which contains value and specification data for a set of variables. The set of variables need not correspond to the variables present in any simulation and they do not contain delay or workspace information. You can use results to initialize the values of variables in a simulation but it cannot be guaranteed that a solution will be possible when using a results file as it could be that not all variables in the simulation exist in the result.

Rewind

For a dynamic run, Rewind takes the simulation back to a previous time point specified by a snapshot, if snapshots are available. If there are no snapshots, or if the simulation is not dynamic, Rewind returns the simulation to time = 0.

Run-time plot

A plot of a set of variables against time.

Script

A script is a set of instructions, written in Microsoft Visual Basic, to automate the setup of a simulation. A simple script is a list of assignments for parameters and variables such as default value, specification, lower and upper bounds. Scripts can also be used to control sequences of simulation runs.

Snapshot

A snapshot is a set of data whose structure corresponds exactly to the current simulation structure, that is, there is a value and spec available for every variable in the simulation, in addition to any delay information and procedure

workspace data. You can use a snapshot to replicate the state of a simulation. Snapshots are generated automatically as the simulation solves, and also on request. Snapshots are stored in a file snapshot.snp in the simulation working directory, which is on the server machine. When the simulation is edited, for example, a new block is added to the flowsheet or a structural parameter is changed, snapshots are archived and they become results. A new results file is generated.

State variable

A state variable is a variable differentiated with respect to time.

Step

Steps through a dynamic simulation one time interval at a time.

Stream

A stream is an instance of a stream type on a flowsheet.

Stream type

A stream type is similar to a pipe and is a special type of model in Aspen Custom Modeler. On a flowsheet, it is represented by a line.

Streams folder

A list of all the streams present in the current flowsheet.

StringSet

A StringSet is a set whose members are strings, for example, component names.

Task

A task is a set of instructions that you can create in Aspen Custom Modeler. A task defines a sequence of actions that take place during a dynamic simulation, such as changing feed flow rate.

Time history

Also known as Recorded History. A set of values for a variable, recorded over a period of time, at communication intervals set in the Run Options dialog box.

Type

Types are the main components of a simulation, such as models, parameters, ports, procedures, streams, tasks, and variables. Types can be either defined in the modeling language or built graphically.

Variable

A variable is a component of an equation.

Variable type

A variable type is used to define an initial guess and range for variables being solved in a simulation.

General Information

Copyright

Version Number: 2004.1
October 2005

Aspen ACOL™, Aspen Adsim®, Aspen Advisor™, Aspen Aerotran®, Aspen Alarm & Event™, Aspen APLE™, Aspen Apollo Desktop™, Aspen Apollo Online™, Aspen AssetBuilder™, Aspen ATOMS™, Aspen Automated Stock Replenishment™, Aspen Batch Plus®, Aspen Batch.21™, Aspen BatchCAD™, Aspen BatchSep™, Aspen Calc™, Aspen Capable-to-Promise®, Aspen CatRef®, Aspen Chromatography®, Aspen Cim-IO for ACS™, Aspen Cim-IO for Csi VXL™, Aspen Cim-IO for Dow MIF™, Aspen Cim-IO for G2™, Aspen Cim-IO for GSE D/3™, Aspen Cim-IO for Hewlett-Packard RTAP™, Aspen Cim-IO for Hitachi PLC (H04E)™, Aspen Cim-IO for Intellution Fix™, Aspen Cim-IO for Melsec™, Aspen Cim-IO for WonderWare InTouch™, Aspen Cim-IO for Yokogawa Centum CS™, Aspen Cim-IO for Yokogawa Centum XL™, Aspen Cim-IO for Yokogawa EW3™, Aspen Cim-IO Interfaces™, Aspen Cim-IO Monitor™, Aspen Cim-IO™, Aspen Collaborative Demand Management™, Aspen Collaborative Forecasting™, Aspen Compliance.21™, Aspen COMThermo TRC Database™, Aspen COMThermo®, Aspen Cost Factor Manual™, Aspen Crude Manager™, Aspen Crude Margin Evaluation™, Aspen Custom Modeler®, Aspen Data Source Architecture™, Aspen Decision Analyzer™, Aspen Demand Manager™, Aspen DISTIL™, Aspen Distribution Scheduler™, Aspen DMCplus® Composite, Aspen DMCplus® Desktop, Aspen DMCplus® Online, Aspen DPO™, Aspen Dynamics®, Aspen eBRS™, Aspen Enterprise Model™, Aspen ERP Connect™, Aspen FCC®, Aspen FIHR™, Aspen FLARENET™, Aspen Fleet Operations Management™, Aspen Framework™, Aspen FRAN™, Aspen Fuel Gas Optimizer Desktop™, Aspen Fuel Gas Optimizer Online™, Aspen General Construction Standards™, Aspen Hetran®, Aspen HX-Net®, Aspen Hydrocracker®, Aspen Hydrotreater™, Aspen HYSYS Amines™, Aspen HYSYS Crude™, Aspen HYSYS Dynamics™, Aspen HYSYS OLGAS 3-Phase™, Aspen HYSYS OLGAS™, Aspen HYSYS OLI Interface™, Aspen HYSYS Tacite™, Aspen HYSYS Upstream Dynamics™, Aspen HYSYS Upstream™, Aspen HYSYS®, Aspen Icarus Process Evaluator®, Aspen Icarus

Corporate

Aspen Technology, Inc.	Phone: (1) (617) 949-1000
Ten Canal Park	Toll Free: (1) (888) 996-7001
Cambridge, MA 02141-2201	Fax: (1) (617) 949-1030
USA	URL: http://www.aspentech.com

Related Documentation

In addition to this document, the following documents are provided to help users learn and use the Aspen Utilities applications.

Title	Content
Aspen Custom Modeler 2004.1 Getting Started Guide	Contains basic hands-on tutorials to help you become familiar with Aspen Custom Modeler.
Aspen Custom Modeler 2004.1 Modeling Language Reference	Contains detailed reference information about the modeling language, including syntax details and examples.
Aspen Custom Modeler 2004.1 Library Reference	Contains reference information on control models, property procedure types, utility routines, port types, and variable types.
Aspen Custom Modeler 2004.1 Aspen Modeler Reference	Contains information on using automation, solver options, physical properties, the Control Design Interface (CDI)), Simulation Access eXtensions, online links, and using external nonlinear algebraic solvers.
Aspen Custom Modeler 2004.1 DMCplus® Controllers Interface	Contains information on using DMCplus with Aspen Custom Modeler or Aspen Dynamics™.
Aspen Custom Modeler 2004.1 Polymer Simulations with Polymers Plus	Polymers Plus is a layered product of Aspen Custom Modeler. It provides additional functionality to the properties package, Properties Plus, enabling polymers to be fully characterized in Aspen Custom Modeler models.

Technical Support

Online Technical Support Center

AspenTech customers with a valid license and software maintenance agreement can register to access the Online Technical Support Center at:

http://support.aspentech.com

You use the Online Technical Support Center to:

- Access current product documentation.
- Search for technical tips, solutions, and frequently asked questions (FAQs).
- Search for and download application examples.
- Search for and download service packs and product updates.
- Submit and track technical issues.
- Search for and review known limitations.
- Send suggestions.

Registered users can also subscribe to our Technical Support e-Bulletins. These e-Bulletins proactively alert you to important technical support information such as:

- Technical advisories.
- Product updates.
- Service Pack announcements.
- Product release announcements.

Phone and E-mail

Customer support is also available by phone, fax, and e-mail for customers who have a current support contract for their product(s). Toll-free charges are listed where available; otherwise local and international rates apply.

For the most up-to-date phone listings, please see the Online Technical Support Center at:

http://support.aspentech.com

Support Centers	Operating Hours
North America	8:00 – 20:00 Eastern time
South America	9:00 – 17:00 Local time
Europe	8:30 – 18:00 Central European time
Asia and Pacific Region	9:00 – 17:30 Local time

Index

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W